البدايات المضيئة

في

Visual Basic

2012

اعداد : معاذ الزفتاوي

Email: tashfeen1985@gmail.com

ت : 201120031403+

مقدمة السلسلة

الحمد لله الذي هدانا وما كنا لنهتدي لولا ان هدانا الله و الصلاة و السلام على خير الانام سيدنا محمد و على اله و صحبه و سلم وبعد

فالى اخواني في الوطن العربي الكبير من الخليج الى المحيط اقدم اليكم هذه السلسة المباركة و التي اسال الله العلي القدير ان تكون سببا في تفوقكم في حياتكم العملية وان تكون سببا في تقدم اوطاننا وان تستعيد امتنا قيادتها للامم في العلم و المعرفة

فهذه السلسلة تشرح كيف تصبح مبرمج محترف فهي تاخذ بيدك من البداية وحتى تصبح مبرمج محترفا مطلوبا في سوق العمل ، فعالم البرمجة عالم ممتع وسهل وليس فيه كما يتصور الكثير انك يجب ان تكون على دراية كافية باللغة الانجليزية او تكون عالما في الرياضيات بالتاكيد اذا هذا تصور خاطئ فعالم البرمجة سهل على اي شخص ان يدخله ولكن بشرط ان يكون طموحا وصبورا وان يكون مستعد ان يطور نفسه بشكل مستمر .... وهذه السلسلة التي اقدمها بين ايديكم هي برهان على ذلك فهذه السلسلة تتكون من الكتب الاتية

1. مبادئ المبرمجة باسلوب OOP وكيف تصمم ادوات التحكم user control
2. فن الرسم باستخدام بمكتبة GDI وكيف تطبقها في حياتك العملية
3. فن التعامل مع ملفات النظام باستخدام مكتبة OI
4. فن التعامل مع قواعد البيانات واستخراج التقارير باستخدام مكتبة ADO.net

هذه االسلسله هي السلسلة الاولى التي اصدرها وبعدها باذن الله سوف اصدر سلسلة جديدة تطبق كل ما تعلمته من السلسلة تطبيقا عملية على ارض الواقع

و الله اسال ان تنال هذه السلسلة اعجابكم وان تزيد من معرفتكم في عالم البرمجة الممتع

المقدمة

الحمد لله و كفى و صلاة و سلام على عباده الذين اصطفى وبعد

فإلى اخواني في الوطن العربي كله وفي بلدي مصر خاصة اقدم اليكم هذا العمل المتواضع الذي اسال الله العلي القدير ان ينال اعجابكم فهذا الكتاب يجمع لك جميع الاساسيات التي يجب ان تتعلمها وانت تريد ان تدخل الى عالم البرمجة وياخذ بيدك ليعلمك كيف تبدا حياتك العملية في عالم البرمجة لاننا ما احوجنا الا وجود مبرمجين يطورون برامج تضاهي بقوتها البرامج العالمية فانني احلم بمن يقوم بعمل لغة برمجة عربية تعمل على بيئة رسومية وتعتمد على تقنية البرمجة الكائنية ويكون لها مكتبة خاصة مثل بيئة الدوت نت فهذا حلم اتمنى ان اراه حقيقة وقد رايت جزء منه حيث طور بعض الاخوة العرب لغة برمجة عربية يعني تكتب الاوامر بغة عربية وقاموا تطويرها باستخدام تقنية الدوت نت وهذا ما اثلج صدري وتمنيت ان ارى هذه اللغة تظهر في النور وتضاهي لغات البرمجة العالمة وغيرها من البرامج التي اتمنى ان تنافس البرامج العالمية في سوق العمل لذا قررت ان اقدم لاحبائي القراء هذا الكتاب البسيط الذي يجمع كل مبادئ البرمجة ويضعك على اول الطريق للعمل في هذا المجال .

واهمس في اذنك عزيز القارئ بان هذه المهنة مثل اي مهنة اخرى تحتاج الى صبر وطول البال وتحتاج الى العمل المستمر بل يجب ان تطور نفسك بشكل مستمر بان تتعلم التقنيات الجديدة الي تنزل في السوق لذا فاني اريد من زملاء المبرمجين الجدد ان يتحلوا بصفة الصبر... نعم الصبر على تعلم التقنيات فهي تحتاج الى ان تقراها مرة وثانية و ثالثة وتجرب الاكواد عدة مرات حتى تفهم لماذا كتبناه هذا الكود، وايضا ستحتاج الى الصبر على جهاز الكمبيوتر لان هذا الجهاز الة غبية لا تفهم ما تريد الا ما تكتبه لها بلغتها فقط فتقوم بتنفيذ ما تأمربه فقط و لا تستطيع ان تدخل الى عقلك لتفهم ما تريده منها

و اريدك ان يكون لديك حب التعلم وان تطور نفسك بشكل مستمر ان تعتمد على نفسك بشكل كلي لانك مهما تعملت من اكواد وادوات واساليب جديدة ستفاجئ بان العميل يطلب بعمل برنامج تستخدم فيه تقنية انت لم تتعلمها في اي مكان او في اي كتاب او بعبارة اخرى بهلجتنا المصرية (**انا ما اخدتهاش في الكلاس**) وكثير ما ستفاجئ بهذه المواقف الغريبة التي تضطران تجري هنا وهناك لكي تحصل على هذه التقنية لتحقيق ما يطلبه منك العميل

كما يجب ان يكون لديك المرونة الكافية لكي تستخدم الادوات التي بين يديك لابتكار تقنية جديدة فكثير من التقنيات التي تظهر في السوق ما هي اللا نتاج ربط عدة ادوات واكواد برمجية بعضها ببعض كما ستلاحظ اثناء جولتنا في هذا الكتاب وفي باقي السلسلة

واخص هذه الفقرة لمن لديهم خبرة جيدة في البرمجة او من اخذ قسط في البرمجة فهذا الكتاب سيكون مرجع سهل له باذن الله للوصول الى الادوات التي يريدها اما من تعلم لغة الفيجوال بيسك 6 و يريد ان يدخل في لغة الفيجوال بيسك دوت نت فاني اخبرة بان ينسى ما تعلمه تماما وينحيه جابنا .... قد يكون الامر صادما لك ولكن هذه هي الحقيقة المرة لان لغة الفيجوال بيسك دوت نت ليست فيجوال بسك 6 فهناك فرق شاسع بين لغة الفيجوال بيسك 6 و الدوت نت فالدوت نت تعتمد على تقنية وفيجوال بيسك 6 تعتمد على تقنية اخرى و انصحك ان لا تكرر الخطا الذي وقعت انا فيه . نحيها جانبا وتعلم هذه الغة من اول و جديد لان لغة الفيجوال بيسك دوت نت ما هي الا لغة C# ولكن عدلتك صيغها لتصبح لغة الفيجوال بيسك كعمل تجاري من شركة مايكروسوفت لجذب مبرمجي لغة الفجوال بيسك 6 فقط لاغير

ننتقل الان الى المادة التي نريد ان نتعلمها في هذا الكتاب ستتعمل مبادئ لغة البرمجة من الصفر فسوف تتعلم الاتي

في الباب الاول

- ما هي المتغيرات وما انواعها
- كيف انشئ الدوال و الاجراءات
- ما الكائنات وما مكوناتها
- ما معنى الوراثة وما معنة تعدد الاوجه
- ما هي الخصائص و الاحداث وما معنى الفويضات
- ما هي بمادئ البرمجة الموجهه OOP
- ما هي الهياكل و الواجهات

الباب الثاني

- ما الواجهات الرسومية ما هو تطبيقات النوافذ
- ما هي ادوات تطبيقات النوافذ
- كيف ننشئ تطبيقات النوافذ
- كيف نحسن من ادائها
- كيف نكتشف اخطاء البرمجة
- بعض الاضافات التي يجب ان تعرفها

كما اريد ان انبهك عزيز القارئ ان هذا الكتاب ليس فقد لمن يمتلك الفيجوال ستوديو اصدار 2012 ولكن لكل الاصدارات وسوف تجد انها واحدة واي اختلاف في الاصدارات سوف اوضحها لك باذن الله فهي بنا نبدا دخول عالم البرمجة

قبل ان نبدا اريدك ان تتعرف عن تاريخ البرمجة

مقدمة عن البرمجة

بدات لغات البرمجة بعد ان قام جوزيف ميري جاكرد في عام 1801 باختراع اله للحياكة يتم التحكم بها من خلال بطاقات مثقبة وهذه البطاقات عبارة عن ورق مقوى مثقوب بتنسيقات معينة تقوم بادخال البيانات المناسبة للالة ، ثم بدات صناعة اول حاسب الي عرفته البشرية في عام 1820 حيث قام البريطاني شارلز باينج باختراع الة تسمى الالة التحليلية وكذلك يتم التحكم بها عن طريق البطاقة المثقبة ايضا وتدار بالمحرك البخاري و في عامل 1890 قام هيرمان هولوريث باختراع الة تعملة على بطاقة مثقبة ولكن على حجم ورقة الدولار ساعدت الحكومة الامريكية باجراء التعداد السكاني في اقل من سنتين ونصف بدلا من 7 سبع سنوات بحيث يتم ان البطاقة تحتوي على عدة ثقوب فمن خلال الثقوب يتم بتحديد بيانات الشخص كالنوعه ومكان الولادة وغيرها

واستمرت استخدام البطاقة المثقبة حتى السبعينيات من القرن العشرين الى ان ظهرت لغات البرمجة و التي يمكن تقسيم تطورها على ثلاثة اجيال

الجيل الاول

ظهرت اول لغة برمجة وهي اللغة المعروفة بالنظام الثنائي Binary system بحيث يتم ادخال الاوامر للحاسب عن طريق استخدام لغة الالة 1-0 بحيث ان 0 تعني الدائرة مفتوحة يعني التيار منفصل و 1 تعني الدائة مغلقة يعني التيار متصل

الجيل الثاني

تعين على المبرمجين ان يطوروا لغة لحل مشكلة البرمجة بلغة الالة حيث انه تم تطوير لغة مجمعة تقوم بترجمة رموز و كلمات مكونة من 1-5 احرف الى لغة الالة فعرفت الى اللغة المجمة Assembly languge

الجيل الثالث

كانت لغة البرمجة Assembly امرا سهل في ترجمة الاوامر للغة الالة لكنها كانت امرا صعبا جدا من حيث التخطيط للبرمجة حيث يتعين على المبرمج تحديد اسم المسجل واسماء الذاكرة ومواقع المتغيرات فيها فكان امرا مرهقا فكان متعين على البرمجين تتطوير لغة لحل هذه المشكة ويتفرغ البرمجين التفكير في تصاميم العليا للبرنامج فتم تصميم اللغات المعروفة لغات عالية المستوى ولغات منخفضة المستوى

لغات منخفضة المستوى : وهي اللغات التي ترتبط بالهارد ويرحيث هي متعلقة الجزء الصلب من الحاسب الالي فتعمل على تحويل الاوامر الى اللغة التي تفهمها الالة المعروفة ب 1 – 0 وتسمى بلغت التجميع Assembly بحيث تجمع كل اللغات البرمجية وتترجمها الى لغة واحدة وهي اللغة التي تفهمها الالة

لغات مرتفعة المستوى : هي اللغات التي تكتب بلغة معروفة و مفهومة وغالبا تكون قريبة من اللغة الانجليزية بحيث يتم كتابة الاوامر و التعليمات التي نريد الكمبييوتر ان ينفذها بشكل معين ثم تحول هذه اللغات ذات مستوى على الى اللغات ذات مستوى منخفض لتحويلها الى لغة الالة ومثال على اللغات عالية المستوى هي لغة الفيجوال بيسك التي نتعلمها في هذا الكتاب و C# و الجافا

تاريخ تطول لغات البرمجة عالية المستوى

في عام *1958* تم تطوير لغة **Fortran** كلغة عامة ثم تم تطوير عدة لغات على مر السنين بعد ذلك ففي عام 1964 قام جون كينمي و تومس كارتز بتطوير لغة البيسك **Basic** ثم طورت لغة **ADA** من قبل وزارة الدفاع الامريكية عام *1970* بحيث يتم توجيه من خلالها الصواريخ ثم قام نيكوس ويرث بتطوير لغة باسكال **Pascal** في عام *1971* ثم في عام 1972 قام دينس ريتشي بتطوير لغة C الشهيرة التي انبثقت منها لغة **C++** عام *1986* التي اصبحت اللغة الام لباقي لغات البرمجة لانها تعتمد على اسلوب البرمجة الحديثة التي تعتمد على برمجة الكائنات الموجهه والتي تم من خلالها تطوير باقي اللغت الحديثة ك **Visual basic** من قبل مايكروسوفت عام *1991* ثم لغة **Java** من قبل مايكرو سيستمز عام *1995* ثم تطورمحموعة **لغات الفيجوال ستوديو** من شركة مايكروسوفت و التي ضمت **الاصدار السادس للغة الفيجوال بيسك** عام *1998* ثم طرحت مايكروسوفت

Visual Stodio.net عام *2000* وهو يجمع بين عدة لغات لتطوير تطبيقات متعددة من عدة لغات اشهرها VB.net التي نشرحها في هذا الكتاب و لغة C# الشهيرة التي راهنت شركة مايكروسوفت عليها بانه سوف تجذب ملايين المبرمجين ، و سوف نتناول مزايا لغات الدوت نت في السطور القادمة

ما هي الدوت نت

الدوت نت عبارة عن مجمع عن عدة لغات مطورة من قبل شركة مايكروسوفت و التي تضم J#.net، VB.net، C# والتي تتميز بعدة مزايا..... و قبل ان نوضح ما هي المزايا، تابع معي التطور الوظيفي للغات البرمجة لكي تستطيع فهم مزايا لغات الدوت نت

عندما تطورت لغت البرمجة كانت اللغت تعتمد على نظام البرمجة الخطية حيث يتم الاعتماد على برمجة على شكل اسطر كلاتي

```
C *** FORTRAN ***
integer i,j,k
write(5,50)
5 format(2X,6Hgood Day)
i=15
if (i.GT.10) goto 10
read(6,80) j
6 format(i4)
do 10 k=1,i-1,2
j=j+1
10 continue
```

كان هذا هو الشكل الاول للغات البرمجة وطبعا كان مناسبا للوقت الذي كانت اللغة طورت فيه لان كل تطور للغة من لغات البرمجة كانت تخدم الوقت الذي كانت فيه وتحل المشاكل التي كانت تواجه المرمجين في ذلك الوقت ولكن مع تعدد المشاكل تظهر تقنيات و لغات لحل هذه المشاكل فبعض هذه اللغة بدات تظهر مشكلة تكرار الاوامر بحيث انك تكتب المعادلة 50 مرة مما سبب ارهاق لكثير من البرمجين وكانت سبب في تطور لغات البرمجة لتبرمجة باسلوب الدالة الهيكلية Function Structur او الاجراء الهيكلي و التي كانت تعتمد على كتابة مجموعة من الامر مرة داخل اجراء وا دالة ثم يتم استدعاء هذا الامر من اي مكان فسهل ذلك المبرمجين في ذلك الاوان ولكن مع تقدم الوقت ظهرت مشكلة التعامل مع البرامج الكبيرة حيث ان هذا النمط لا تضاهي البرامج التي تتعامل مع البيئة الرسومية وذلك بسبب انك تكتب كود طويل جدا لكي تقوم بكتابة كود يطور نافذة ويندوز و تقوم هذه النافذة بنفيذ الاوامر الحسابية وغيرها مما ادى الى نفور كثير من البرمجين وتركهم مهنة البرمجة ولكن في وقت قصير جدا تم تطوير التقنية التي تحمل الكلمة السحريةي Visual التي تعمل ى على عمل واجهة رسومية بحيث ترسم واجهتك باستخدام الماوس مما ازاح الكثير من صدور المبرمجين بحيث اصبحت المهنة سهلة جدا .... ولكن كل عمل بشري يشوبه النقص فقد ظهرت مشكلة غريبة هي انك وانت تبرمج برامج كبيرة تكتشف انك تكرر نفس الكود عدة مرات فهذا يؤدي الى حدوث اخطاء كثيرة مما جعل الشركات تتنافس في تحسين التقنية الى ان ظهر تقنية معروفة بالCom التي سهلت على المبرمجين تطوير تطبيقاتهم وذلك بسبب استخدامها تقنية برمجة الكائنات الموجهة التي تحتوي على كائنات كل كائن به خصائص و اساليب ولكن ما عاب هذه ان هذه التقنية ادت الى حدوث عبوب رهيبة وحدث انهيارات كبيرة في البرامج بسبب انك يجب ان تعمل على تفيح الذاكرة من المتغيرات التي لا تحتاجها لان ترك المتغيرات بدون تفريغ يؤدي ملئ الذاكرة اكثر من اللازم فتؤدي الى انهيار كامل للبرامج و الكثير من المشاكل الاخرى لا يطول المقام لذكرها ... كل هذه المشاكل كانت سببا في ظهور تقنية الدوت نت التي نحن بصددها الان

ما هي مزايا الدوت نت ؟؟

1-تجميع اللغات حيث تستطيع تطوير تطبيق معين باستخدام عدة لغات بحيث يتم تجميع اللغات الى مترمجم وتحويلها للغة التجميع Assimbley عنطريق موحد الترجمة ان CLR Common langue runtime الذي يحتوي على وحدة تسمى IL intermediate langue بحيث يتم ترجمة فورية للغات البرمجة اثناء وقت التشغيل

2- العمل في عدة انظمة تشغيل لا يقتصر العمل على نظام التشغيل ويندوز ولكن يمكن العمل في نظام التشغيل لينوكس و ماكنتوش و الاندرويد

3 – وجود مكتبة مجمعةNetFramwark حيث ان هذه المكتبة تحتوي على عده اكواد توفر على المبرمجين الوقت في كتابة الاكواد وتوفر لهم كثير من الاكواد التي تساعدهم في تطوير تطبيقاتهم بدون الرجوع الى ملفات ال Regestry كما بها تقنية كما مكتبة الnetFramwar تعتبر المكتبة الام بحيث انك تستطيع برمجة برنامج واحد يعمل على عدة اماكن وعدة انظما من تطبيقات ويب او تطبيقات موبايل و اتطبيقات نوافذ

4- الترمجمة الفورية JIT

مما يزيد حبا في هذه المهنة انك وانت تكتب الكود المصدري تجد من يراجع خلفك ويصحح اخطائك وينبهك وذلك بفضل التقنية الترجمة الفورية بحيث انك تستطيع عمل برنامج وتجربه فتجد ان البرنامج يترجم لك الكوادك ويحولها الى لغة تناسب فهم الحاسب ويقوم بتنفذها واذا حدث اي خطا يقوم بتنبيهك عليه

5- **التجميع**

تستطيع عزيز القارئ ان تجبمع بين تطبيقات النوافذ و الويب و الموبايل في برنامج واجد يجمع بيهنا مجمع واحد بفضل تقنية Assemply

كما تستطيع ان تجمع ملفاتك و اكوادك وكل ما تحداجة في تطبيقك في مكان واحد دون الذهب و اللف والدوران

هذه اهم مزايا الدوت نت باختصار شديد جدا وحتى الا اطيل عليك في هذه المقدمة انتقل الان الى الباب الاول

الباب الاول

اساسيات في البرمجة

في هذا الباب سنتعرف على مبادئ البرمجة التي تشمل المتغيرات و جمل التدفق و الدوال و الاحداث و برمجة الكائنات الموجهة

ويشمل الباب الاول الفصول الاتية

- الفصل الاول : المتغيرات و الثوابت والعمليات الحسابية و المنطقية
- اللفصل الثاني : جمل التدفق الشرطية و التكرارية و المصفوفات
- الفصل الثالث : الدوال و الاجرائات
- الفصل الرابع : التصنيفات و الاحداث و تفويضات
- الفصل الخامس :برمجة الكائنات الموجهة ومحددات الوصول و الهياكل و الواجهات

البدايات المضيئة

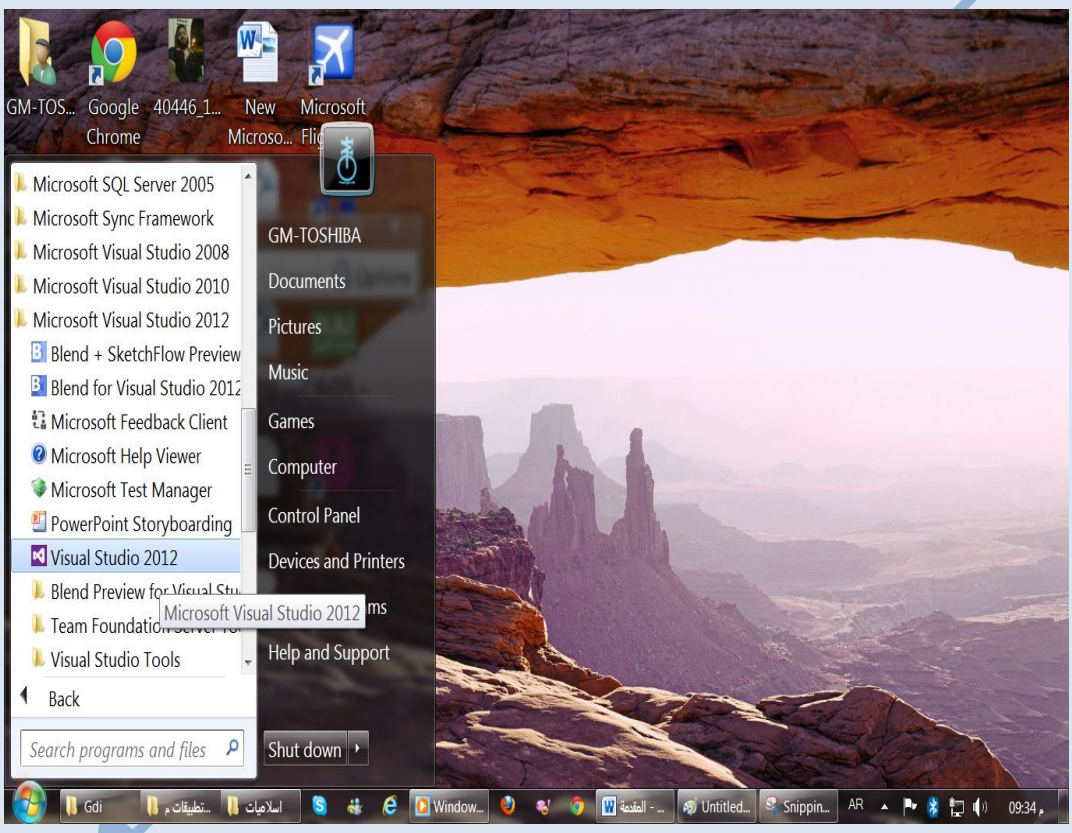
الفصل الاول

المتغيرات و الثوابت

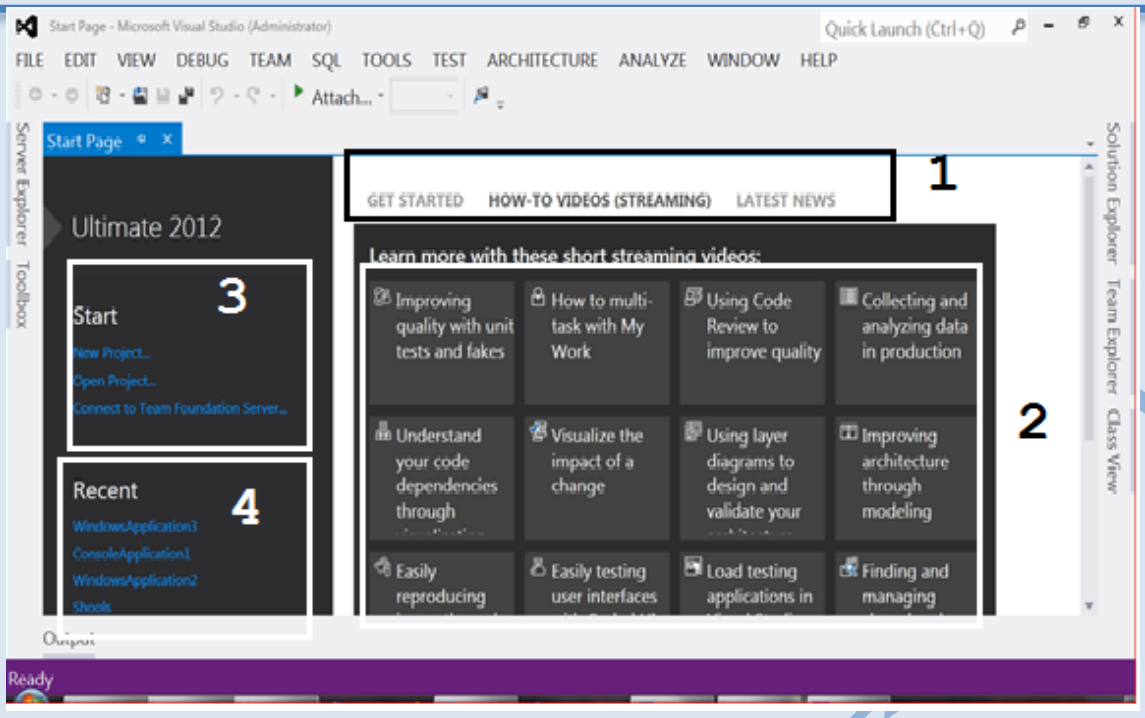
في هذا الفصل سنتناول المتغيرات و الثوابت و انواعها و طرق التحويل الانواع

نبذة عن بيئة البرمجة

قبل ان نبدا اريدك ان تتعرف على الشكل الخارجي للبرنامج الذي سوف نعمل به بعد ان تنصيب البرنامج قم بفتح البرنامج viaual studio 2012 من قائمة Start ثم All pogram افتح المجلد Microsoft Visual Studio 2012 قم بفتح البرنامج



سبدا بالتحميل فتجد واجهة البداية تظهر كالشكل الاتي



اذا امعنت النظر في هذه الشاشة فستجد انها مقسمة لاربع اجزاء وهي

الجزء 1 و 2 هما عبارة عن شاشة تتصل بالانترنت حيث انها تتصل بشكل مباشر بالموقع الاساسي لشركة مايكروسوفت وهو **msdn** فالقسم رقم 1 يضع لك اختيرات هل تريد تعلم عن طريق الفيديو فيكون الاختيار الافتراضي الذي نحن امامه وهو **how to –vedio** وهو قسم معروف في موقع الmsdn والقسم **Get standerd** فيكون التعليم بالمقالات المكتوبة في الموقع اما القسم الاخير وهو استخدام البحث داخر الموقع

اما القسم رقم 2 تابع للقسم رقم 1 بحيث يظهر في هذه الشاشة ملفات الفيديو التعليمي من موقع msdn **احب ان اوضح ان شكل هذه الشاشة مختلف بعض الشيء عن شاشات الاصدرات السابقة في 2010 و 2008 لان الاصدارات 2010 تشبه 2012 في بوابات البحث التي في قسم رقم 1 ولكن شكل القسم رقم2 مختلف بعض الشيء**

اما القسم 3 فهو قسم Start الخاص بالفيجوال ستوديوحيث به ثلاثة خيارات هي

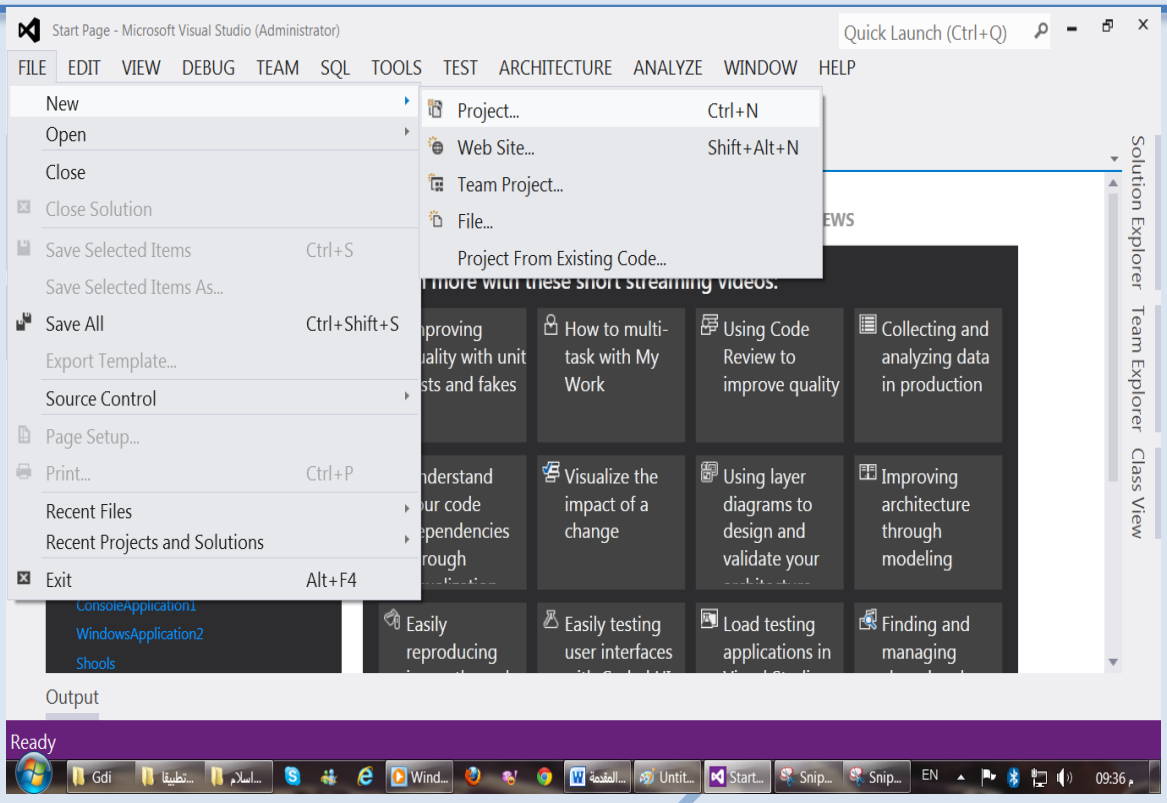
New project وهو فتح مشروع جديد

Open project وهو فتح مشروع كان موجود وتريد ان تكمل العمل فيه

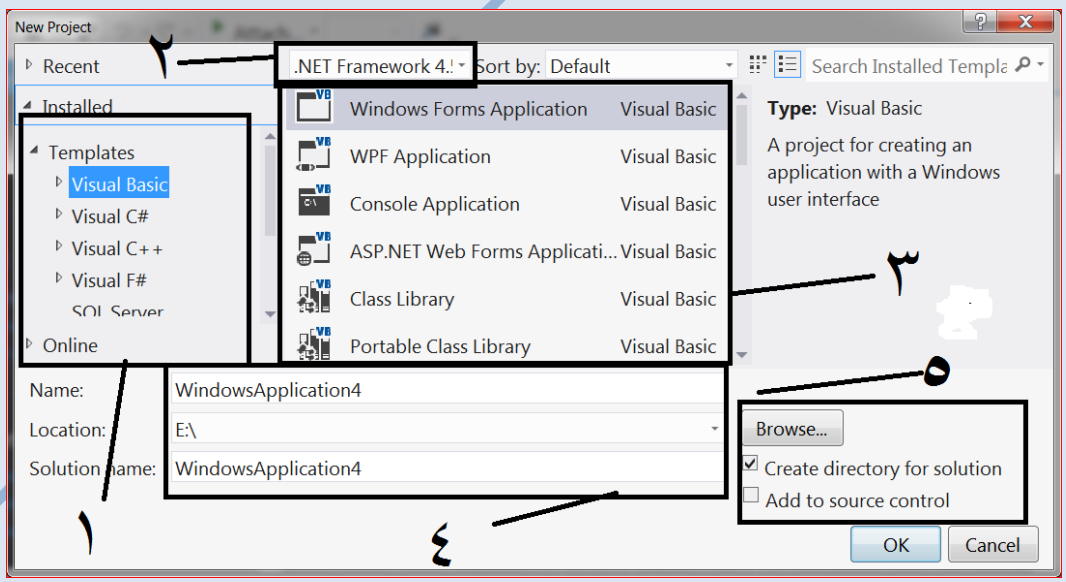
Contact to team foundation server : تستخدم في التعامل الشبكي مع فريق العمل

اما الشاشة رقم 4 فهي شاشة تحتفظ بالملفات التي فتحتها اخر مرة

نريد الان ان نقوم بفتح مشروع جديد من قائمة File افتح القائمة الفرعية new ثم انقر Project



سيظهر صندوق حوار



ستجد اني قمت بترقيم النافذة ليسهل عليك فهم محتوياتها

1	يحدد نوع اللغة التي تسخدمها في تطوير التطبيق كما تحدد نوع التطبيق هل يكون تطبيق Windows او web او هواتف كفية
2	يحدد اصدار المكتبة NetFrameWork الرئيسية التي من خلالها نقوم بتطوير تطبيقاتنا وطبعا بما اننا في اصدار 2012 فان المكتبة NetFrameWork من اصدار 4.5 وهو اخر اصدار

3	يحدد نوع التطبيق الذي نريد استخدامه وسوف نستعرضها لاحقا
4	يحدد مكان المشروع على القرص الصلب ونحدد اسم المشروع ايضا
5	هنا نحدد هل المشروع يضاف لاي مشروع سابق كنا نقوم بتطويره اوان يكون مشروع جديد مستقل حيث ان الاختيرا الاول ☑ Create directory for solution يحدد ان نفتح مشروع جديد بشكل مستقل اما الاختيار الثاني ☐ Add to source control نحدد اضافة المشروع لفريق العمل الذي تعمل معه على الشبكة او اضافته لمشروع كنت قد طورته وتريد ان تضيف هذا المشروع عليه

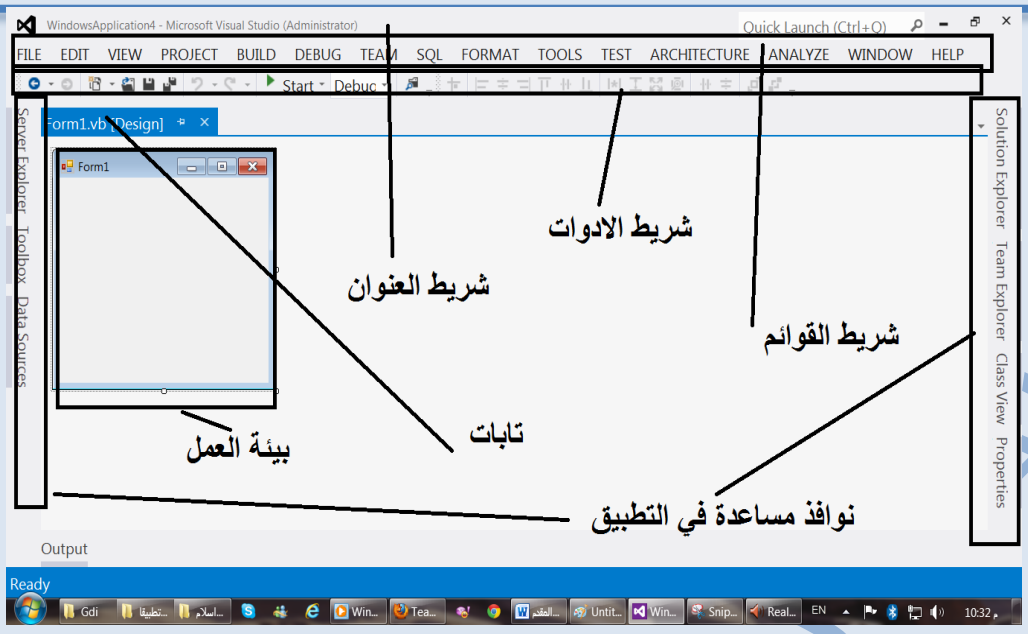
سوف نقوم الان بالتعرف على انواع المشروعات التي نريد استخدامها و التي اعطينها رقم 3 من النافذة السابقة

Windows Application	هي تطبيقات النوافذ التي نتستخدمها في اجهزتنا كاي تطبيق مثل تطبيقات الاوفيس او تطبيقات برامج المحاسبية وغيرها
WPF	هي تطبيقات نوافذية ولكن من نوع اخر يسمى **Windows Presentation Foundation**
Console Appiocation	تطبيقات من نوع Consol و التي شبه تطبيقات نظام التشغيل dos
Asp.net WebForm	هنا نقوم بعمل تطبيقات ويب ومواقع انترنت
Class library	هي عبارة عن مجمع كامل للاكواد حيث يسهل عليك الوصول اليها وسوف نتناولها في الفصل الخامس
Windows Control Liberary	هذا التطبيق الذي يتم تطوير ادوات التحكم لتطبيقات الويندوز وهو الذي سنقوم بشرحه في الفصل التاسع

كما يوجد العديد من التطبيقات الاخرى ولكننا سوف نستعرض فقط بعض منها في هذا الكتاب وهي تتمثل في Windows Application و Consol application و Class library و Windows Control Liberary

ففي هذا الفصل سوف نستعرض شكل البيئة الخارجي لتطبيقات الويندوز ثم سوف نتناولها بالتفاصيل في الفصل السادس في الباب السادس ان شاء الله

اختر تطبيقات النوافذ Wintows Application وتظهر لنا في الشكل الاتي



شريط العنوان

اذا كنت استخدمت تطبيقات الاوفيس ستجد ان البيئة مشابهة بها في توزيع الادوات ففي شريط العنوان يوضح اسم المشروع و به مكان للبحث بحيث تستطيع البحث عن اي عنصر في بيئة التطوير وهذه الخاصية لا توجد الا في الاصدار 2012 كما به ازرار الاغلاق و التصغير و التكبير

شريط القوائم

هذا الشريط يتم وضع قوائم متعددة المهام ومنها مشابه بتطبيقات الاوفيس مثل قائمة File وسوف سنتعرض بعض منها اثناء جولتنا في هذا الكتاب

شريط الادوات

كتطبيقات الاوفيس سوف تجد هناك ادوات لتسهل عليك الوصول اليه واستخدامها بدلا من البحث عنها في القوائم

التابات(اللسان)

هذه التابات ستساعدك للوصول لاقسام مشروع النشطة تخيل ان تستخدم اربع نوافذ في عملك ستجد انك تحتاج للوصل الي اقسام مشروع النشطة التي تستخدمها اثناء عملك فتسهل لك التبابت الوصول اليها كما

بيئة العمل

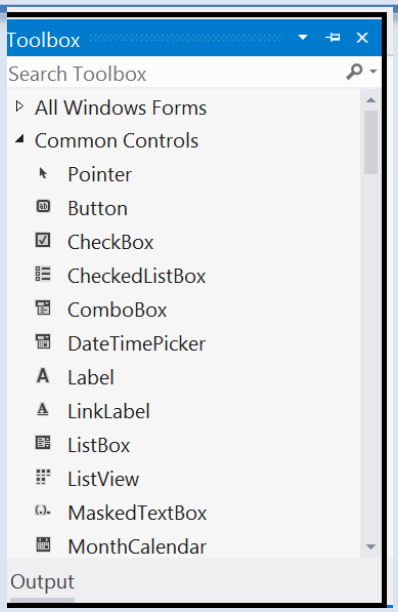
هي البيئة الاساسية التي نقوم بوضع ادواتنا عليه نقوم ببرمجتها

النوافذ المساعدة :

هي مجموعة من النوافذ التي تسخدمها اثناء تطويرك للمشروع واليك بعض هذه النوافذ

ToolBox

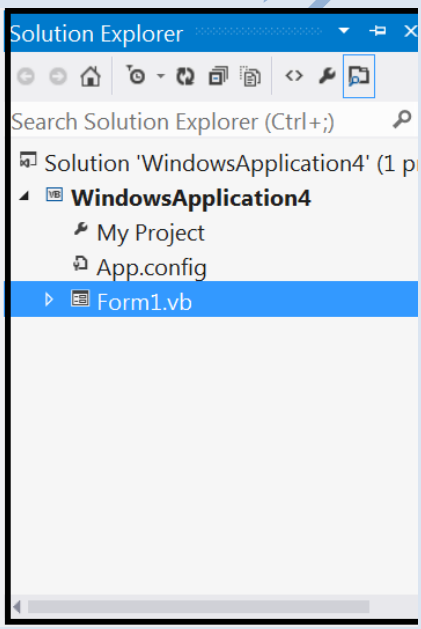
وهو عبارة ن نافذة تحمل ادوات العمل كالازرار و صندوق النصوص و العنوان وغيرها



ويمكن اظهارها عن طريق النقر على المفاتيح Ctrl +Alt+x او من النافذة

Selution Exprition

هي تستعرض مكونات المشروع اذا كان المشروع من عدة نوافذ او تصنيفات



Property

هذه النافد تستعرض خصائص النموذج و الادوات التي نستخدمها في تطبيقاتنا كلون الخلفية و لون الخط وغيرها من الخصائص وسوف نتناولها بالتفصيل في الفصل السادس

بعد هذه الجولة السريعة التي استعرضنها في بيئة تطبيقات النوافذ سوف نستعرضها في الفصل السادس بالتفصيل ان شاء الله سوف ننتقل الى مبادئ البرمجة وسوف نبدا بالمتغيرات وسوف نتعلم معا في هذا الفصل كيف تفتح مشروع جديد وكيف تغلقه وكيف تحظه وكيف تستعيده اثناء شرحنا في الكتاب انتقل معي الى الفقرة الثانية ونبدا بعون الله وضع اللبنة الاولى في البرمجة

المتغيرات

هي عبارة عن موقع معين في الذاكرة العشوائية RAM بحيث تخزن البيانات في مكان في الذاكرة وتقوم باستدعاء المخزن باسمه التي خزنته به .... ويتم الاعلان عنها في البرنامج بغلة الفيجوال بيسك باتباع القاعدة الاتية

```
Dim Variable Nmae As VariableType
```

Dim: من كلمات اللغة وهي تعلن عن وجود متغير في الذاكرة ويمكن استبادالها بكلمات اخرى ك privat, publc,static كما سنوضح لاحقا

Variable Nmae: هو اسم المتغير وهو الاسم الذي نستدعيه اثناء العمل وذلك بسبب تكتب برنامج كبير فيكون هناك كثير من الخازن في الذاكرة لذا كان لازما ان يميز كل مخزن عن اخر باسم يميزة عن غيره حتى يسهل الوصول اليه وعند تسمية المتغير يجب مراعاة الشروط الاتية :

1-ان لا يبدأ برقم فمثلا 45xsum او 23x هذا غير مسموح به ولكن x2 او sum23Ahmmad هذا يكون اسم متغير صحيح

2-و لا يحتوي على رموز كهذه الرموز ? # $ @ لكن لو استخدمت الرمز _ المعروف بـunder score فهو يحل محل المسافة لان المسافة غير مسموح بها ايضا فيكون المثال هكذا Name_of_studnet ولكن لو سميت المتغير بهذا الاسم Ahmmad@Ali فهي تسمية خاظئة و يحبذ التنوع بين الاحرف الكبيرة و الصغيرة كما يحبذ ان تستخدم اسماء متغيرات معقولة فمثلا لو كان المتغير يخص مرتبات الموظفين في قسم المبيعات فيكون Salere_Salary او SalaryOfSales لان الاسماء المبهمة والتي ليس لها معنى ستجعلك تعاني معها لانك لن تستطيع ان تتذكر ما هي مهمة هذا المتغير ولماذا قمت بالاعلان عنه وخاصة في البرامج الكبيرة و المعقدو وكثيرة الاومر

3-ان لا يكون من كلمات اللغة ك Dim , sub ,class

As Variable Type: كلمة As من كلمات اللغة و التي تخبر الكمبيوتر بنوع التغير الذي سنضعه مكان التعبير Variable type حيث ان نوع المتغير ما هو الا شفرة تحدد للكمبيوتر مكان تخزين المتغير و المساحة التي يجب ان يفردها للمتغير

و الجدول الاتي يحدد الانواع واحجام الذاكرة المطلوب تخزينها

النوع	الحجم	الاستخدام
Byte	1 Byte	يستخدم في تخزين اعداد صحيحة موجبة من 0 الى 255
sByte	1 Byte	يستخدم في تخزين اعداد صحيحة موجبة وسالبة من -128 الى 128
short	2 byte	يستخدم في تخزين عن صحيح قصير من 32768- الى 32768

ushort	2 byte	يستخدم في تخزين عن صحيح قصير موجب فقط من 0 الى 32768
Long	8 byte	يستخدم في تخزين عن صحيح طويلة 9E18- الى9E18
ulong	8 byte	يستخدم في تخزين عن صحيح طويل 9E18الى0
integr	4 byte	يستخدم في تخزين عن صحيح 2146483648- الى 2146483648
uinteger	4 byte	يستخدم في تخزين عن صحيح 0 الى 4294967295
Double	8 byte	يستخدم في تخزين ارقام حقيقية مع وجود رقم عشري لكن يخزن ارقام بحجم كبير 3.4E83 - الى 3.4E83
Single	4 byte	يستخدم في تخزين ارقام حقيقية مع وجود رقم عشري لكن يخزن ارقام بحجم صغير 1.7E803 - الى 1.7E803
Decemal	12 Byte	يستخدم في تخزين الرقم عشري
Date	4 Byte	يستخدم في تخزين التاريخ و الوقت
Boolen	2 Byte	يستخدم في تخزين متغيرين فقط هما True, False
Opject	4 Byte	هو النوع الاب لانه يتحمل كل انواع المتغيرات ويقوم بتحديد نوع نفسه من خلال البيانات التي يدخلها المبرمج عليه
Char	2 byte	هذا نوع حرفي لكنه يحمل حرف واحد فقط مثل "b"
String	على حسب الحرف	هذا النوع يخزن النصوص بحيث انه يخزن نصوص وكلمات واحرف و رموز مثل "رقم العملي# 1# هو / 01235689" فتجد ان النص بين علامتين التنصيص هو نص يخزن على داخل متغير من نوع String

وسميت المتغيرات بهذا الاسم لانه يمكنك ان تغير قيمتها من فترة الاخرى فيمكن ان تكتب فمتغير قيمة ثم تمسحها وتضع قيمة اخرى

وطرق الاعلان عن المتغيرات وتخصيص قيم للمتغيرات كثيرة ومتنوع ففي الكود الاتي واحدة من هذه الطرق بحيث انك تعلن عن متغير في سطر و تخصص له قيمة في سطر اخر

```
Dim x as integer

X = 5
```

وهذه طريقة اخرى للاعلان عن المتغيرات بحيث تعلن عن عدة متغيرات في وقت واحد ولكن يجب ان يكون بنفس النوع ولا يسمح بتخصيص اي قيمة الا لكل متغير على حدة كما هو موضح في الكود الاتي

```
Dim x,y,z As integer

X=5

Y = 10

Z = 2
```

ويمكن ان تعلن عن عدة متغيرات وتخصص لها قيمة اثناء الاعلان وتكون بانواع مختلفة كما في الكود الاتي

```
Dim x As ingeger = 2 , b As bouble = 1.55 , m As string = “What is your name ?”
```

ملحوظة هاااااامة

عند تخصيص اي قيمة للمتغيرات الرقمية فاننا نعطيه بشكل مباشر بدون وضع اقواس مثل x = 5 بيناما اذا كنت تريد تخصيص قيمة نصية يجب انك تعطيها القيمة بين علامتي التنصيص بهذا الشكل “القيمة “ مثل s =“Ali” بينما تخصص التاريخ للمتغير الذي من Date يكون بين العلامتين # سنة/يوم/شهر# مثل d = #02/29/2012#

واريد ان انبهك ان المتغيرات الرقمية التي لا تاخذ اي قيمة تكون قيمتها الافتراضية هي 0 و تكون القيمة الافتراضي للمتغيرات النصية تكون"" وهي تعني ان المتغير فارغ لا يوجد نص به واذا كان متغير منطقي تكون القيمة الافتراضية False

اذا كنت محتار ولا تستطيع فهم ما اهمية المتغيرات وما معنى حجز مخزن داخل الذاكرة دعني اعيدك لايام الطفولة الجملية

اتذكر عندما كنا في المدرسة وكنا نكره في مادة الرياضيات المسائل الكلامية مثل هذه المسئلة

باع سامر 154 قطع من الحلوى القطعة الواحدة ب12 جينهات ثم اشترى 160 قطع القطعة تساوي 8 جنيه فا هو الربح في هذه المعاملة

فكنا عندما نبدا في الحل يجب ان نحدد المعطيات وثم نحدد ما هو المطلوب ثم نبدا الحل بكتابة المعادلة ثم نكتب الارقام ثم نحدد الناتج النهائي و نكتب التمييز في اخر الحل

فدعني اخبرك ان هذه الخطوات تسمى الخوريزمات التي نستخدمها في البرمجة

فالمعطيات هي مدخلات النظام فتكون عبارة عن قيم المتغيرات ففي المسالة السابقة كنا نقول

ما باعه سامر = 154 قطعة

سعر القطعة الواحدة = 12 جينهات

وهكذه فهذه هي المدخلات المسئلة ففي البرمجة نقوم بتخزين هذه القيم في متغيرات

المطلوب هو المتغير الذي نستخدمه لاستخراج الناتج

ففي المسئلة القيمة ما ربحه سامر من المعاملة هو المتغير النهائي من حل المسئله فهو يعتبر مخراجات المسئلة

والحل هي المعادلات و الاجرائات و الاوامر التي نستخدمها

ففي المسلهة السابقة نقوم بعمل الاتي

1- اجمالي المبيعات هي : عدد القطع المباعة × سعر القطعة = 154×12=1848
2- اجمالي المشتريات هي : عدد القطع التي اشتراها × سعر شراء القطعة = 8×160 =1280
3- ما ربحه سامر هو اجمالي المبيعات – اجمالي المشتريات = 1848-1280 = 568

ولكن السؤال هنا : ما همية الذاكرة ولم نستخدم المتغيرات؟

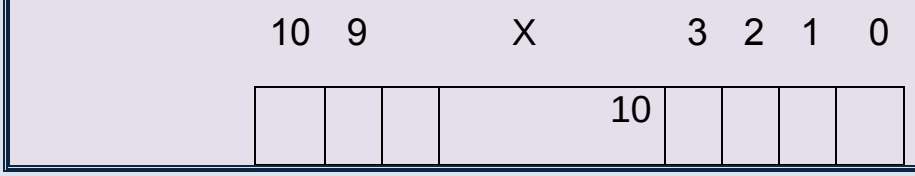
احب ان اذكرك باننا كنا نستخدم في هذه المسائل المسودة او الهومش لحساب القسمة المطولة او الضرب لاننا لا نستخدم الالة الحاسبة فمثلا كنا نستخدم المسودة لضرب رقم طويل مثل هذين الرقمين 154×12فكنت تستخدم مسودة جانبية لاجراء هذه الحسبة ....هذه المسودة هي الذاكرة ...فالذاكرة هي المسودة التي نضع فيها الارقام التي نحسبها بحيث انك تكتب مثلا الرقم 154 و الرقم 12 ثم تحسب المسالة وتكتب الحل في نهاية هذه الحسبة في مكان الحل الاساسي وعند الانتهاء من الحل تقوم بمسح المسودة استعدادا لكتابة ارقام اخرى ... هذا مثال تقريبي للذاكرة .. فالذاكرة تخزن البيانات استعدادا لاجراء الحسابات عليها وعند الانتهاء منها تمسح القيم التي بداخل الذاكرة وعند الانتهاء من البرنامج تمسح المتغيرات وتعود الذاكرة كما كانت قبل فتح البرنامج

وانا لا اقصد ان الذاكرة تخزن الارقام فقط وانما تخزن ايضا البيانات التي تخص النصوص و التاريخ و الوقت وغيرها من البيانات الي نستخدمها في برامجنا .

وتستطيع ان تتخيل ان الذاكرة عبارة عن عدة مربعات تخزن بداخلها القيم وكل مربع يمثل واحد بايت فلو عندنا الف واربعة وعشرون مربع يعني عندنا الف اربعة وعشرون بايت يعني واحد كيلو بايت ..... سوف ناخذ جزئ من الذاكرة ونقربه اكثر فيكن شكله هكذا

10 9 8 7 6 5 4 3 2 1 0

نحن لدينا احدى عشر مربع يعني معنا احدى عشر بايت فنفترض اننا اعلنا عن متغير اسمه X نوعه رقمي integer واعطيناه قيمة 10 فهذا اننا ستجز في الذاكرة اربعة مربعات لان هذا النوع يحجز 4 بايت فيكون شكله هكذا

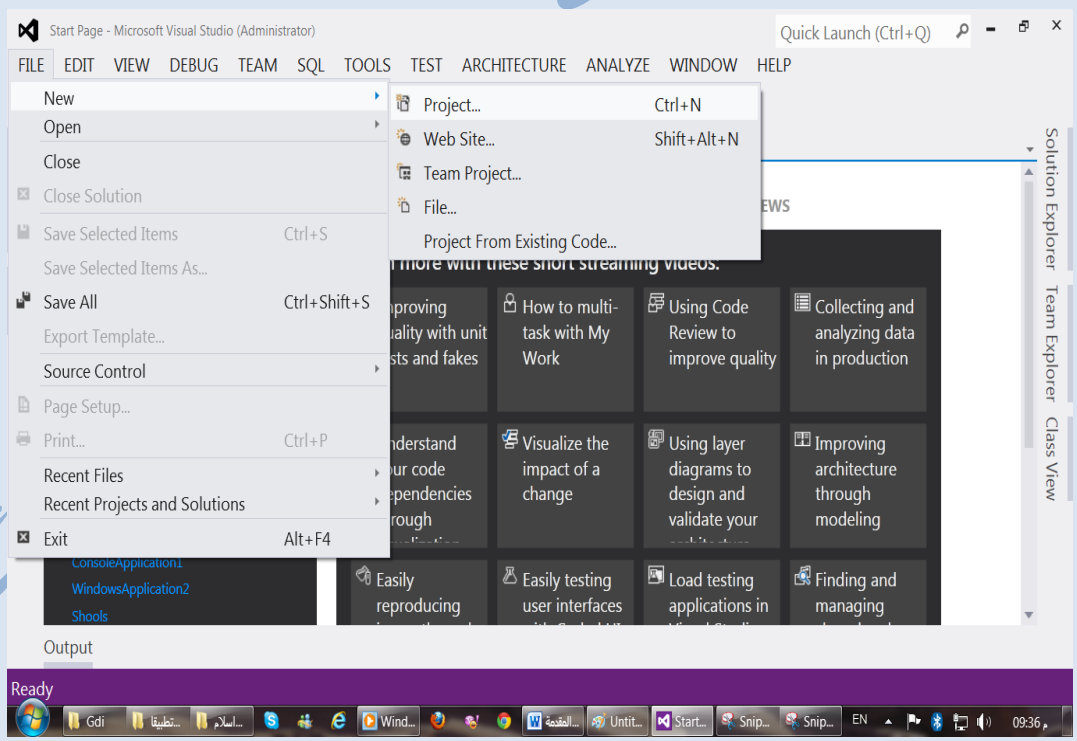


واريد ان اوضح لك عزيز القارئ ان تحديد مكان وحجم المخزن في الذاكرة يتولها الكمبيوتر بنفسه لان لغة الفيجوال بسيك هي التي تحدد مكان تخزين في الذاكرة النيابة عنك

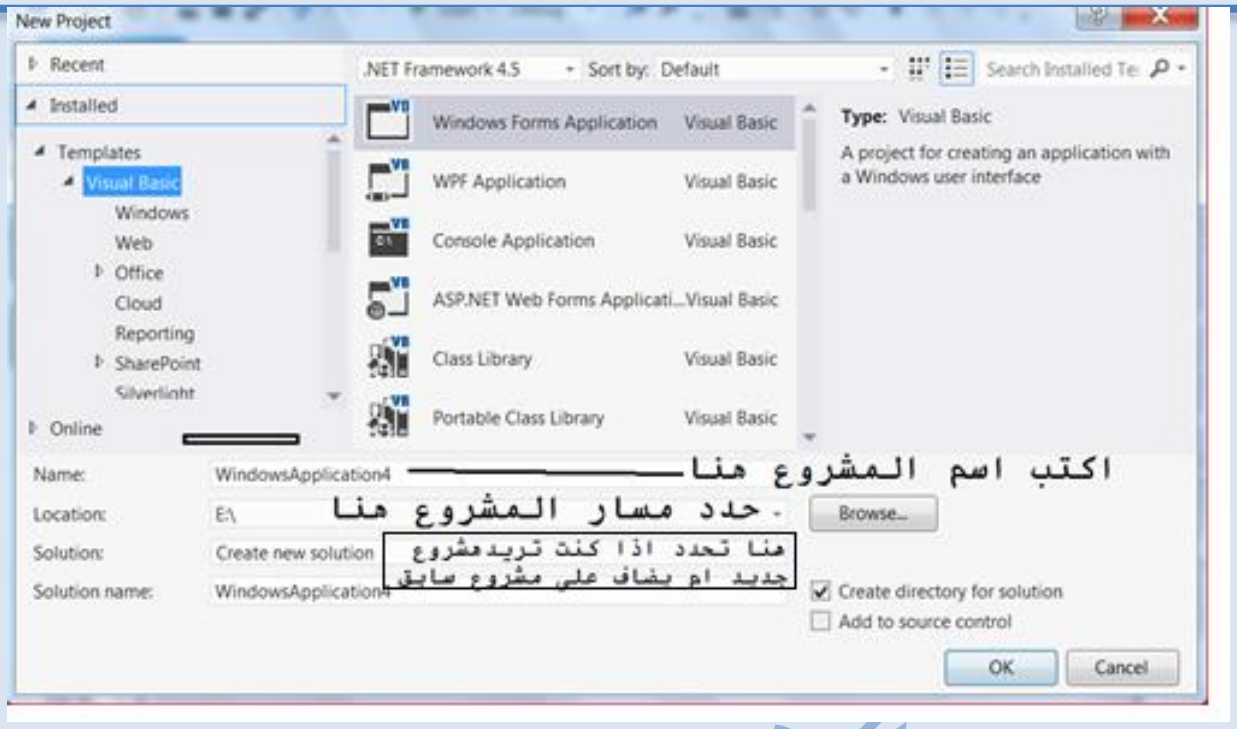
برنامجك الاول

تتعلم كيف تقوم بفتح مشروع جديد وكتابة كود فيه ثم تحفظه وتغلقه على انه اول مشروع لك ففي هذاه الفقر ستتعلم كيف تخزن البيانات في الذاكرة وكيف تستخرجها حيث انك تقوم بتخصيص بيانات للمتغير ثم تخرج هذا البيان للمستخدم

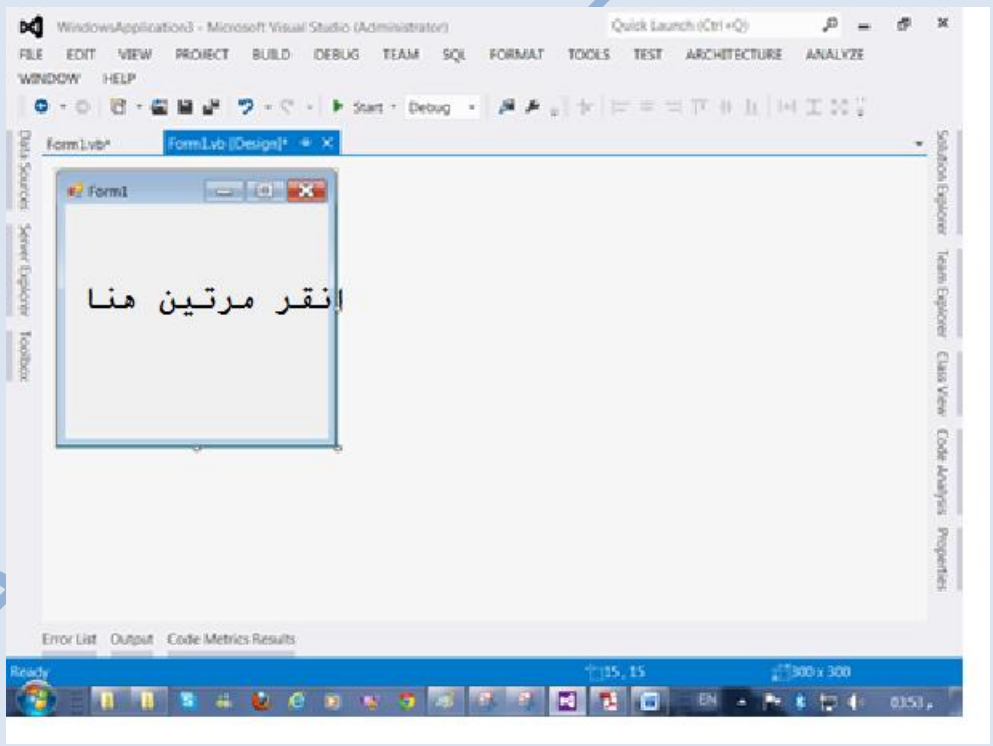
قم بفتح مشروع جديد وافتح مشروع Windows من قائمة File او من الجزء Start الذي شرحناه في اول هذا الفصل



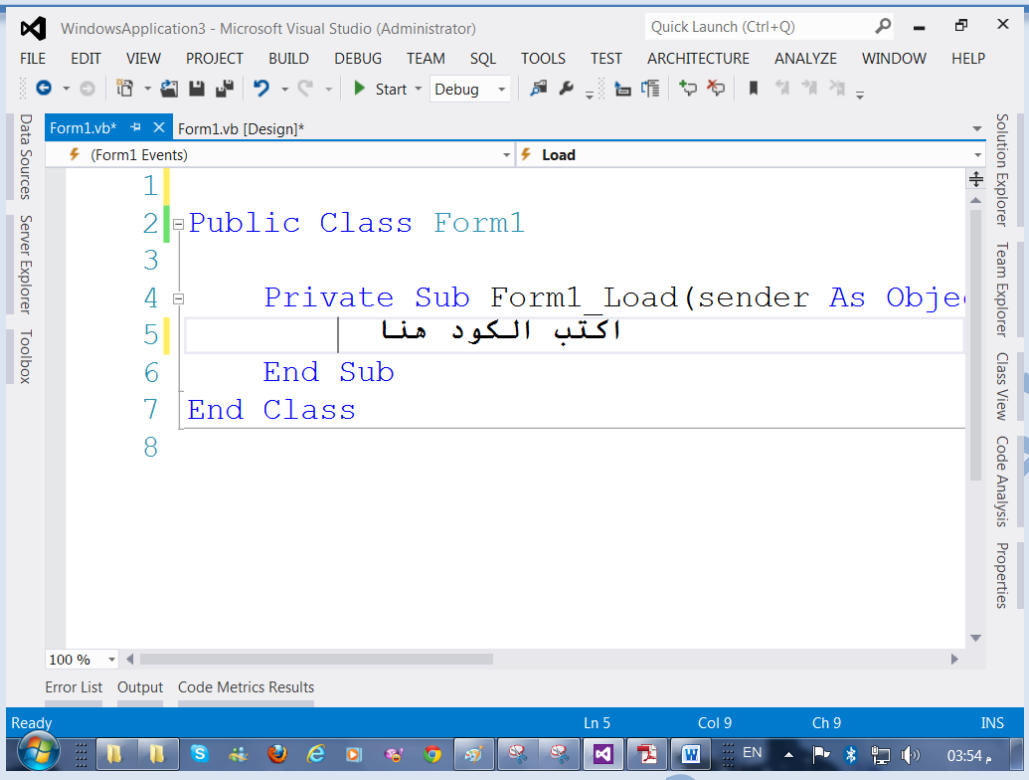
فيظهر لك صندوق النص الاتي



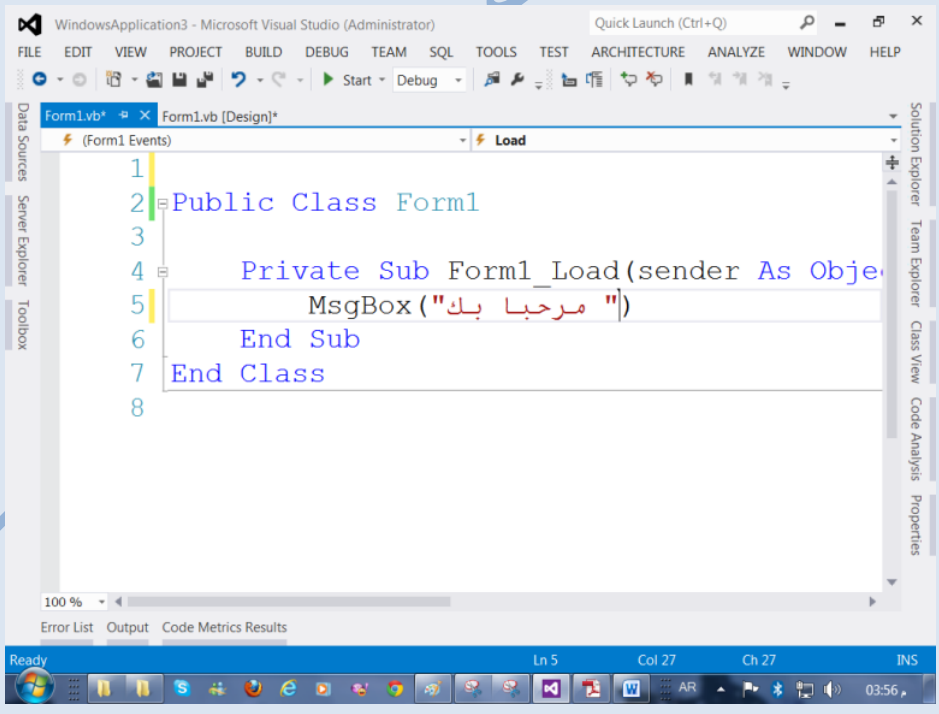
اكتب اسم المشروع وليكن Frist Poject وتستطيع ان تحدد مسار المشروع لانه الفيجوال ستوديو يقوم بحفظ المشروع بشكل متواصل حتى لا يضيع الكود الذي نكتبه



انقر على النموذج مرتين فتظهر لك شاشة الكود كما هو موضحة في الشكل الاتي

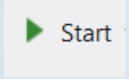


في المكان المشار اليه بكتابة الكود سوف نظهر لك رسالة ترحب بك فاكتب الكود الاتي في المكان المخصص لكتابة الكود كما في الشكل الاتي

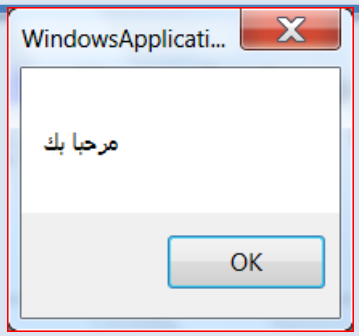


في الكود السابق قكتبنا الجملة

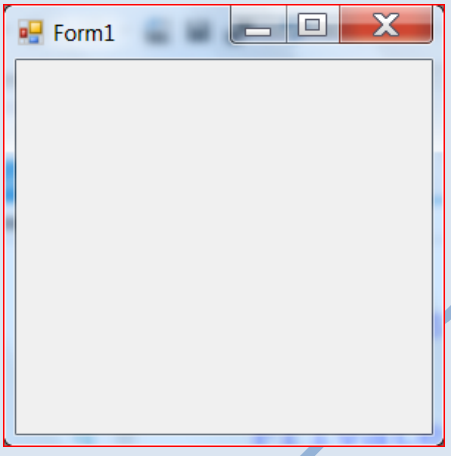
```
MsgBox("مرحبا بك ")
```

هذا الكود يظهر لك رسالة ترحب بك قم بالنقر على المفتاح F5 او على الزر 

من على شريط الادوات فتظهر لك الرسالة كما في الشكل الاتي



اذا نقرت على ok تظهر الشاشة كما في الشكل الاتي

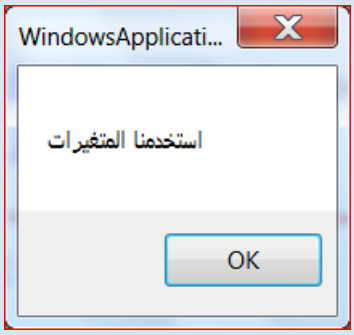


اذا اردت ايقاف البرنامج فمن الزر من شريط الادوات

ما رايك لو نعدل في شكل البرنامج فنستخدم المتغيرات سوف نعدل الكود كما هو موضح في الشكل الاتي

```
Dim s As String
s = " استخدمنا المتغيرات "
MsgBox(s)
```

لو شغنا البرنامج سوف تظهر لنا الرسالة الاتية

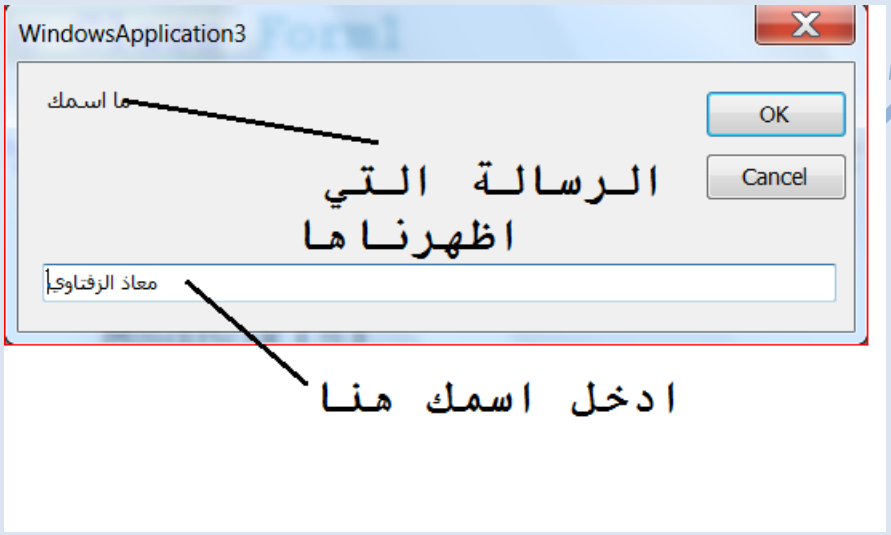


كما لاحظت اعلنا عن متغير نصي اسمه S وادخلنا عليه عبارة "استخدمنا متغيرات " ثم استخرجناها بوضع المتغير داخل الدالة msgbox

سوف نعدل في الكود ايضا بحيث يصبح البرنامج تحاوري بحيث انك تطلب من المستخدام ادخال اسمه ثم تظهر له رسالة تكتب اسمه .... لذا عدل الكود السابق ليكون هكذا

```
Dim s As String
s = InputBox("ما اسمك")
MsgBox(s)
```

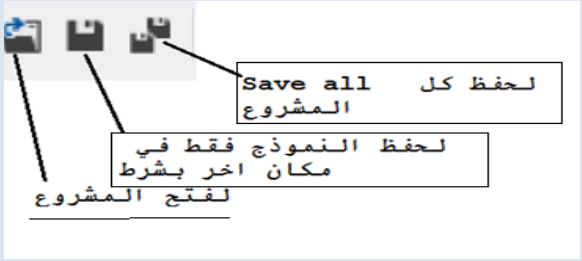
وعند التشغيل يظهر لك صندوق الادخال كما في الشكل الاتي ادخل فيه اسمك كما في الشكل الاتي



ثم اذا نقرنا على ok ستظهر لك رسالة تحمل اسمك الذي ادخلته

تلاحظ اننا استخدمنا متغيرات كوسيط لاخذ الاسم من الدالة inpotbox التي تظهر صندوق ياخذ القيمة التي يدخلها المستخدم وتكون هذه القيمة نصية ثم يعطيها للمتغير ومن المتغير ناخذها وضعها في الرسالة

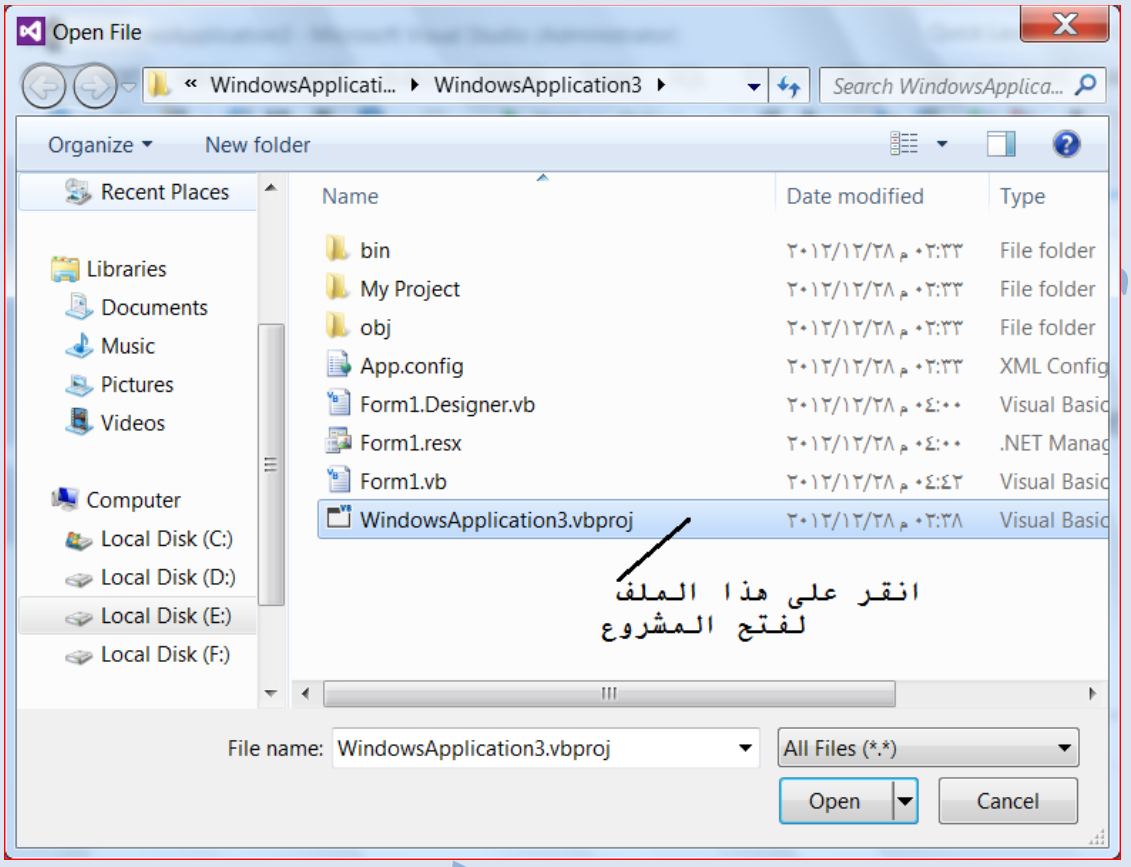
اذا اردت حفظ هذا البرنامج وانصحك ان تحتفظ به لاننا سوف نستعمله في هذا الفصل كثيرا فمن قائمة File انقر على Save All اذا كنت تريد حفظ المشروع كله او من على شريط الادوات انقر على اي زر من الازرار الاتية



ملحوظة هاااااااامة

في كل الاصدارات السابقة ستجد نفس الازرار ولكن شكل الايقونة مختلف

واذا اردت فتح المشروع مرة اخرى بعد ان اغلقته انقر على زر open او من القائمة File انقر على open فيظهر كل صندوق الحوار الاتي



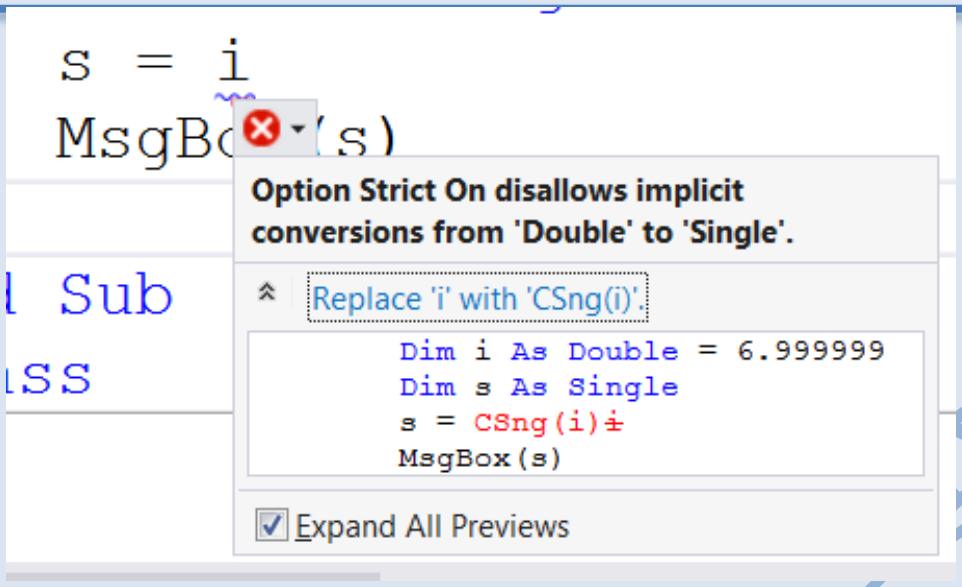
انقر على الزر open سيتم فتح المشروع

التحويل بين انواع المتغيرات

ما معنى تحويل ؟؟؟؟ ذكرنا في المتغيرات ان النوع يعني حجز مكان في الذاكرة ليتم وضع قيمة معينة ولكن ماذا لو انك اردت تخصيص قيمة لمتغير يحجز مساحة كبيرة مثل متغير من نوع long لمتغير يحجز مساحة صغيرة مثل متغير من نوع short او integer في هذه الحالة تحد ستجد ان المترجم يعترض على طريقة التخصيص .... لاحظ معي الكود الاتي

```
    Dim i As Double = 6.999999999999
        Dim s As Single
S = i
```

في الكود السابق خصصنا قيمة كبيرة لمتغير يحجز مساحة صغيرة في الذاكرة ستلاحظ ظهر علامة خطا ويعترض المترجم كما في الشكل الاتي



لو نقرت مرتين على المقترح الذي يعطيه لك سيتحول السطر الثالث الى الشكل الاتي

```
s = CSng(i)
```

تلاحظ ظهور دالة اسمها Csng هذه الدالة تعمل على تضييق حجم الرقم من حجم كبير الى حجم صغير يناسب حجم المتغير الصغير الذي هو من نوع Sng المخصص له القيمة و الجدول الاتي به دوال التخصيص الاتية

نوع التغير الذي تريد التحويل اليه	الدالة
Byte	Cbyte()
integer	Cint()
Long	CLng()
Short	Cshort()
string	Cstr()
Char	Cchar()
opject	Copj()
Date	Cdate
Decimal	Cdec()
Boolen	Cbool ()

ملحوظة هاااااااااامة

يجب ان تعلم ان علامة = علامة تخصيص وليس مساواه يعني ليست مثل علمة = التي نتسخدمها في الجبر بمان A=B اذا B=A هذا التعبير في البرمجة تعبير خاطئ التعبير الصحيح A = B يعني ان القيمة الموجودة في المتغير A ستمسح من الذاكرة وتستبدل بقيمة المتغير B

الثوابت

هو مثل المتغيرات مخزن داخل الذاكرة ولكنه يستخدم للقراءة فقط يعني اذا اعلنا عنه فاننا يجب ان نعطيه القيمة ولا يسمح ان نغيرها ابدا اثناء عملنا انظر معي الكود الاتي

Const x as ingeger = 5

كلمة Const هي كلمة اساسية تعلن عن وجود متغير في الذاكرة ولكن للقراءة فقط

اما X فهو اسم المخزن الثابت ويسمى بنفس شروط التسمية التي يسمى بها المتغير

اما As integer فهو مثل المتغير يعلن عن النوع او بمعنى ادق يعلن عن مكانه و حجمه في الذاكرة

اما 5= فهذه القيمة التي ندخلها على المتغير فيجب عليك اثناء الاعلان عن ثابت ان تضيف اليه القيمة التي تعطيه قيمته التي لا تتغير ابدا الا اذا اردت ان تغيرها فيجب ان تعود للمكان الذي اعلنت عنه فتغير قميته

السؤال الذي يطرحه كثير من القراء ما اهمية هذا المخزن الذي لا يسمح لاحد ان يغيره فاجيبك اتذكر الرقم الثابت لحساب نصف قطر الدائره ؟؟ انظهر معي المثال الاتي

تخيل اننا نريد حساب قطر الدائرة نقوم بكتابة الكود الاتي

```
Const pi As Single = 3.14
Dim r As single = 55
Dim x As Single
x = Pi*r*2
```

في الكود السابق نعلن عن ثابت وهو pi بحيث يحمل القيمة الثابتة لحساب مساحة الدائرة وهي 3.14 اما في السطر الثاني فهو المتغير x الذي يسيحمل قيمة محيط الدائرة ، اما في المعدلة التي في السطر الثالث فتعني الاتي

X هو المتغير الذي يستقبل محيط الدائرة وعلامة = تعني ان قيمة المعادلة(Pi*r*2) التي يمين العلامة ستخصص قيمتها في المتغير X

اما المعادلة Pi*r*2 فPi هي الثابت الذي اعلنا عنه وخصصنا له القيمة وناتج حساب الرقمين r هو نصف القطر قيمته 55 سم مضروب في 2 ....

وهذه المعادلة طبقا لقانون محيط الدائرة

حسننا ما رايك ان نطبق هذا الكود في البرنامج الذي كتبناه في اول فقرة و الذي استخدمنا فهي صندوق ادخال و الرسالة سوف ندخل طول نصف قطر الدائرة ثم نظهرها في الرسالة قم بفتح المشروع من الزر open من القائمة File او من الجزء Recent كما هو موضح في الشكل الاتي

ابحث عن المشروع الذي حفظته والذي كان اسمه Frist Poject الذي شرحناه في فقرة برنامجك الاول ثم عدل على الكود الذي كتبناه كما هو موضح في الشكل الاتي

```
Const pi As Single = 3.14
Dim r As Single = CSng(InputBox("ادخل مساحة الدائرة"))
Dim x As Single
x = pi * r * 2
MsgBox(x)
```

ثم انقر على F5 ستظهر لك صندوق الادخال يخبرك بادخال طول نصف قطر الدائرة فاذا ادخلت نص سوف يعترض ويخرج من البرنامج لذا يجب ان يكون الداخل رقم فاكتب الرقم ثم انقر ok ستظهر لك رسالة تخبرك بمحيط الدائرة

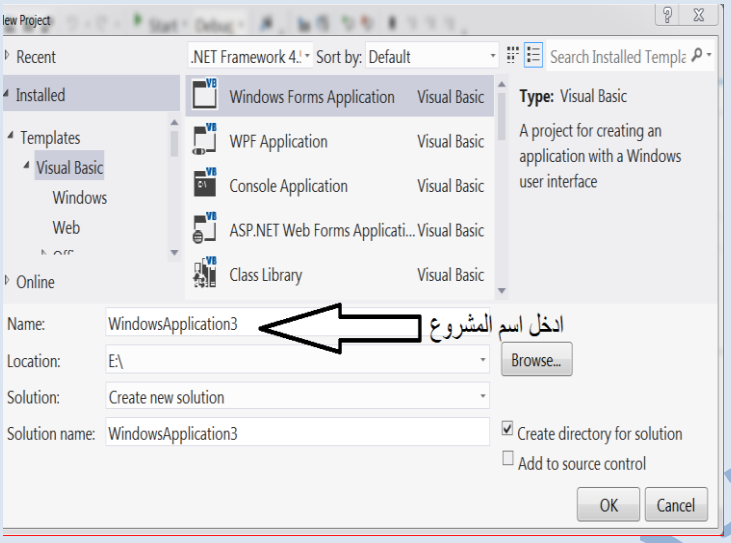
المتغيرات الساكنة

هي متغيرات خاصة يعلن عنها داخل الاجراء لكن هذه المتغيرات تحتفظ بقيمتها بعد الانتهاء من الاجراء حيث ان المتغيرات المعلنة داخل الاجراء وغير ساكنة تفقد قيمتها عنند الانتهاء من الاجراء ولكي يتم الاعلان عنها فاننا نتسخدم كلمة Static بدلا من كلمة Dimمثل

```
Static x as integer
```

ولكي نوضح الفرق بين المتغيرات الساكنة و المتغيرات الحالة اليك المثال التي

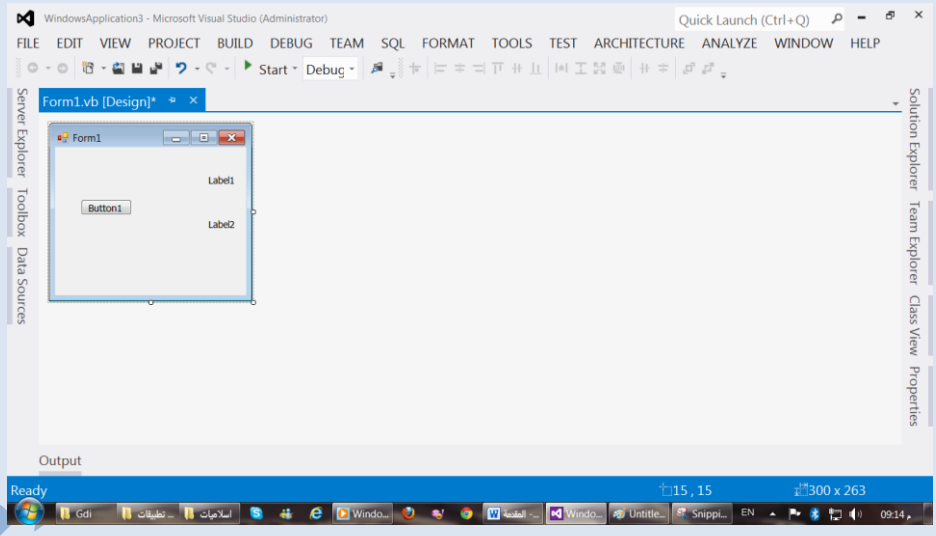
افتح مشروع جديد من الفيجول بيسك ويكون نوعه Windows Form كما في المثال الاتي



في هذا المشروع لن نستخدم صندوق الادخال و الرسالة لان التطبيقات لم تصمم لهاتين الكلمتين فقط لذا سنتقدم خطوة للامام ونتعلم كيف نضع ادواتنا على النموذج فمن صندوق الادوات ToolBox

سوف نضع ثلاثة ادوات زر button Button و صندوقين عنوان lable A Label

فيكون شكل النموذج كما في الصورة الاتية



ثم انقر على الزر مرتين فتظهر لك الشاشة الاتية

```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
                                الامر هنا يحدث عند تحميل البرنامج
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
                            الامر هنا يحدث عن
    End Sub                 النقر على الزر
End Class
```

اريدك ان تركز هنا قليلا تلاحظ وجود كودين الاول اذا كتبت عنده فان الامر ينفذ عند تحميل النموذج والكود الثاني عندما يقوم المستخدم بالنقر على الزر ولكن كيف عرفت ان هذا للتحميل وهذا عند النقر على الزر ؟؟؟؟

في الكود الاول ستلاحظ ملاحظتين الاولى ستجد كلمة Form_load وهي تعني عند تحميل النموذج و الملاحظة الثانية في اخر السطر ستجد كلمة handie mybase.load وعند كتابة الكود يجب ان لا يخرج عن نطاق الاجراء يعني لا تكتب الكود بعد كلمة End sub وهي تعني نهاية الاجراء المخصص لتنفيذ الاوامر عند تحميل النموذج

وكذلك عند النقر على الزر ستجد كلمة Button1_click وسوف نكتب الكود الاتي:

```
1 Dim x As Integer = 5
2  Static y As Integer = 4
3            x = x + 1
4        y = y + 1
 5       Label1.Text = x
 6       Label2.Text = y
```

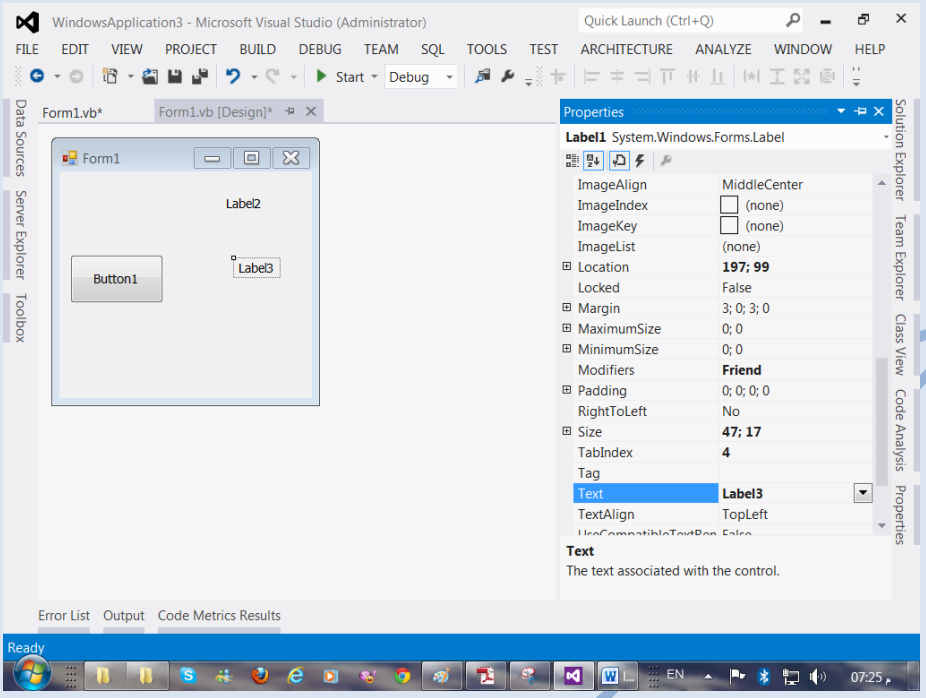
فيكون الكود على الشكل الاتي

```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim x As Integer = 5
        Static y As Integer = 4
        x = x + 1
        y = y + 1
        Label1.Text = x
        Label2.Text = y


    End Sub
```

- في السطر الاول اعلنا عن متغير x نوعه رقمي صحيح وخصصنا له القيمة 5 وهذا متغير حالة
- في السطر الثاني hاعلنا عن متغير y نوعه رقمي صحيح ورفقنا له قيم 4 خمسة وهذا متغير ساكن

- ثم قمنا باضافة قيمة رقم واحد على المتغير xبحيث تزيد قيمته التي يحتفظ بها ولتكن 5 فنضيف اليها 1 فتكون 6 و كذلك في المتغير y وذلك في السطرين الثالث و الرابع

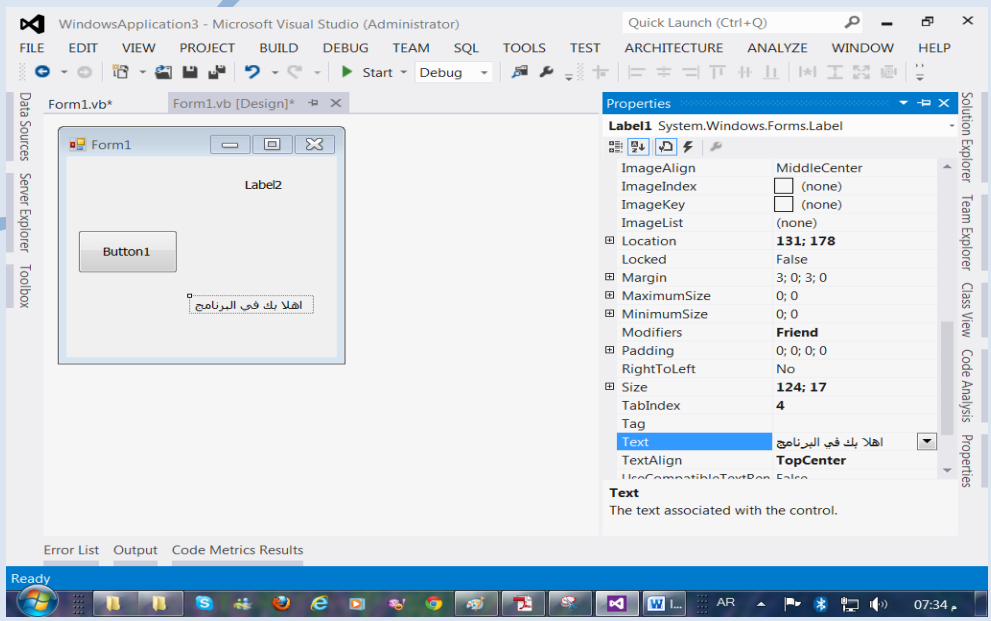
في السطرين الاخيرين سنتوقف قليلا لاننا يجب ان نتعلم كيف نتعامل مع خصائص الادوات ...قف على الاداة label1 ثم انقر على F4 فتظهر لك نافذة الخصائص



في الناحية اليمنى للشاشة تجد نافذة الخصائص وهي تحدد خصائص كل اداة اسمها او لون الخلفية او شكل الخط ولكن كل هذا سوف نتناول هذا في الفصل السادس بالتفصيل اريد فقط ان ناخذ خاصيتين هما name , text وهما خاصيتين موجودتين في كل ادوات تطبيقات النوافذ

الخاصية name هي تحدد اسم الاداة التي نستخدمها في كتابة الكود فالصندوق الاول اسمه lable1 واذا اردنا ان نغيره عن اسمه يجب اتباع الشروط الذي شرحنها في تسمية المتغيرات فالاسم # lable خطا و 1label ايضا خطا ولكن lbl1 صحيح

اما الخاصية Text وهي الخاصية تحتفظ بقيم نصية هذه القيم تقوم الاداة باظهارها قم بتغيير الخاصية Text للاداة lable1 فتكون "اهلا بك في البرنامج " ستلاحظ ظهور الاداة على الشكل الاتي



نعود الان للكود ففي السطرين الخامس و السادس خصصنا القيمة الناتجة من المتغير X للخاصية Text للاداة label1 و قيمة المتغير y للخاصية Text للاداة label2

وعند الضغط على الزر F5 فيتم تشغيل البرنامج عند الضغط على الزر نلاحظ ان صندوق العنوان الاول label 1 يظهر على قيمة 6 كلما نقرنا عدة مرات على الزر لان هذه القيمة من التغير X الذي هو متغير حالة فاننا نقوم بالنقر على ازر فيقوم باستدعاء الاوامر التي كتبنها وعند الانتهاء من تنفيذ الاوامر فان القيمة المخصصة للمتغير x تمسح من الذاكرة واذا استدعينا الامر فانه يخصص له الرقم الذي كتنباه في اول الاجراء اما الصندوق العنوان الثاني الذي خصصنا له قيمة المتغير y يتغير الرقم 6 ثم اذا نقرت على الزر مرة اخرى فيكون الرقم 7 ثم 8 ثم 9 وذلك ان المتغير يحتفظ بقيمته حتى لو تم الانتهاء من الاوامر ولكنه يفقد القيمة بعد اغلاق البرنامج تماما

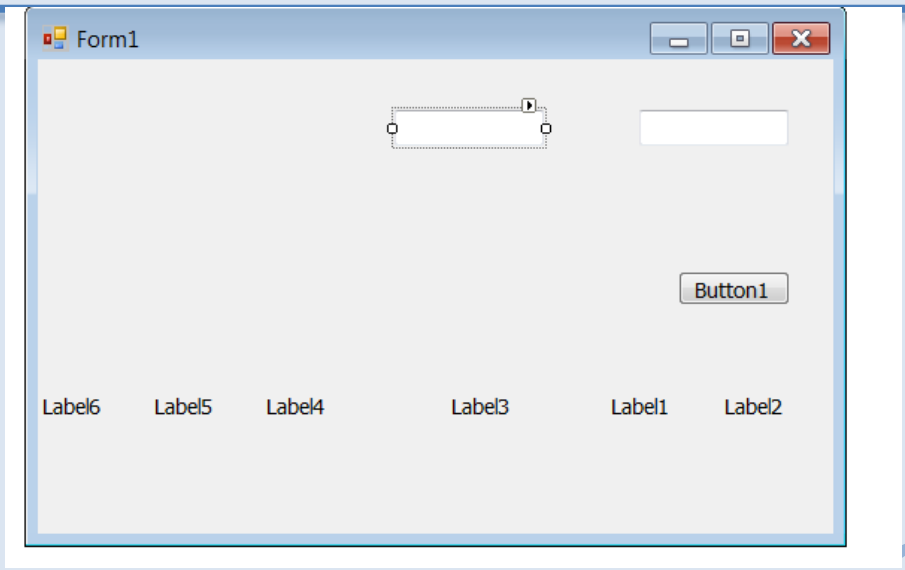
المعاملات الرقمية

المعاملت هي الاومر التي تستخدم في العمليات الحسابية و العمليات المنطقية كالجمع و الطرح و المقارنة اذا كان اكبر من او اصغر من او يساوي وليك جدول في العمليات الحسابية

الرمز	يستخدم في
+	يستخدم في عملية الجمع
-	يستخدم في عملية الطرح
*	يستخدم في الضرب
/	يستخدم في القسمة والناتج رقم صحيح و عشري
^	يستخدم في الاس
Mod	اظهار باقي القسمة بحيث يكون الناتج من قسمة 2 mod 3 يكون 1
/	يستخدم في القسمة الصحيح بحيث يكون الناتج رقم صحيح وليس عشري

مثال :

قم بعمل مشروع جديد وسمه Account ثم اضف زرButton و سبعة صناديق عناوين lable و صندوقي نصTextBox فيكون شكل النموذج كلاتي



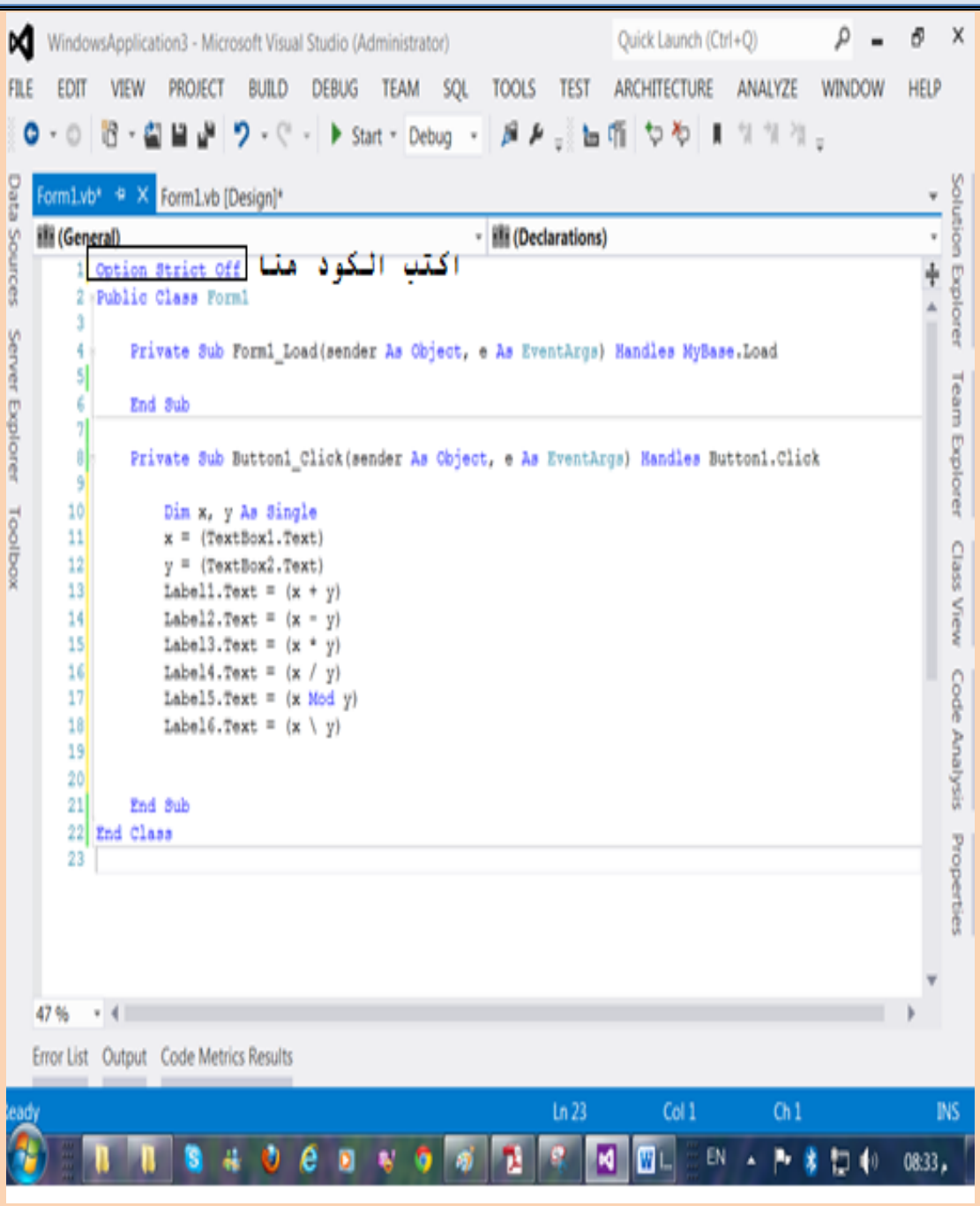
ثم نقوم بكتابة الكود في اجراء الخاص بالحدث Button_ Click ويكون الكود كلاتي

```
Dim x, y As Single
        x = CSng(TextBox1.Text)
        y = CSng(TextBox2.Text)
        Label1.Text = (x + y).ToString
        Label2.Text = (x - y).ToString
        Label3.Text = (x * y).ToString
        Label4.Text = (x / y).ToString
        Label5.Text = (x Mod y).ToString
        Label6.Text = (x \ y).ToString
```

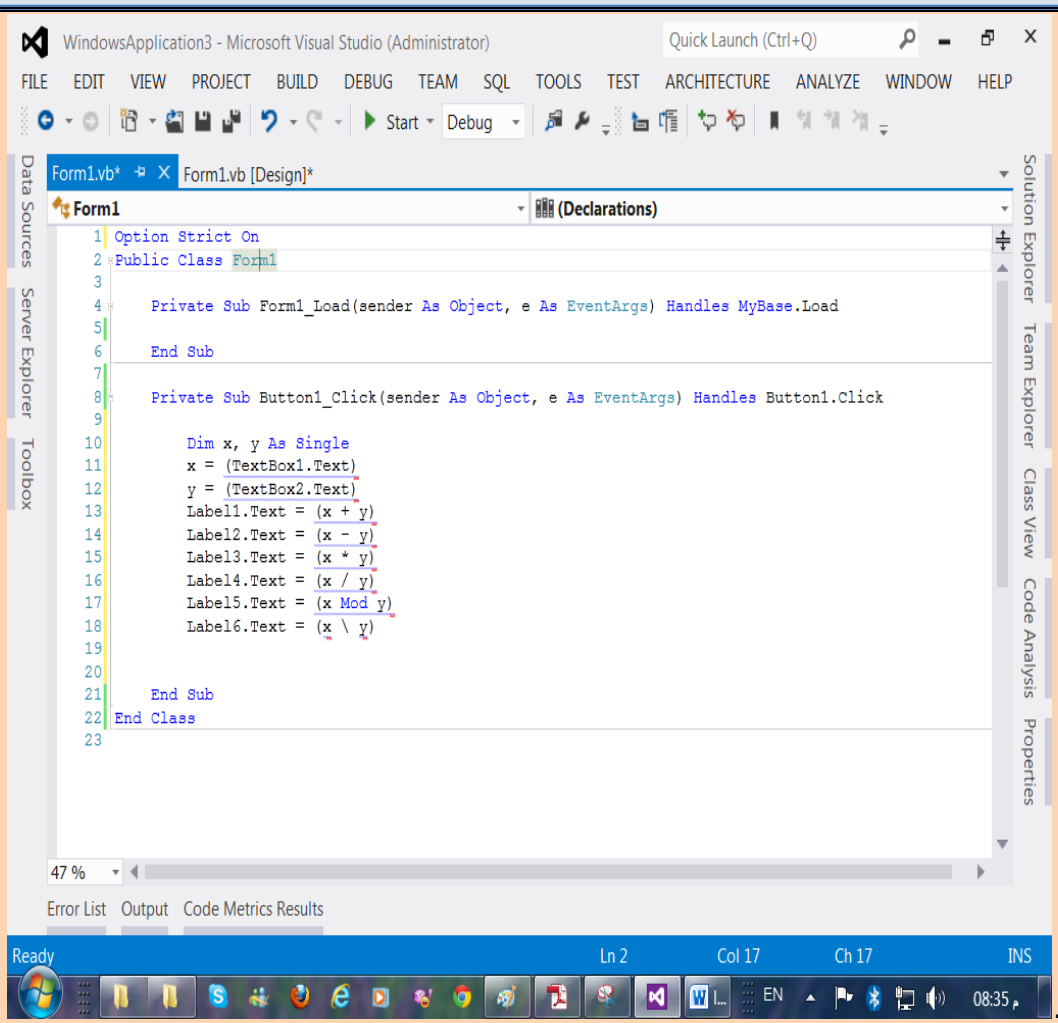
الكود السابق اعلنا عن متغيرين من نوع Single هما X,y بحيث اننا سنستخدم المتغيرين لتخزين القيم التي يدخلها المستخدم من صندوقين TextBox وهو صندوق يستقبل اي قيمة يدخلها المستخدم فيخزن هذه القيمة على الخاصية Text ونحن نعرف ان هذه الخاصية مشتركة في كل الادوات وهي خاصية نصية تخزن القيم بها فقمنا باخذ القيمة التي يدخال المستخدم من الصندوقين Textbox1 , textbox2 على المتغيرين x,y ثم قمنا بعمل عمليات حسابية على القيمتين المخزنتين في المتغيران X,y بحيث اننا اظهرناها على خاصية text لصندوق العنوان label1 ناتج عملية الجمع بين المتغيرين و في الlabel2 ناتج عملة الطرح و في label3ناتج عملية الضرب و label4 حاصل عملية الفسمة و label5 عملية باقي القسمة label6 حاصل عملية قسمة الرقم الصحيح

ستلاحظ في تخصيص قيم للمتغيرين x,y استخدمنا دالة التحويل Csng() ولكل خاصية Text في اي label ستجدنا نستخدام الدالة Tostring و التي تشبه الدالة Cstr() هذه الدوال المستخدمة في التحويل بين المتغيرات

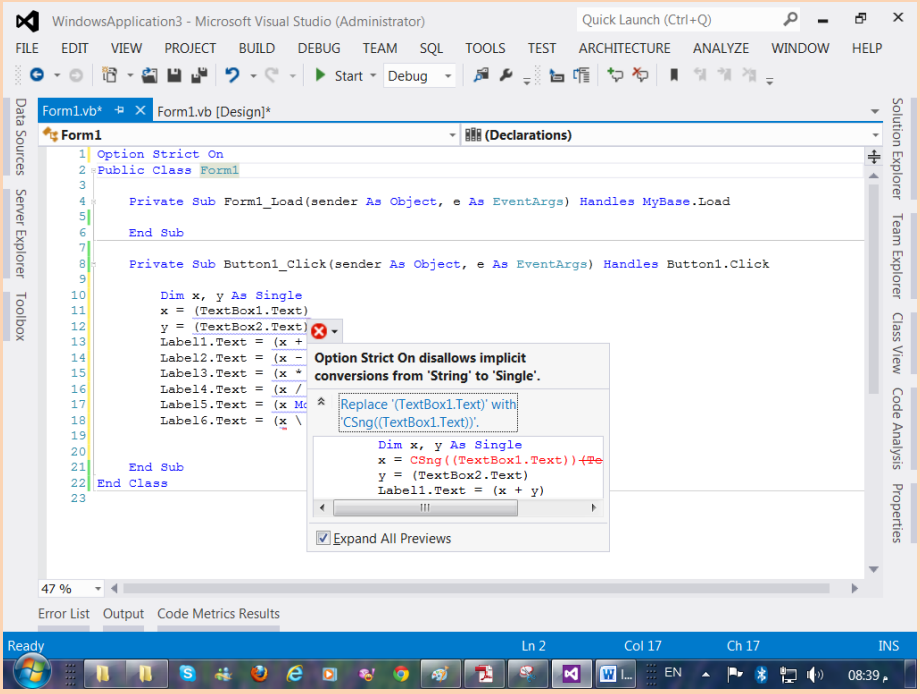
نستطيع ان نستغني عن دوال التحويل بين المتغيرات ولا نحتاج اليها وذلك لو اضفنا الجملة الاتية option stic off في اعلى الكود كما في الشكل الاتي



هذه الجملة تعني انك سوف تستغني عن جمل التحويل كما تلاحظ في الكود الذي كتبته في الشكل السابق فاني كتبت نفس الكود السابق الذي شرحته ولكن قمت بمسح دوال المسئولة عن تحويل المتغيرات ماذا لو عدلت الكود لكي يكون option stric on كما هو في الكود الاتي

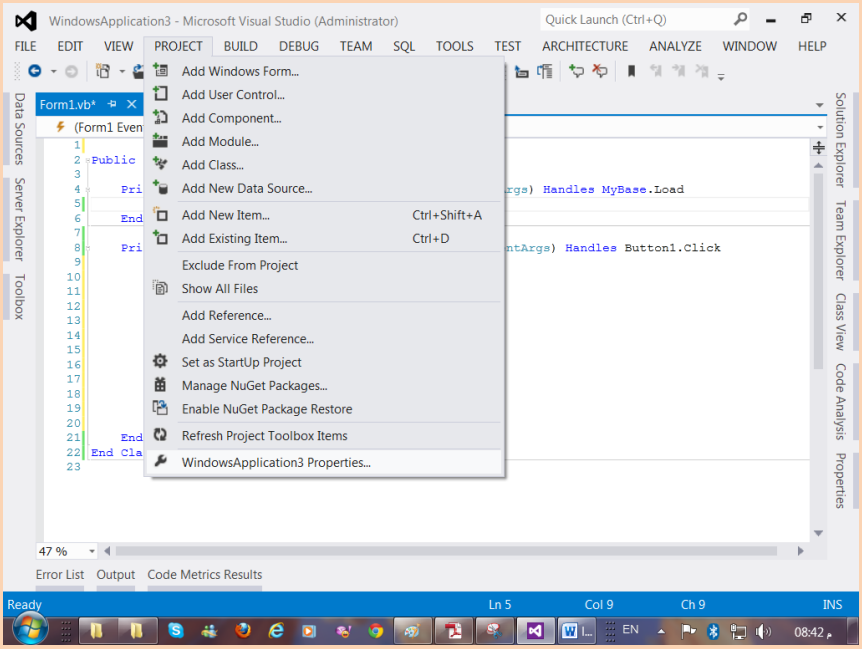


ستلاحظ ظهور خط ازرق تحت كل متغيرات تعني ان هناك خطا ترجمة واذ وقفت تحت النقطة الحمراء التي تحت كل متغير ستلاحظ ظهور رسالة تطلب منك ادخال دالة التحويل كما في الشكل الاتي



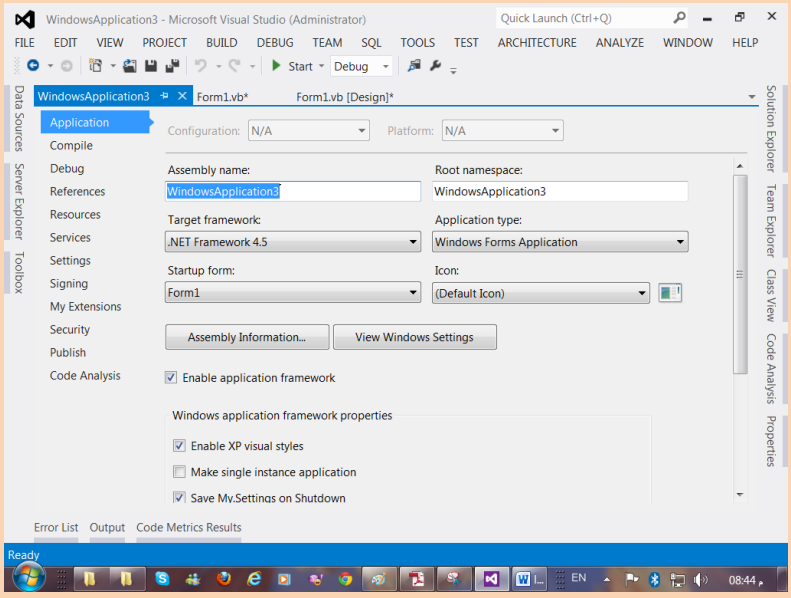
واذا كنت تريد ان تستغني عن دوال التحويل طوال عملك في المشروع لانك كتب الجملة option stric سوا كنت تريدها on او off فسوف تجد ان هذه الجملة مخصصة فقط في هذا النموذج دون غيره فافترض ان لدينا 10 نماذج ولا نريد ان نكتب هذه الجملة في كل نموذج فنتبع الخطوات الاتية

من القائمة project اختر اخر خيار وهو property كم هو موضح في الشكل الاتي



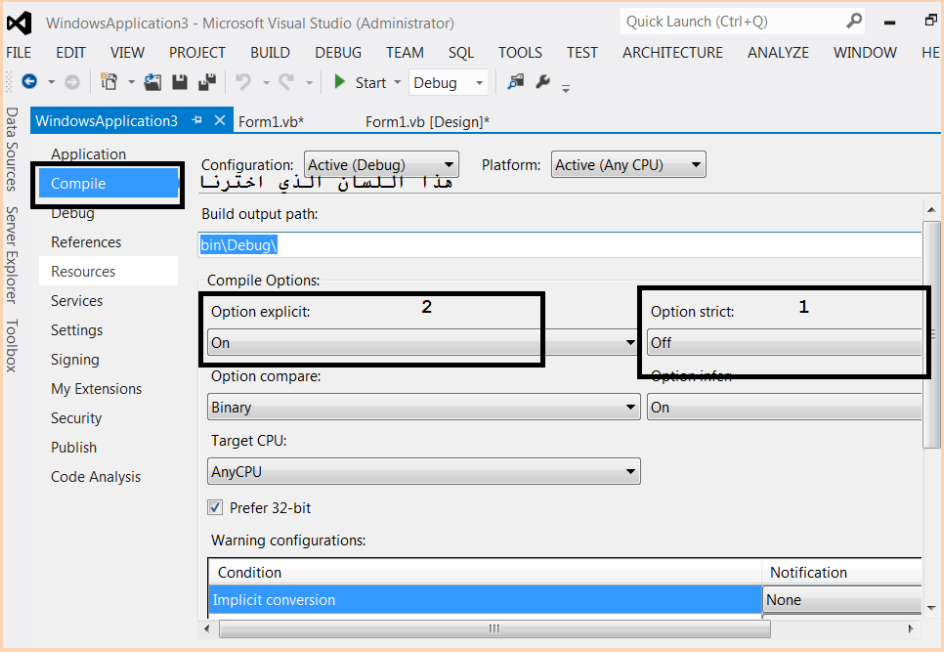
الخيار الاخير في القائمة project هو اسم المشروع ثم كلمة properties

فتظهر لك الشاشة الاتية



هذه الشاشة سوف نتعرف عليها طول رحلتنا في هذه السلسلة و الان اريدك ان تفتح اللسان Compil

فتظهر لنا الشاشة الاتية



في الشكل السابق تلاحظ اني وضعت ثلاثة مربعات

مربع على السان الذي يفتح هذه الشاشة وهو compile حيث ان هذه النافذة مقسمة لعدة اقسام وسوف نختر قسم الثاني وهو compile

اما المربعين الاتيين هما

1 هذا المربع يحيط بالكلمة option stric والاختيار الافتراضي هو off يعني انك تستغني عن دوال التحويل بين المتغيرات ولكن اذا اردت ان تستخدمه قم تغييرها الى on

2 المربع الثاني يحيط كلمة اخرى هي option explicit وهي تتيح لك ان تستخدم المتغيرات دون الاعلان عنها يعني انك تكبت المتغير بشكل مباشر وتخصص له قيمة فاذا كان الاختيار off فانك تكتب x=5 بدون ان تستخدم كلمة الاعلان عن المتغير x يعني لا تكتب جملة الاعلان وهي dim x as integer ثم تكتب x=5 وانما تكتب المتغير بشكل مباشر

ولكن الاختيار on يعني انك يجب ان تعلن عن متغيرات قبل استخدامها

اكواد تخصيص مختصرة

واليك جدول اكواد التخصيص المختصرة حيث توفر عليك كتابة بعض الاكود بصورة مختصرة

الرمز	يستخدم في
x+=5	اضافة رقم 5 للمتغير x
x-=5	طرح رقم 5 للمتغير x
x*=5	ضرب رقم 5 للمتغير x
x/=5	قسمة رقم 5 للمتغير x
x^=5	قيمة اس رقم 5 للمتغير x

الكود X+=5 هو مختصر للكود X = X+5 بمعنى لو ان قيمة X تساوي 5 ثم اضفت وتريد ان تضيف الى قيمته رقم 5 فانك تستدعي المتغير X وتضيف اليه رقم 5 فتصبق قيمة x تساوي 10

المعاملات العلائقية

المعاملات العلائقية هي تستخدم في اختبار العلاقة بين طرفين بحيث انك تختبر هل القيمة التي في المتغير A تساوي ام الكبر ام اصغر من القيمة التي في المتغير B

واليك عزيز القارئ الرموز المتسخدمة في المعاملات العلائقية

الرمز	يستخدم في
>	اذا كان الطرف الايمن اصغر من الايسر
<	اذا كان الطرف الايمن اكبر من الايسر
=	اذا كان الطرف الايمن يساوي الايسر
<=	اذا كان الطرف الايمن اكبرمن او يساوي الايسر
>=	اذا كان الطرف الايمن اصغرمن او يساوي الايسر
<>	اذا كان الطرف الايمن لا يساوي الايسر

لنفترض ان لدينا متغيرين هما A و B فنقوم باختبارا هما فمعنى ان A<B اننا نختبر هل قيمة المتغير A اصغر من القيمة التي في المتغير B وهكذا

المعاملات المنتطقية

تخيل لو ان تختبر بين علاقتين : العلاقة الاولى هل قيمة المتغير A تساوي قيمة المتغير B ولنرمز اليها بالعلاقة X و العلاقة الثانية هي هل قيمة المتغير C يساوي قيمة المتغير D ولنرمز اليها بالعلاقة Yفاذا كان المتغير A=B تصبح قيمة العلاقة x تساوي true وكذلك في العلاقة Y تصبح true اذا كانت C =D فالمعاملات المنتطقية تعقد المقارنة بين العلاقتين x , y كما هو موضح في الشكل الاتي

الرمز	يستخدم في
And	اذا كان العلاقتين x,y تساوي true فيكون الناتج true
Or	اذا كان احد العلاقتين x,y تساوي true او كلاهاما فيكون الناتج true
Not	يعكس العمليات اذا العلاقتين x,y او احدهما تساوي true الى False و العكس
xor	لربط العلاقتين x,y فاذا كان احدهم يساوي true فان الناتج يكون true و اذا كان كلاهما يساوي true او false فانا يعطي قيمة False

بالنسبة لاختبار العلاقات سوف نشرحها بالتفصيل في الفصل الثاني في جمل الشرط

تنفيذ الاولويات

درسنا ونحن في المرحلة الاعدادية كيفية الحسابات الجبرية فكنا عندما تصادفنا مسئلة على الشكل الاتي X = 5*6+(2+7^3)/8 فكنا تتبع خطوات حسابية حتى نحل هذه المسائل فكنا نفك الاقواس ثم نحل الاسس وهكذا ففي الجدول الاتي خطوات الحسابات الجبرية التي نستخدمها في الحساب اليدوي وتستخدم في الحاسب الالي بنفس الاولية بحيث يحسبها الكمبيوتر بنفس الحسابات التي نحسبها بايدينا

ترتيب الاولية	المعامل
1	فك الاقواس
2	حساب الاس
3	فك الاشارة السالبة
4	حساب الضرب و القسمة
5	القسمة و الصحيحة
6	باقي القسمة
7	حساب الجمع والطرح

المعاملات النصية

هو معامل واحد فقط وهو + او & بحيث انه يجمع بين نصين بعني لو لدينا الكود الاتي

```
Dim s as string = “hi”
Dim s2 as string = “Moaz”
```

ونريد ان نجبع بين النصين فنكتب الكود الاتي

```
Dim ss as string = s+s2
```

يعني ان نجمع بين كلمتين ولكن ستجد ان الكلمة تصبح هكذا hiMoaz لذي سوف نعدل الكود ليكون كلاتي

```
Dim ss as string = s+” ”+s2
```

فيكون الناتج hi Moaz بحيث اننا اعطينها مسافة بين الكلمتين حيث ان المسافة تعتبر حرف من ضم الحروف

وبذلك نكون قد انتهينا من هذا الفصل وننتقل الى الفصل الثاني الذي بها جمل من اهم الجمل المستخدمة في البرمجة الى هناك

الفصل الثاني

جمل التحكم بالتدفق و المصفوفات

في هذا الفصل ستتعلم كيف نتشئ برنامج الاول بشكل حقيقي فهذا الفصل سنتعلم اهم شيء في لغة البرمجة وهي جمل التحكم التي تعتبر العصب الاساس في لغة البرمجة فبدون هذه الجمل تفقد اللغة رونقها و جمالها فهي تعطي التعليمات الاساسية للحاسب لتنفيذ البرنامج ....ففي هذا افصل سنقوم بشرح

الجزء الاول جمل التحكم

1- حمل الشرط
2- جمل التكرار
3- حمل القفز

الجزء الثاني هو شرح المصوفات

جمل الحكم بالتدفق

جمل الشرط

بعيدا عن نظام البرمجة تخيل معي انك تقرر انك تريد ان تذهب لشراء ثوب جديد وتريد ان تذهب مع صديقك فتتصل به وتطلب من ان ياتي معك فيقول انه سوف ياتي معك ***اذا*** انتهى من عمله قبل الساعة الخامسة مساءا فتبرمج داخل راسك انه اذا انتهى قبل عمله قبل الخامسة فانه سوف ياتي معي اذا لم ينتهي قبل الخامسة فانه لم ياتي معي وعلي ايجاد صديق اخر ياتي معي

بمعنى ادق انه اذا انتهى قبل الساعة الخامسة فان الشرط تحقق يعني يكون الناتج true لذا فانه يحقق ناتج الشرط انه سوف يأتي معي ولكن لو انه انتهى بعد الخامسة يعني الشرط لم يتحقق يعني الناتج False فانه لم اتي معي لذا يتوجب علي ايجاد صديق اخر

بهذة الطريقة يبرمج بها حياتنا اليومية وكذلك في لغة البرمحة فانك تعطي الشروط للكمبيوتر لتنفيذ اوامر معينة لكي نتمكن من برمجة الكمبيوتر فنجد ان جملة الشرط تنقسم الى قسمين هما

- جملة If
- جملة Select Case

جملة IF

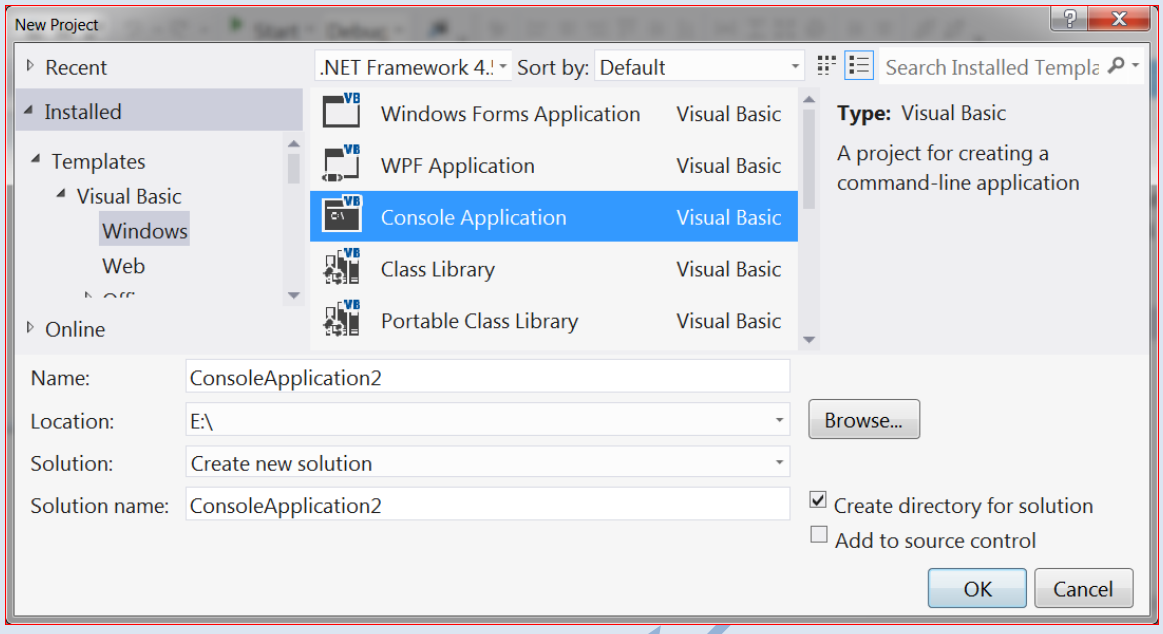
هذة الجملة تتعامل مع شرط معين بحث انه اذا تحقق فانه ينفذ الامر الذي تحته والا فانه ينفذ الامر الاخر

ولها عدة **صيغ واول صيغة** هي

```
If Condition then Statement
```

في الشكل الاول هذا نقول ان كلمة *Condition* يحل محلها الشرط الذي نريد اختباره فاذا كان الناتج True يعني الشرط محقق فانه ينفذ الامر *Statement* وهي تنفيذ الامر اذا تحقق الشرط ولفهم ذلك تابع معي المثال الاتي

نفتح مشروع جديد ولكن نختار مشروع من نوع console كما هو موضح في الشكل الاتي



مشروع كونسل هو عبارة عن مشورع يعرض النتائج على شاشة سوداء كشاشة الدوس عند فتح المشروع تظهر لنا شاشة الكود بشكل مباشر على الشكل الاتي

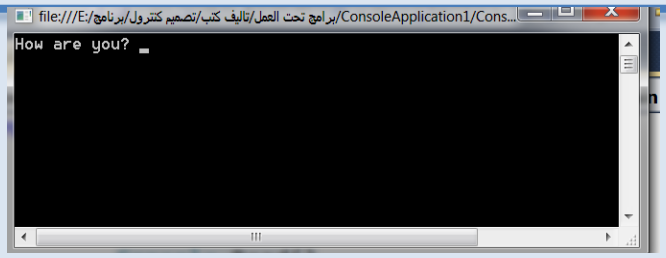
```
Module Module1

    Sub Main()

    End Sub

End Module
```

الاجراء الاساسي الذي يظهر النتائج

في الشكل السابق نقوم نلاحظ وجود اجراء main وهو اجراء اساسي يقوم بقراءة الكود وينفذ الاوامر على البرنامج فهو بمثابة نقطة الانطلاق لكي يبدا الكمبيوتر في تشغيل البرنامج ولكي تسطيع فهم ذلك قم كتابة الكود الاتي في الاجراء main كلي تفهم ما يقوم

```
Console.Write("How are you? ")
Console.Readline()
```

ثم قم بالضغط على F5 فنلاحظ الشكل الاتي :



في الكود السابق نخبر الكمبيوتر بان البرنامج Console بكتابة الامر (“How are you? “)

ليظهرها على الشاشة السوداء اما في السطر الثاني فانك تتركه يفتح التطبيق حتى تضغط على اي زر في لوحة المافتيح لانك اذا اغفلت السطر الثاني فانه يعرض السطر الاول ثم يغلق بسرعة

ولفهم ماهي الاوامر التي في التطبيق كونسل انظر معي في هذا الجدول

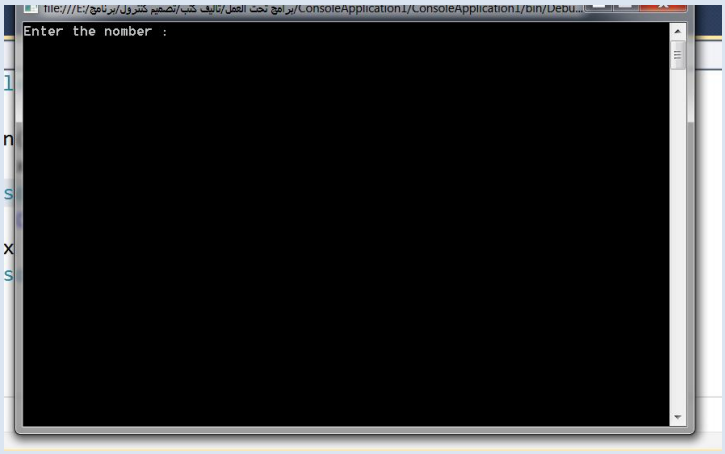
الدالة	استخدامها
Write(متغير نصي)	يستخدم في اظهار المخرجات مثل الرسالة msgbox
Read()	يستخدم على المدخلات inbutbox
Writeline(متغيرنصي)	عرض مخرجات في سطر منفصل
readline()	عرض مدخلات في سطر منفصل

وسوف نستوفي ذلك في الامثلة القادمة في شرح الباب الاول

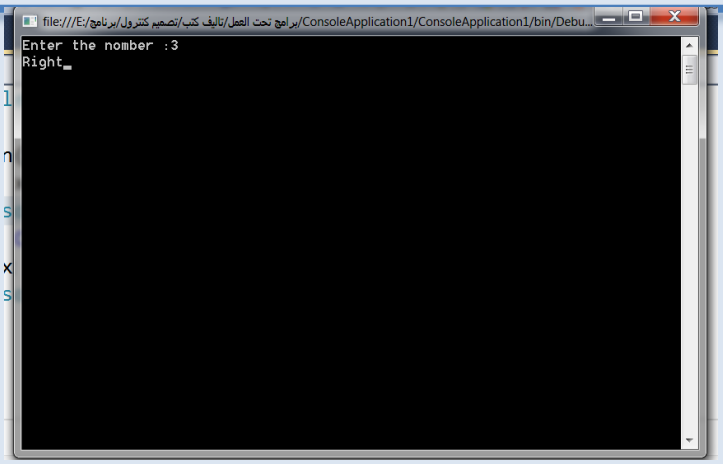
الان نعود لجمل الشرط وننفذ البرنامج الاتي فاكتب الكود الاتي في الدالة main

```
Dim x As Integer
Console.Write(“Enter the nember :”)
  x = Cint(Console.ReadLine())
If x > 2 ThenConsole.Write(“Right”)
Console.ReadLine()
```

عند تشغيل البرنامج سنجد بظهور شاشة الكونسول بالشكل الاتي



عند ادخال اي رقم وليكن 3 فسوف تجد البرنامج يعطيك الشاشة الاتية



وعند ادخال رقم اصغر من 2 سنجد انه لا عطي شيء ويخرج من البرنامج

في الكود السابق قمنا بالاعلان عن متغير رقمي وسميناه ب X في السطر الاول

في **السطر الثاني** طلبنا من الكونسول بكتابة طلب من المستخدم ان يدخل رقم

في **السطر الثالث** جعلنا المتغير X يستقبل رقم من الكونسل من الدالة RightLine حيث ان هذه الدالة تستقبل اي قيمة يكتبها المستخدم مثل الtextbox

في **السطر الرابع** نعطي الشرط اذا كان الرقم المكتوب اكبر من رقم 2 فانه سوف ينفذ الامر الذي يلي كلمة then والا فانه ينتقل للسطر الخامس

السطر الخامس ضروري جدا في الكونسول لنك اذا لم تكتبه فلن ستطيع قرائة الاومر المنفذة بعد الشرط وستجد البرنامج يخرج من تلقائه بعد كتابة الرقم و الضغط على زر ادخال Enter ولكن عند كتابة السطر فانك تستطيع قرائة الامر النفذ بعد الشرط ثم اضغط على الزر Enter للخروج من البرنامج

خلاصة القول ان جملة الشرط التي تستخدم الان هي عند تحقق الشرط فانه ينفذ الامر الذي بعد كلمة Thenكما هذه الصيغة تتيح فقط كتابة امر واحد فقط لكن عندما تريد كتابة عدة اوامر يمكن كتابتها ولكن نفصل بينها بعلامة : قم باستبدال جملة الشرط بهذه الجملة

```
If x > 2 Then Dim i As Integer = x :Console.Write("number" + i.ToString + "is Right")
```

فيكون شكل الكود كلاتي

```
Dim x As Integer
Console.Write("Enter the nember :")
  x = Cint(Console.ReadLine())
 If x > 2 Then Dim i As Integer = x :Console.Write("number " + i.ToString + " is Right")

Console.ReadLine()
```

وعند تجربة البرنامج تلاحظ انه عند تحقق الشرط يقوم بتنفيذ امرين الامر الاول هو الاعلان عن المتغير الرقيمي وهو I فياخذ القيمة من التغير x

و الامر الثاني هو اظهار قيمة المتغير I ... لو لاحظت في الامر الثاني اننا كتبنا الامر `Console.Write("number " + i.ToString + " is Right")`

حيث انك طلبت من الكونسل اظهر القيمة الاتية

"number" وهي قيمة نصية ثم اظهرنا قيمة المتغير i ولكن حولنها لقيمة نصية باستخدام الدالة tostring وعند تخصيص قيمة متغير في سياق كلام تكتبه يجب ان تقطع الكلام الذي تكتبه باغلاق النص باستخدام علامة التنصيص " ثم تضيف قيمة المتغير باستخدام العلامة + او العلامة & حيث ان هذه العلامة تقوم باضافة نص على نص كما شرحنها في الفصل الاول فلو كانت قيمة المتغير I هي 5 فيكون سياق الكلام هو "number 5 is Right

الصيغة الثانية

```
If Condition then
Statement 1
Statement 2
...
....
End if
```

هي نفس الصيغة الاولى ولكنك تنفذ عدة اوامر تحت كلمة Then يعني ليس في نفس السطر ولن تحتاج للعلامة : للفصل بين الاوامر في هذه الحالة يجب كتابة الامر End if عند الانتهاء من تنفيذ الامر لاغلاق هيكل الجملة

```
Dim x As Integer
Console.Write("Enter the nember :")
  x = Cint(Console.ReadLine())
If x > 2 Then
Dim i As Integer = x
 Console.Write("number " + i.ToString + " is Right")
End if
Console.ReadLine()
```

الصيغة الثالثة

```
If Condition then
Statement1
Statement2
Else
Statement 1
```

```
Statement 2

End if
```

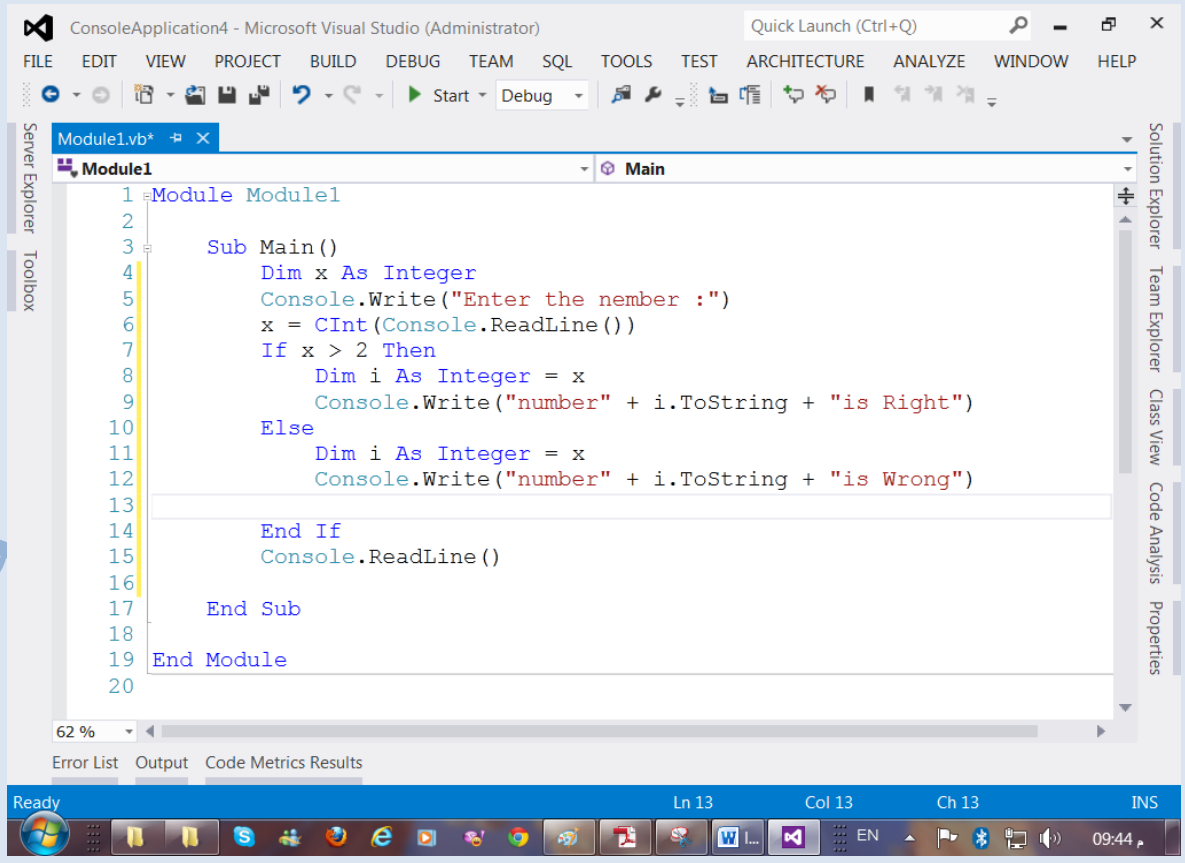
هذه الصيغة سوف نتوقف عندها قليلا فهذه الصيغة مختلفة بعض الشيء

هذه الصيغة تختبر الشرط فاذا كان متحققا يعني اعطى القيمة True فأنه ينفذ الشرط الذي بعد كلمة then واذا لم يتحقق يعني القيمة تكون False فانه ينفذ الامر الذي بعد كلمة Else

جرب هذه الصيغة في المثال السابق وبدل من ان تكتب جملة الشرط السابقة استبدلها بهذه الصيغة المعروفة ب if then Else كما ترى في الكود الاتي

```
If x > 2 Then
Dim i As Integer = x
Console.Write(“number” + i.ToString + “is Right”)
Else
Dim i As Integer = x
Console.Write(“number” + i.ToString + “is Wrong”)

EndIf

Console.Read()

EndSub
```

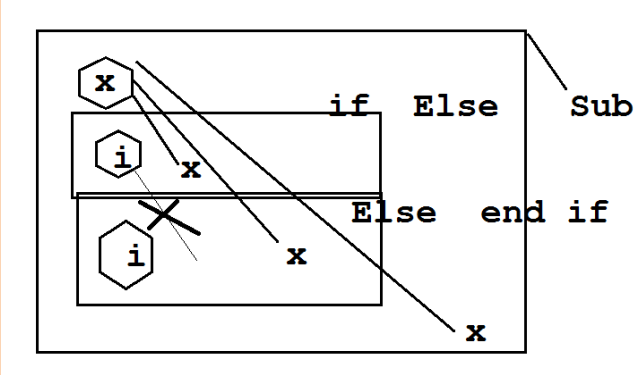
فيكون شكل الكود على النحو الاتي



وجرب البرنامج

اريد ان انبه هنا ان المتغير المعلن عنه دخال هيكل الشرط if الذي سميناه ب I يجب ان تعلم انه مده مقتصر داخل حاوية الهيكل بمعنى انك اذا اعلنت عن المتغير I ما بين then و Else فان مدى I مقتصر بين هاتين الكلمتين التي نسميها بلوك ولكن اذا اعلنت عنه خارج الهيكل فان المدى يصل في بلوك الاجراء الذي في داخال كما يمكن قرائته من داخل الهيكل كما فعلننا في المتغير X لذا لا تستطيع قرائة المتغير I في البلوك الخاص بكلمة else كما تلاحظ اننا قمنا بالاعلان عن متغير بنفس الاسم ولم يعترض الكمبيوتر

انظر للشكل الاتي



في الشكل السابق نقول انك تستطيع ان تقرا قيمة المتغير X في كل البلوكات التي تمثل على شكل مربعات لذا لا يسمح بان تعلن عن متغير X اخر يعني متغير يحمل نفس الاسم داخل المدى الذي يحمله المتغير الاول بينما يمكن ان تعلم ان المتغير I مداه مقصور بين البلوك الصغير لذا يمكن تسمية اي متغير اخر بنفس الاسم طالما لا يصله مدة المتغير i كما كما تراه في البلوك الثاني لو انك استدعيت المتغير Iالمعلن عنه في البلوك الاول لقراءة قيمته من خارج البلوك المعلن فيه سوف يعترض الكمبيوتر لانه لا يعرف المتغير الا في داخل البلوك يعني في داخل المدى المحدد له

هذه الملحوظة سوف تفهمها في فصل لاحق في شرح العمر الافتراضي ومدى كل المتغيرات في الفصل الخامس

هناك صيغة اخرى للجملة If

```
If Condition then
Execute order
Else ifCondition then
Execute order

End if
```

هذه الصيغة تتغامل مع جملة الشرط في وضع عدة احتمالات فاذا لم يتحقق الشرط الاول فانه يختبر الشرط الثاني الموجود بعد Else if

تمرين متروك للقارئ

صمم برنامج يقوم باستقبال رقم الطالب فاذا كان الطالب حاصل على رقم اقل من 5 من 10 فانه فيكتب البرنامج كلمة Fail واذا كان اعلى من 5 فانه يدخل في شرط اخر بحيث انه اذا كان اعلى من 7 فانه يكتب Excealent والا يكتب Succeed

تلميح

استخدم الصيغة السابقة ، if ..... then..... elseif ..... end if امع العلم ان جملة الشرط التي بداخل جملة الشرط الاولى تعتبر جملة كاملة يمكن ان يكون بها كلمة Else ويجب ان يكون بها End if اريدك ان تجرب هذا التدريب بنفسك لتتعلم كيف تدرب نفسك في كتابة الكود

تطبيق على المعاملات المنطقية

احب ان اذكرك باننا كنا قد تعلمنا في *الفصل الاول المعاملات المنطقية* ولان نقوم بعمل تطبيق عليها اتذكر المعاملات المنطقية التي تتمثل في And , Not , Or هذه المعاملات المنطقية ولكن لم نطول في شرحها وقلنا اننا سوف نستعرضها لاحقا ركز معي في هذا المثال الاتي

افتح مشروع كونسل جديد ثم اكتب في الدالة Main الكود الاتي

```
1  Dim x, y, z, m As Integer
2  Console.WriteLine("Enter the first no :")
3         x = Console.ReadLine
4  Console.WriteLine("Enter the Second no :")
5         y = Console.ReadLine
6 Console.WriteLine("Enter the 3 ed no :")
7         z = Console.ReadLine
8  Console.WriteLine("Enter the 4 th no :")
9       m = Console.ReadLine
10  Dim A As Boolean = (x > y)
11  Dim B As Boolean = (z < m)
12  If A And B Then
13   Console.WriteLine("Good")
14   Else

15   Console.WriteLine("Bad")

16   EndIf
```

الان نقوم بشرح الكود

من السطر الاول الى السطرالتاسع : قمنا بتعريف متغيرات رقمية وطلبنا من المستخدم ادخال قيم رقمية للمتغيرات عن طريق الكونسل وقد شرحنا كيفية عمل ذلك سابقا

ثم قمنا بالاعلان عن متغيرين منطقيين نوع Boolean الذي يتعامل بقيمتين فقط هما true و False وقمنا بادخل المعاملات علائقية كاكبر من و اصغر من كما في الكود السابق فاذا كانت القيمة المدخلة على المتغير X اكبر من قيمة y فان العلاقة صحيحة True واذا لم تكن كذلك فان العلاقة False وكذلك في المتغير B

نأتي الي صلب هذا الموضوع وهو العلاقات المنطقية فاننا عملنا جملة شرط if و التي تختبر اللعلاقتين فاذا كان المتغير A يساوي True و B تساوي True فان الاختبار صحيح يكون True فينتقل لتنفيذ الامر تظهر شاشة الكنسل السوداء بكلمة good واذا كان احدهما False يعني ان يكون A= true و B = false فان الاختبر يساوي False فينتقل للبلوك الخاص ب Else ويكتب كلمة Bad

فتخيل لو انك غير علاقة الشرط وكتبتها if A Or B then فان شكل العلاقة سيتغير بمعنا ان **or** تقول ان لو احد الطرفين في العلاقة كان صحيحا فان الاختبار سيكون True يعني لو ان المتغير A يساوي True و B تساوي False فان الاختبار يكون True يعني ان اي المتغيريين يجب ان يكون True اما لو كان المتغيريين False فان الاختبار سيكون False

اما كلمة **Not** فانها تعكس كل ما قلنا سابقا لان Not تنفي الشرط بمعنى انك اذا استخدمت كلمة Notفي الاختبار كما في هذا الكود if Not a then فاذا كان a يساوي True فان الاختبار سيكون خاطيء False وينفذ الامر الذي بعد Else لانه ينفي الاختبار بمعنى يحول ال True الى False

كلمة Xor فانا تكون صعبة قليلا وقد لا تستخدمها كثير ولكن اعطيك فكرة عنها فهي تقول انك اذا كان كلا العلاقتين تساويان True او False بمعنا ان A = true و B = True او العكس ان A = False او B =False فان الاختبار يكون False فلو كان احدهم يساوي True و الاخر يساوي False فان الاختبار يكون True

ويمكن تلخيصها في الجدول الاتي

شرط العلاقة		النتائج			
A	B	A And B	A Or B	A Xor B	Not X
True	True	True	True	False	False
True	False	False	True	True	False
False	True	False	True	False	True
False	False	False	False	False	True

وبعد ان انتهينا من النوع الاول من جملة الشرط ننتقل الى النوع الثاني التي تعتمد على تعدد الاحتمالات وهي Select case

تمرين متروك للقارئ

قم باعادة التدريب السابق ولكن استبدل كلمة Or بدل من And

جملة الSELECT CASE

تخيل معي انك تبرمج برنامج لمدرسة تختبر درجات الطلاب في امتحان النهائية الصغرى تساوي 10 و الكبرى تساوي 20 فانك تقول اذا حصل الطالب على درجة اعلى من 10 فانه يكون ناجح ولكن اذا طلب منك كتابة التقديرات على الشرائح الاتي

الطالب الحاصل من 18 الى 20 يكون التقدير ممتاز Excellent

الطالب الحاصل من 16 الى 18 يكون التقدير جيد جداvery good

الطالب الحاصل من 13 الى 15 يكون التقدير جيد good

الطالب الحاصل من 10 الى 12 يكون التقدير مقبولaccepted

الطالب الحاصل اقل من 10 يكون راسب fail

الطالب الحاصل على الدرجة 20 يكون ممتاز جداvery Excellent

في هذه الحالة فاننا نستطيع عمل البرنامج باستخدام الصيغة if then elseif ولكنني افضل ان اعلمك الصيغة Select case فهي اسهل من صيغة if فحلات تعدد الاختبارات على متغير واحد بمعنى انك سوف تختبر متغير واحد في عدة احتمالات

ولكي نختبر حالة الطلاب فانك ستفتح مشروع كونسل جديد ثم تكتب الكود الاتي في الدالة main

```
Dim X As Integer
Console.WriteLine(“Enter the Student Degree :”)
        X = Cint(Console.ReadLine())
Select Case X
Case 20
Console.WriteLine(“V Exllent”)
Console.ReadLine()
Case 11, 12
Console.WriteLine(“Accept”)
Console.ReadLine()
Case 13 To 15
Console.WriteLine(“good”)
Console.ReadLine()
Case 16, 17
Console.WriteLine(“V good”)
Console.ReadLine()
CaseIs>= 18
Console.WriteLine(“  Exllent”)
Console.ReadLine()
Case Else
Console.WriteLine(“Fail”)
Console.ReadLine()
End Select
```

في الكود السابق قمنا بعمل اختبار على المتغير X الذي اخذ قيمة الدرجة الطالب بحيث اننا لو نظرت الى هيكل الجملة فانك ستجدها تتكون في الشكل الاتي

```
Select Case Variable
```

```
Case Condition 1
Statement 1
Case condition 2
Statement2
Case Else
Statement3
End Select
```

هذة الجملة توضح الاتي

انها تختبر المتغير فاذا كان التغير مطابق للشرط الاول condition 1 فانه ينفذ الامر الذي هو بعد كلمة Case الذي رمزنا له بالرمز Statement1 واذا لم يتحقق فننتقل للشرط الثاني condition 2 فاذا تحقق فنفذ الامر statement2 واذا لم يتحقق اي من الشروط وكتبنا Case else فهو يعني انه اذا لم يكن محقق اي من الشروط السابقة فانه ينفذ الامر statement 3 مع العلم انك اذا لم تضعها فانه يخرج من الهيكل دون ان ينفذ اي امر واريد ان انوه انه انه اذا نفذ اي امر من الاوامر السابقة فانه يخرج على الفور من الهيكل

دعنا الان نعود للمثال السابق و الكود الذي كتبناه لنطبق عليه ما قمنا بشرحه

فانه بعدما قمنا بالعلان عن المتغير وقمن بادخاله في جملة الاختيار Select فانه يجرى على الاختبار الاول وهو Case 20 بمعنى هل يساوي رقم 20 فاذا كان X = 20 فانه ينفذ الامرواذا لم يساويه فانه ينتقل الى الاختبار الثاني وهو Case 11 , 12 بمعنا انه يساوي رقم 11 او 12

الاختبار الثالث Case 13 to 15 وهو معنه اذا كان يساوي اي رقم يبدا من 13 وينتهي الى 15 حيث انه يحتسب معه الارقام العشرية اذا كان المتغير من نوع long او single

وكذالك في الاختبار الرابع

اما الاختبرا الخامس وهو Case is >=18 فانه يختبر اذا كان x اكبر من او يساوي 18 وطبعا سوف سيتعدى الرقم 20 لاننا لم نحدده بمعنى يجب انك لو كتبت مثلا رقم 100 فسوف يكون الناتج Exllent لانك لم تحدده لذا يجب ان نعدل الكود الى ان يكون الاختبار هو Case 18 to 20 ولكني فقط كنت اعطيك اشكال استخدام الشرط في جملة Select Case

اما الاختبار الاخير فاذا كان X لا يساوي ما سبق فانه ينفذ الامرالخاص بالاختبار Else

و الان بعدما انتهينا من جمل الشرط ننتقل الى جمل التكرار وهي تعتبر مشابهه بعض الشيء لجمل الشرط و لكن هي تختف عنها بشيء واحد وهو تكرار الامر حتى يتحقق الشرط فتعالى معي نفهم فكرة هذه الجمل

جمل التكرار

جمل التكرار او ما تسمى جمل اللوب loop هي جمل مهمتها تكرار الامر الذي يوجد في هيكلها حتى يتحقق الشرط المطلوب واذا لم يتحقق تستمر في اعادة تنفيذ الامر حتى يتحقق الشرط او تستمر الي ما لا نهاية وهي لها عدة صيغ فالنبدا الاسهل فيها

جملة DO WHILE

لو نظرنا في هيكل الجملة فنجد الاتي

```
Do while Condition

Statements

.....

Loop
```

في هذا الهيكل يشرح انه اذا تحقق الشرط اي اصبح يساوي True فاننا نقوم بنفيذ الامر حتى يصل الى جملة Loop فنعود للكمة Do while فاذا كان الشرط متحقق فنعود لنفيذ الامر واذا لم يتحقق فانه يخرج من الحلقة التكرارية

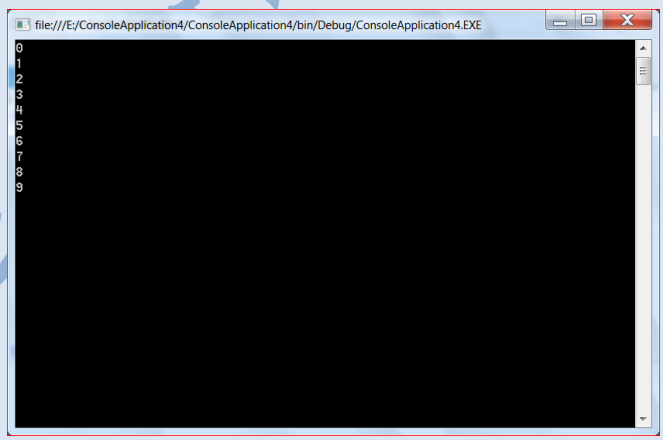
مثال

قم بفتح مشروع كونسل جديد واكتب الكود الاتي في الدالة Main

```
Dim i As Integer
Do Whilei< 10
Console.WriteLine(i.ToString)
i += 1
Loop
Console.ReadLine()
```

في الكود اعلنا عن متغير اسمه i فالمتغير i تكون قيمته 0 لاننا لم ندخل عليه اي قيمة ثم ندخل في عملية اختبار فاذا كان المتغير I اصغر من 10 فانه يدخل الحلقة اتكرارية فيكتب في الكونسل المتغير من خلال الدالة WrileLine ثم نجلعه يضاف اليه 1 وذلك في الكود ***i+=1*** هذا الكود يعني ان قيمة i ستضاف اليه رقم 1 ثم نصل الى كلمة loop فيعود لاختبار الشرط مرة اخرى حتى يكون قيمة الاختبار False يعني تكون قيمة المتغير اعلى من رقم 10فانه يخرج من الحلقة التكرارية

شغل البرنامج ستلاحظ ظهور الشاشة كلاتي



خلاصة القول اذا كان الاختبار في الحلقة التكرارية يساوي True سيتم تنفيذ الامر حتى يصبح الاختبار False

جملة DO UNTIL

هذه الجملة عكس الاولى فاذا كان الشرط يساوي False فانه يستمر في تنفيذ الامر حتى يكون True كما هو موضح في الهيكل الاتي

```
Do until Condition

Statements
```

```
Loop
```

تمرين

سنقوم بكتابة نفس الكود السابق ولكن باستخدام جملة Do until فيكون الكود كلاتي

```
Dim i As Integer
Do  until i> 10
Console.WriteLine(i.ToString)
i += 1
Loop
Console.ReadLine()
```

هذه صيغة جديدة لجملة التكرار

```
Do

Statements

Loop  while Condition
```

هذه الجملة لا تختلف كثيرا عن سابقتها ولكن في بادئ الامر فان الامر ينفذ مرة ثم يختبر الشرط عكس الاولى فانه يختبر قبل ان ينفذ اي امر

بمعنى لو اننا في المثال السابق وعدلنا الكود الخاص بالحلقة التكرارية الى

```
Do
Console.WriteLine(i.ToString)
i += 1
Loop While i = 10
```

فانه ينفذ الامر اولا ثم يختبر رقم 1 ثم يختبر اذا كان يساوي عشرة فانه لا يساويه يعني الشرط غير محقق يعين False فانه يخرج عن الحركة التكرارية

جملة FOR TO NEXT

هذه الجملة غالبا ما تحب الكتب ان بتدا بشرحها قبل جمل Do loop لكني احببت ان اتدرج معك من ناحية تحقيق الشرط حتى تستطيع فهم هذه الجملة فهذه الجملة توفر عليك كتابة كود كثير في عملية تحقيق الشرط لذا اثرت ان تفهم جمل Do loop لتشابهها من جملة If ثم الانتقال بك الى هذه الجملة ...هذه الجملة يكون هيكلها كلاتي

```
For Counter = Start to End  Stepx

Statments

-

Next Counter
```

في الجملة يكون ال Counter هو المتغير نقوم باختباره بحيث يكون له قيمة ابتدائية Start فيقوم بزيادتها حتى تصل للقيمة النهائية End بحين يكون قيمة الزيادة او النقصان من خلال الكملة Step فيتم تنفيذ الامر حتى تصل الجملة للقيمة النهائية

طبعا لفهم ما قلته سابقا نقوم بتنفيذ المثال السابق ولكن باستخدام الجملة For واليك الكود

```
Fori = 0 To 10 Step 2
Console.WriteLine(i.ToString)
next
```

في الكود السابق اعطينا للمتغير I قيمة ابتدائية Start تساوي 0 وقيمة نهائية وهي 10 ونعطي امر القفز من الجملة Step وهي تشبه الامر i+=2 بحيث ان المتغير I يزيد كل مرة تكرار على حسب القيمة التي تلي كلمة Step فلو كانت قيمة المتغير I تساوي 2 بعد ان ينتهي التكرار تصبح قيمة المتغير I تساوي 4 وهكذا حتى تصبح قيمته تساوي 10 او تزيد فيخرج الكمبيوتر عن الحلقة التكرارية

ملحوظة هامة

يمكنك الاستغناء عن جملة Step في جملة التكرار فهذا يعني ان قيمة الزيادة في المتغير ستكون رقم 1

تنويه

قد لا تشعر باهمية هذه الجمل في الوقت الحالي ولكن مع تقدم العلم ستعلم باهميتها عندما تتعامل مع المصفوفات او قواعد البيانات فتعرف مدى اهمية هذه الجمل

جمل القفز

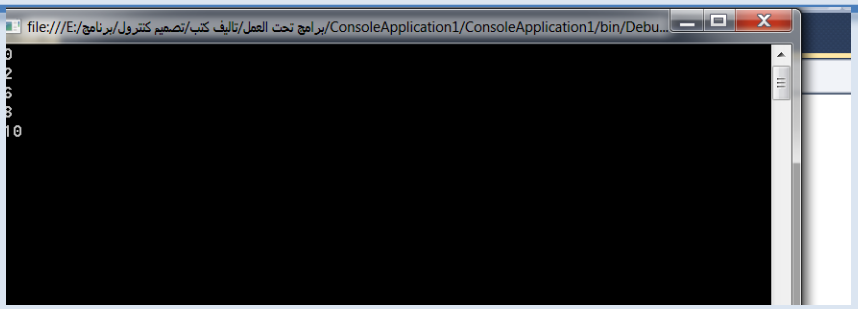
جملة CONTINUE

هذه الجملة تقوم بايقاف العمل تحت الحلقة التكرارية او القفز فهي تستخدم جملة next او loop

لايضاح ذلك في المثال السابق عد كود حلقة التكرار الى الكود الاتي

```
Fori = 0 To 10 Step 2
Ifi = 4 Then Continue For
Console.WriteLine(i.ToString)
Next
```

نجد اننا عملنا اختبار للقيمة I فاذا كانت تساوي القيمة 4 فانها تقفز الى الكلمة Next ثم تعود لتعيد الاختبار مرة اخرى فنجد ان اناتج بعد تشغيل البرنامج في الشكل الاتي



تلاحظ انه قام بكتابة الرقم 2 ثم 6 اي قفز من الرقم 4 لان الشرط اصبح صحيح

GOTO

جملة Go To هي جملة التي تقوم بعمل قفز من سطر الى سطر اخر ولكي تستطيع فهمي اكثر اذهب الى اي مشروع كتبه ثم اكتب الكلمة في اي سطر في لاجراء الاتية

Start :

ثم اذهب في اي مكان اخر في الاجراء واكتب جملة Go to ثم الكلمة Statr فتصبح Go to Start

ثم في حالة التنفيذ ستلاحظ اذ اوصل المترجم الى السطر الذي فيه جملة Go To يذهب الى السطر الذي في كلمة Start

لا يشترط كلمة Start يمكن كتابة اي جملة اخرى بشرط اتباع شروط التسمية التي شرحنها في المتغيرات ثم كتباة النقطين (:)

مثال

اكتب الكود الاتي في الاجراء main

```
            Dim i As Integer
            Console.WriteLine("Enter the  nomber:")
            i = CInt(Console.Read)
            If i < 5 Then
                GoTo moaz
            Else
                GoTo Baraa
            End If
Baraa:
            Console.WriteLine("no")
            Console.ReadLine()
moaz:
            Console.WriteLine("yes")
            Console.ReadLine()
```

في الكود السابق اعلنا عن متغير سوف نختبره

في السطرين الثاني و الثالث استخدمنا الكنسل لاخذ القيمة من المتسخدم

ثم استخدمنا جملة الشرط لاختبار المتغير فاذا كان اقل من 5 فاننا ننتقل للكود الذي هو تحت moaz والا ننتقل للكود الذي هو تحتد Baraa مع العلم انه سيكمل باقي الاسطر و يدخل الى الكود الذي هو تحت القسم maoz واذا اردت ان تخرج خارج الاجراء او تقوف عمل الكود فاكتب كملة Exit Sub للخروج خارج الاجراء

المصفوفات

المصفوفات ARRAY

تخيل معي انك تكتب كود به سماء زملاءك في العمل او في الكلية او الفصل وتريد انت تجعل لكل واحد في الفصل متغير خاص به فانك تعلن عن متغيرات كثيرة لكي تكفي كل الاسماء فتخيل انك تريد ان كتبك وكود للاسماء الاتي

- **محمود احمد علي**
- **خالد حسن خالد**
- **محمد صالح ابراهيم**
- **علي علي علي**

فانك تعلن عن متغير خاص محمود ثم متغير خاص بخالد ومتغير خاص بمحمد واخر لعلي حتى تحل مشكلة معينه فانك ستحتاج لمجهود طوييييييييل واكواذ كثير وتخيل انك تكتب كود لدفعة لا يقل عددها عن 500 فرد هل من المنطقي انك تعلن عن 500 متغير هذا صعب كثيرا

لذا ظهرت لدينا المصفوفاتarray التي تتعامل مع هذه المشكلة بصورة سريعة

المصفوفات هي عبارة عن عدة متغيرات مشتركه بنفس الاسم ونفس النوع ولكنك تفرق كل متغير عن الاخر بالرقم الموجود بين القوسين المعروف ب index او الفهرست الذي سهل لك لوصول اليه .

للاعلان عن المصفوف يجب اتباع الصيغة الاتية

```
Dim Array Name(Max index) As Data type
```

في هذه الصيغة نعلن لكلمة Dim و التي سنتناول شرحها بالتفاصيل في فصول قادمة عن معناها

اما كلمة *Array Name*فهي تعني ادخال اسم المتغير او اسم المصفوفة نتبع نفس شروط التسمية التي شرحناها في تسمية المتغير

اما بعد الكلمة التي بين القوسين و التي تعني *Max index* التي تعني ادخال القيمة التي تحدد عدد المتغيرات دخال المصفوفة التي تسميها عناصر members ولكي تحدد عدد عناصر المصفوفة عليك بتحيدد عدد عناصر المصفوفة وليكن عندنا 5 عناصر وعند الاعلان و الكتابة بين القوسين يكون عدد العناصر مطروح منه 1 القيمة الفهرسية للعنصر تبدا ب 0 فيكون الاعلان عن العناصر على الشكل الاتي

```
Dim X(4) As Integer
```

ولان نوضح انك اعنت عن مصفوفة تحمل اسم X نوعها رقمي بها 5 عناصر لانك تحدد العناصر ثم تطرح منها 1 لانك تخبر المصفوفة باكبرقيمة عنصر عندك يعني عناصر المصفوفة مرقمة من 0 و 1 و 2 و 3 و 4 عددها خمسة واكبر قيمة دليل 4

و الان نقوم بادخل اعطاء قيمة لعناصر المصفوفة

x(0) = 2

x(1)= 3

x(2)=4

x(3) = 50

x(4)= 70

ترى الان كيف يتم تخصيص قيمة للعناصر مع العلم انك اذا لم تعطي قيمة الي عنصر في المصفوفة فان فان العنصر ياخذ قيمة افتراضية 0 اذا كان نوع المصفوفة رقمي و "" اذا كان نوع المصفوفة نصي و قمية False اذا كانت نوع المصوفة منطقي

عمليات على المصفوفة

و الان نقوم بعمل بعض العمليات على المصفوفات كايجاد اكبر قيمة مفهرسة او اصغر قيمة و بحث عن عنصر ميعن في المصفوفة وذلك باستخدام الدوال التابعة للمصفوفة وهي مندرجة من التصنيف الخاص بالمصفوفات لذا لكي نستدعي اي دالة تخص المصفوفة فاننا نقوم بالتي

اسم الدالة .X

حثي ان X هو اسم المصفوفة ثمن تختار اسم الدالة التي نريدها و الجدول الاتي يوضح اهم الدوال التي نستخدمها

اسم الدالة	الاستخدام
x.GetLowerBound(dimension)	هذه الدالة تقوم باعادة اصغر قيمة مفهرسة يعيني القيمة 0 اذا بدات المصفوفة بالقيمة صفر اما عن dimension فهو معامل رقمي يحدد مدى ابعاد المصفوف وسنشرحها بعد ان نبين الدوال
x.GetUpperBoun(dimension)	هذه الدالة تحدد اعلى قيمة مفهرسة في المصفوفة بمعنى انه اذا كانت المصفوفة بها 5 عناصر فان القيمة ستكون 4
x.GetLength(dimension)	هذه الدالة تحسب عدد عناصر المصفوفة

ولان مع دوال اخرى ولكن يتم استدعاءها بطريقة ثانية وهي كتابة كلمة

اسم الدالة .Array

اسم الدالة	الاستخدام
Array.BinarySearch(x, 50)	هي دالة تقوم بعمل بحث عن عنصر معين في المصفوفة ثم تقوم بارجاع رقم العنصر اما عن القيمة X فهو اسم المصفوفة التي نريد البحث فيها و القيمة 50 هي القيمة التي نريد البحث عنها
Array.Sort(x)	هذه الدالة تقوم بترتيب عناصر المصفوفة ترتيب تصاعدي

المصفوفة الدينميكية

تخيل لو انك تريد عمل مصفوفة ولكنك لا تستطيع تحديد عناصرها لعدم التاكد من عدد عناصرها العناصر فيها فانك تعلن عن هذه المصوفة الدينميكية في تم الاعلان عنها على الشكل الاتي

```
Dim x() as integer

او

Dim x as integer ()
```

عندما نريد ان نخصص قيمة للمصفوفة فيتم اعطائها القيمة عند الاعلان كلاتي

```
Dim x () As integer = {1,2,3,4,5}
```

كما تستطيع ان تاخذ قيمة مصفوفة اخرى ولكن بشرط ان تكون من نفس النوع ونفس البعد بعدين والكلام عن الابعاد

Dim x () as integer	مصفوفة دينامكية غير معلومة العناصر
Dim m(2) as integer	مصفوفه بها عدد عناصر
I(0) = 1	ادخال قيمة لغناصرا لمصفوفة
I(1) = 20	
I(2)= 30	
X = i	تخصيص المصفوفة معلومة عدد العناصر للمصفوفة الدينميكية

سنشرحها في السطور القادمة فتابع معي الكود الاتي

ملحوظه هاااااامة جدا

لا يسمح لك ان تضيف قيمة لاي عنصر من عناصر المصفوفة الدينميكية الا اذا كنت قد خصصت لها قيمة مسقبا مثل هذه المصفوفة

```
Dim x () As integer = {1,2,3,4,5}
```

بينما لو اعلنت عن مصوفة دينكيكية مثل dim x () as integer (مصفوفة دينيكية غير معلومة العناصر) واذا خصصت قيمة للعنصر 0 بهذه الطريقة x(0)= 5 سوف يعترض الكمبيوتر ويعطي

استثناء

لذا يجب ان تحدد عناصر المصفوفة الدينميكية بتخصيص للمصفوفة الدينميكية مصفوفة اخرى معروفة العناصر كما في الثال السابق باننا خصصنا مصفوفة معلومة العناصر X = i حيث ان x مصفوف دينميكية و I مصوفة معلومة العناصر

او تخصص قيم للمصفوفة اثناء الاعلان عنها مثل هذه المصفوفة

Dim x () As integer = {1,2,3,4,5}

تطبيق عملي

سوف نتعلم كيف يتم التعامل بشكل عملي على المصفوفات وتجد مدا علاقتها بجمل التكرار loop فاليك مثال مبسط عن كيفية استخدام المصفوفات

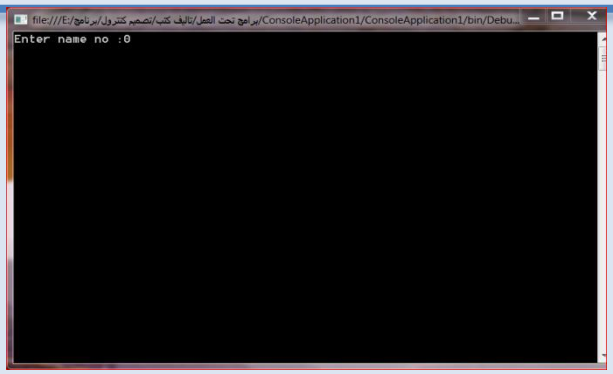
افتح مشروع كونسل جديد وادخل على الدالة main الكود الاتي

```
1 Dim x(6) As String
2 for i As Integer = 0 To x.GetUpperBound(0)
3 Console.WriteLine("Enter name no :" + i.ToString)
4 x(i) = Console.ReadLine()
5 Next
6 Console.WriteLine("The name you Enterd is ")
7 For i As Integer = 0 To x.GetUpperBound(0)
8 Console.WriteLine(x(i))
9 Next
10 Console.ReadLine()
```

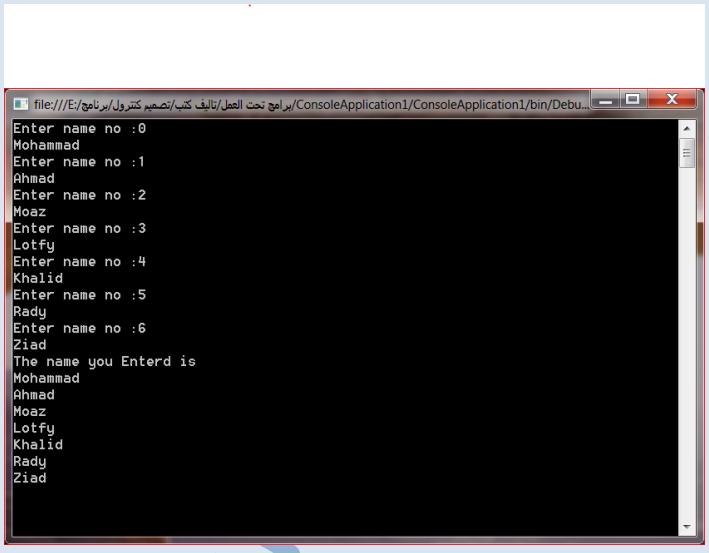
في السطر الاول قمنا باعلان عن مصفوفة من 7 عناصر ونوعها نصية

في السطر الثاني قمنا بعمل حلقة تكرارية مهمتها هي ان تقوم بتكرار الامر الذي يتيح للمستخدم بادخال الأسماء عدد عناصر المصفوفة

من السطر السادس الذي الحلقة التكرارية قمنا بعمل حلقة تكرارية لعرض الاسماء التي كتبناها من قبل وعند تشغيل البرنامج نجد انه يعرض لكل الشاشة الاتية



وبعد ادخال كل عناصر المصفوفة ستجد انها تعرض عليك الاسماء مرة اخرى في الشكل الاتي



اعادة الاعلان عن عناصر المصفوفة

اذا كنا نتعامل مع مصفوفة نصية بها 5 عناصر ونريد ان نغير عدد عناصرها الى 8 عناصر فاننا نتبع الخطوات الاتية

```
ReDim ArrayName (New max Index)
```

هي المصفوفة التي نريد تغيير عدد عناصرها ولكن بشرط ان يتم الاعلان عنها مسبقا

New max Index هي رقم العناصر الجديدة

وعند تطبيق القاعدة السابقة انظر معي الكود الاتي

```
ReDim  x  (7)
```

اظن انك تعلم لماذا اعطينا لعدد عناصر المصفوفة رقم 7 وليس 8 لتمط تعطيها قيمة اكبر دليل في المصفوفة

المصفوفة متعددة الابعاد

في السطور السابقة شرحنا المصفوفة ذات بعد واحد ولكن تخيل انك تريد ان تجعلا المصفوفة متعددة الابعاد بمعنى تخيل معي الاتي

انت في المصفوفة ذات البعد الواحد فانك تدخل بيانات من عمود واحد يعني بيانات طلاب فقط او الدرجات فقط فلو انك قمت بادخال البيانات و الدرجات في مصفوفة واحدة يعني تخيل انها في الجدول الاتي

Ahmmad	20	14
Moaz	20	16
Ali	15	18
Yasser	20	20

هذا الجدول به صفوف و اعمدة يعني بعدين لمصفوفة بها 4 صفوف و 3 اعمدة

وعند الاعلان عنها فاننا نعلن عنها كلاتي

```
Dim x(3,2) As string
```

بذلك قمنا باعلان عن مصفوفة ذات اربع صفوف و ثلاثة اعمدة

فلو اردنا ان نكتب اسم Moaz في مكانه في المصفوفة فانه موجود في الصف رقم 1 و العمود رقم 0 لانك تعلم ان العد يبدا من الصفر

```
X(1,0) = "Moaz"
```

ولو كان مصفوفة ذات بعدين دينميكية فهي يتم الاعلان عها هكذا

```
Dim x(,) as string
```

ويتم تغيير عدد عناصرها هكذا

```
Redim x (2,2)
```

ملحوظة

هل تذكر الدالة `GetUpperBound(dimension)` كان لدينا المعامل `dimension` الذي يحدد رقم بعد المصفوف فلو كانت المصفوفة ببعدين تكون قيمة المعامل `dimension 1` بحيث يكون البعد الاول قيمته `0` و البعد الثاني قيمته `1` كما يمكن ان تكون المصفوفة متعددة الابعاد وليس بعدين هكذا المصفوفة `dim x (,,,,) as string` حيث انها تكود مصفوفة بها 5 ابعاد وتكون قيمة المعامل `dimension` هي 4

احب ان اطمئنك اذا كنت غير مرتاح في فهم المصفوفات ولا تستطيع تطبيقها بشكل عملي فاني اعدك بان افرد لها تمارين مخصوصة في الجزء الثالث من هذه السلسلة وهو الكتاب الذي يشرح التعامل مع قواعد البيانات لاننا سنحتاج اليها بشكل كبير

الفصل الثالث

الدوال و الاجراءات

اي عمل كبير يجب ان يتكون من مجموعة من اجزاء صغيرة فمثلا صناعة السيارات و الطائرات عي عبارة عن تصنيع اجزاء صغيرة ثم نربطها ببعضها حتى يتكون الجسم الكبير كطائرة او سيارة او مبينى او حتى كتاب فالكتاب يكتب على عدة فصول او حتى انت تقسم الكتاب على عدة اجزاء حتى يسهل فهمها

هذا الكلام ينطبق على البرامج بحيث انك عندما تطور برامجك يجب علي ان تقسم عملك الى عدة اجزاء حتى لا يفوتك اي شيء ويكون العمل المتقن .

ولكي تقوم بتقسيم البرنامج يجب ان يحتوي البرنامج الكبير الى مجموعة البرامج الصغيرة هذه المجموعة الصغيرة تقسم الى برامج اصغر ثم نقوم بربطها بعضها ببعض هذه البرامج الصغيرة مكونة من عي عبارة عن دوال واجراءات ثم يتم تجميع الدوال و الاجراءات مع عناصر اخرى كالخصائص و الاحداث فنكون برنامج اكبر اسمه تصنيف Class ثم نربط التصنيف مع تصنيفات اخرى فنكون مشروع او برنامج ويتم اعداد بعد ذلك هذا البرنامج لكي يكون في يد المستخدم

ففي هذا الفصل سوف نشرح الان اللبنة الاولى في بناء البرامج وهي الدوال و الاجراءات

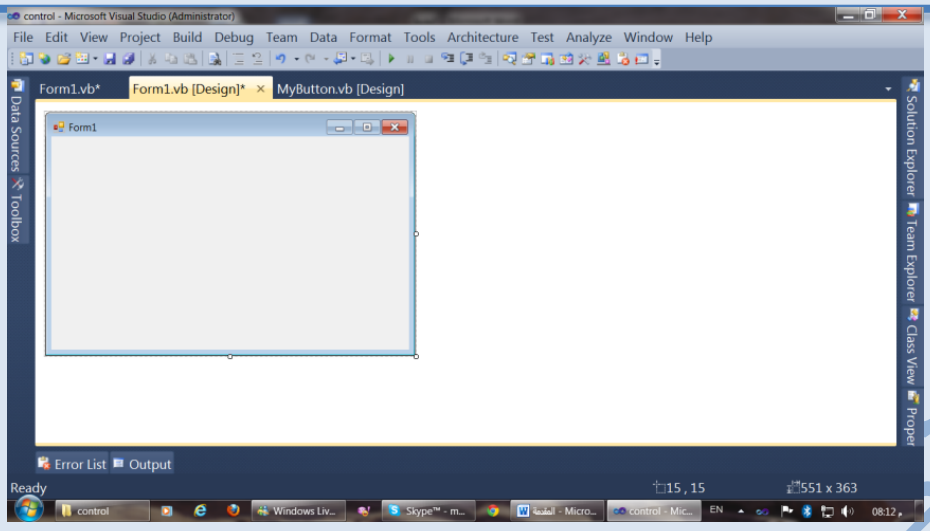
الاجراءات

الاجراء عبارة عن برنامج فرعي صغير مستقل به مجموعة من الاوامر بحيث اننا نقسم البرنامج الكبير الى مجموعة من البرامج الصغيرة ، فالاجراءات و الدوال عبارة عن وحدة بناء البرامج الكبيرة مثل قوالب الطوب التي نرصها لبناء بيت كبير وهذه الاجراءات تقسم الى :

اجراء يستدعى بالاحداث

هو برنامج فرعي لا يتم تفعيله الا عند وقوع حدث ما مثلا يقوم المستخدم بعمل ضغط على الماوس او كتابة مدخلات بطريقة معينة فيتولد الحدث وطريقة تصنيع الحدث وتفعيله سنناقشها في الفصل التالي ولكن احب ان اخبرك ان الاجراء المستدعى من الحدث يمكن ان تعرفه في التطبيق الخاص بالنوافذ ، ولكي تتعرف اكثر افتح مشروع تطبيقات النوافذ Windows Application

فتجد الصورة الاتية

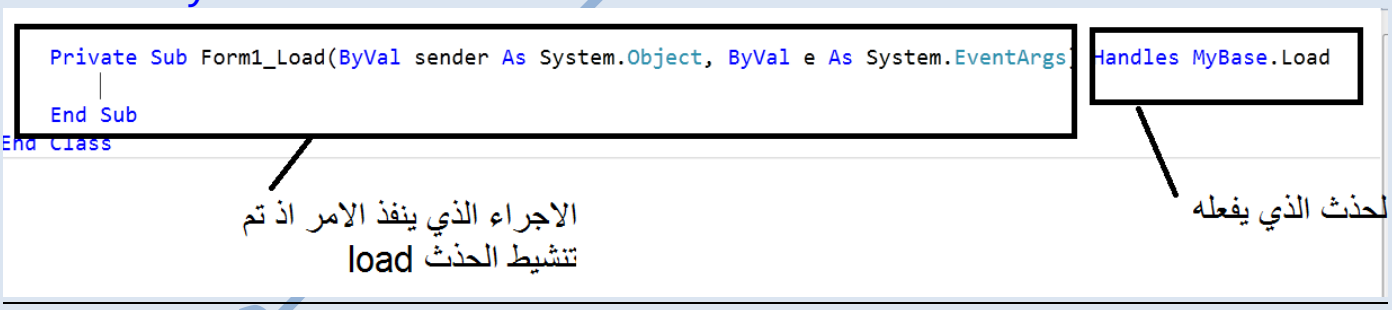


وعند الضغط على النموذج مرتين تظهر شاشة الكود على الشكل الاتي

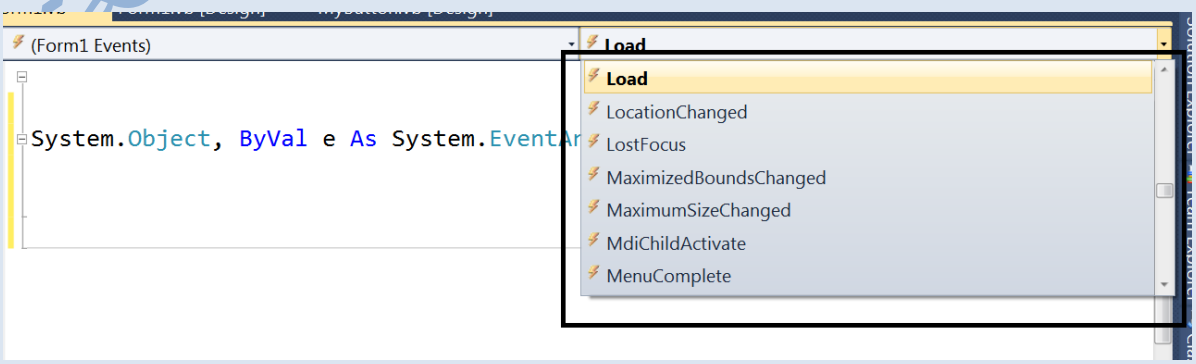
```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.Even

    End Sub
End Class
```

في الشكل السابق ترى الكود تحت المربع هذا هو الاجراء بحيث انه يفعل عند تحميل النافذة عن طريق الحدث load ولكي تتاك انظر اخر السطر الاول ستجد هناك الكلمة الاتية

```
Handles MyBase.Load
```



و كلي تحب ان تستدعي اجراء ليقوم بتنفيذ امر معين عند مرور الماوس على النموذج انتقل الى القائمة اعلى يمين شاشة الكود كما هو موضح في الشكل الاتي



اختر الحدث المرور الماوس mouse_move وسيظهر الخاص بعمل بمرور الماوس وافضل ان اشيره هذا النوع في الفصل القادم باذن الله حيث اننا سوف نتناول هذا الاجراء بشكل تفصيلي في قسم التعامل مع الاحداث انتقل معي الان للنوع الثاني من الاجراءات وهو الاجراء الفرعي المستقل

اجراء فرعي

هو برنامج فرعي يساعد على تقسيم البرنامج حتى لا يتم تكرار الكود كما يسهل الوصول اليه و يسهل تعديله و اصلاحه هذا البرنامج الفرعي يعتبر حجر البناء للبرامج .

ولكي نقوم بانشاء الاجراء الفرعي اتبع معي هذه القاعدة الاتية

```
[Public , Private]Sub  Sub Name (Argument)

Statements

End Sub
```

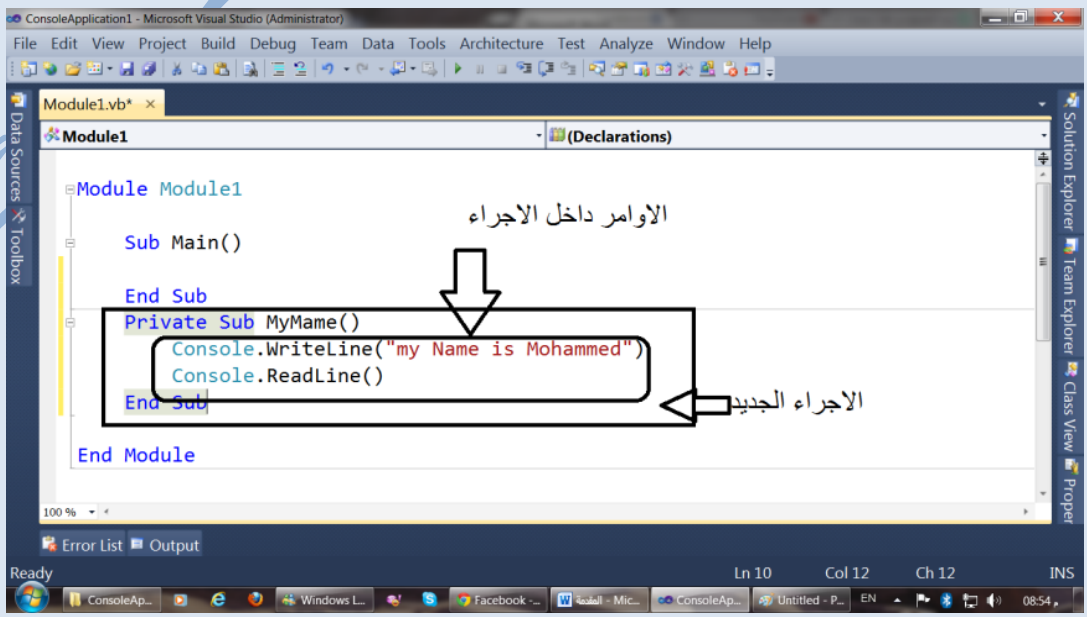
[Public , Private] : سيتم شرحها بالتفاصيل الممل في الفصل الخامس فالافضل اننا سنستخدم التعبير private في الوقت الحالي فهي تحدد مدى رؤيتنا للاجراء

Sub Name هو اسم الاجراء الفرعي ونتبع في تسميته نفس قواعد تسمية المتغيرات كعدم البدئ برقم او ادخال اي رميز ما عدا الرمز (_) وعدم استخدام كلمات اللغة

Argument : هو المعامل الذي نمرر اليه قيم و المعامل عبارة عن متغير نعلن عنه بحيث اننا اثناء استدعائنا للاجراء نعطي للمعامل قيمة هذه القيمة نستخدمها في داخل الاجراءكما سنوضح في المثال الاتي

مثال

افتح مشروع كونسل جديد واكتب الكود الاتي لكن خارج الاجراء main كما في الشكل الاتي



تلاحظ شكل الاجراء وكما موضح في الشكل اننا كتبنا اوامر البرنامج الفرعي بين كلمة privat sub Myname و End sub

```
Private Sub MyMame()
Console.WriteLine(“my Name is Mohammed”)
Console.ReadLine()
EndSub
```

في السطر الاول اعلنا وجود اجراء فرعي خاص بالكملة private sub هم حددنا اسمه حيث اننا جعلنا اسمه هو MyMame ثم وضعنا قوسين () حيث اننا لم نعلن عن اي معامل او متغير نحتاج اليه لان الاعلان عن المعاملات امر اختياري على حسب حاجتك للبرنامج

في السطرين الثاني و الثالث كتبنا الاومر المطلوبة تنفيذها في البرنامج

في السطر الاخير قمنا بانهاء الاجراء End Sub

لو شعلنا البرنامج لن تجد شيء الانك قمت انشاء الاجراء ولم تفعله ولتفعيله يجب استدعاءه في الاجراء الذي يقوم البرنامج بتنفيذ البرنامج من خلاله وهو main وطريقة الاستدعاء تكون كلاتي

```
Sub Name(argument Value)
```

Sub Name : هنا نكتب اسم الاجراء المطلوب تنفيذه

argument Value : هنا نعطي القيمة التي نريد تمريرها للبرنامج الفرعي لاجراء المعالجة عليها

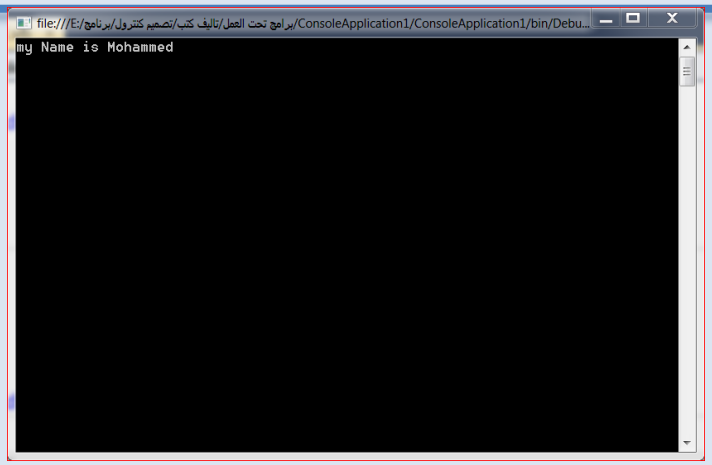
لكي تستطيع فهم ما سبق سنعود للمثال السابق و اكتب اسم الاجراء داخل الاجراء main في الشكل الاتي

Myname()

لكي يكون الكود كالاتي

```
Module Module1

    Sub Main()
        MyMame()
    End Sub
    Private Sub MyMame()
        Console.WriteLine("my Name is Mohammed")
        Console.ReadLine()
    End Sub

End Module
```

وعند تشغيل البرنامج ستجد انه نفذ الاوامر في الشكل الاتي



تلاحظ ان الامر الذي كتبناه في الاجراء MyMame انها تم تنفيذها في الاجراء main

نحن الان نريد ان نجعل البرنامج مرن بحيث انه لا يصنع من اجل الشخص محمد فقط ولكن نريد ان يكون لكل الاشخاص لذا سنقوم ببعض التعديلات على الكود السابق سنضيف معامل من نوع نص لكي يحمل الاسم الذي نريد ان يظهر على الشاشة

سوف نعدل الاجراء الذي كتبناه في المثال السابق الى ان يضم معامل وندخل المعامل ليتم معالجته داخل الاجراء

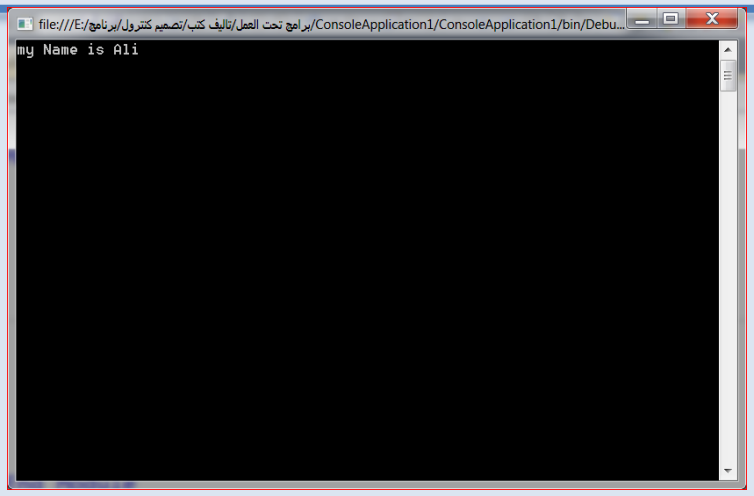
```
Private Sub MyMame(ByVal text As String)
Console.WriteLine("my Name is " + text)
Console.ReadLine()
End Sub
```

تلاحظ التعديل على الاجراء بالاعلان على المتغير داخل قوس المعاملات وهو text من نوع نصي String
في السطر الثاني قمنا بتعديل الكود بحيث اننا استبدلنا كلمة Mohamed بالمتغير text بحيث اننا عند استدعاء الاجراء وندخل مثلا كلمة Ali فتظهر لنا الجملة My Name is Ali

وعند استدعاء الاجراء من داخل الاجراء main فاننا يجب علينا اعطاء قيمة للمعامل كما في شكل الاتي من داخل الاجراء main

```
MyMame("Ali")
```

هنا اعطينا قيمة للمعامل ان يدخل كلمة علي الى الاجراء لنفيذ الامر كما في الشكل الاتي بعد تشغيل البرنامج



ولكنك يجب عليك ان تجعل البرنامج مرن يجب ان يدخل المستخدم الاسم لذلك عدل كود الاجراء main الى الشكل الاتي

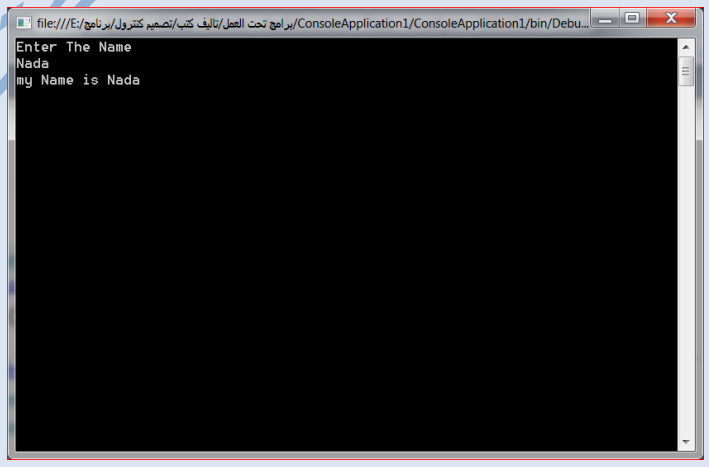
```
Console.WriteLine("Enter The Name")
Dim x As String = Console.ReadLine
MyMame(x)
```

في السطر الاول طلبنا من المستخدم ان يدخل الاسم

في السطر الثاني اعلنا عن المتغير يستقبل القيمة التي يدخلها المستخدم

في السطر الثالث عدلنا المعامل بحيث انه ياخذ القيمة التي في المتغير X حيث ان المتغير x قد خزن بداخل القيمة التي ادخالها المتسخدم في السطر الثاني

قم بتشغيل البرنامج فتظهر لك نافذة تطلب من ادخال الاسم ادخل كملة Nada ثم انقر على Enter من لوحة المفاتيح فيظهر لك الشاشة كلاتي



تلاحظ ان الكلمة Nada التي ادخلناها تم تخزينها في المتغير x ثم انتقلت الكلمة من المتغير x الى المتغير Text الذي اعلنا عنه كمعامل دخال الاجراءMyMame ثم ظهرت بعد ذلك بعد كلمة

My Name is

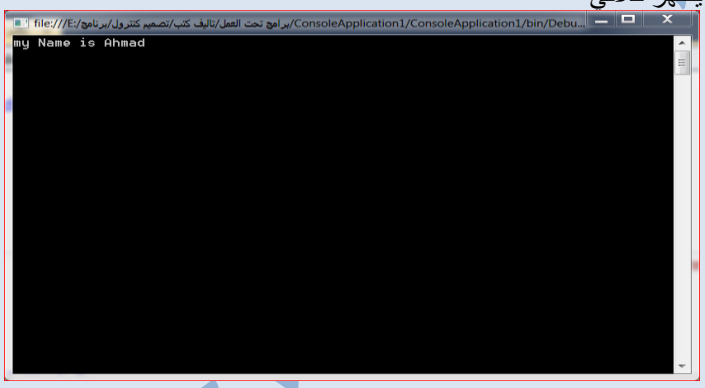
الى هنا تكون تعرفت على كيفية عمل اجراء و كيف ندخل معاملات اليه ولكن ما رأيك ان يكون المعامل اختياري عند استدعاءه بمعنى اننا عندما نستدعي اجراء معين يكون اجراء يكون لدى المعامل قيمة افتراضية تغنيك عن كتابة اي قيمة اثناء استدعاء الاجراء الا اذا كنت تريد التعديل عليها لذا سوف نجري تعديل في البرنامج السابق بحيث اننا نعدل الاجرء Mymame الى الكود الاتي

```
Private Sub MyMame(Optional ByVal text As String = "Ahmad")
Console.WriteLine("my Name is " + text)
Console.ReadLine()
End Sub
```

تلاحظ في السطر الاول كلمة optional تعني معامل اختيار للمستخدم ان يدخل القيمة او لا وعند الاستدعاء نقوم بكتابة الكود الاتي

```
MyMame()
```

وعند تشغيل البرنامج تلاحظ البرنامج يظهر كلاتي



تلاحظ كلمة Ahmad الذي كتبناها كقيمة افتراضية للمعامل بمعنى ننا نترك الامر اختياري اما ان نعدل على القيمة الافتراضية او نتركها كما هي

ولكي نتمكن ما انشاء معامل اختياري اتبع مع القاعدة الاتية

[*Public , Private*]Sub *Sub Name* (optional *Argumentname*AS *Argument type* = *Default Value*)

نتكلف فقط عن المعامل المتغير يجب كتابة كلمة optional قبل كتابة المتغير ثم اسم المعامل ثم نوعه بعد ذلك يجب كتابة القيمة الافتراضية

الدوال

الدوال شبيهة الشبه الكبير بالاجراءات ولكنها تختلف عنه اختلاف بسيط جدا هو اننا بعد تنفيذ الاوامر الموجودة داخل الدالة تقوم بعمل ارجاع قيمة واحدة كناتج عن هذة الدالة هذه القيمة نستخدمها في باقي برامجنا ولتوضيح ذلك تعالى نتغرف عن كيفية انشاء الدالة

[*Public , Private*]Function *Function Name* (*Argument*)As *Function Type*

```
Statements
Return Value
End Function
```

[*Public , Private*] : سيتم شرحها بالتفاصيل الممل في الفصل الخامس فالافضل اننا سنستخدم التعبير private في الوقت الحالي فهي تحدد مدى رؤيتنا للاجراء

Function Name تسمية الدالة بنفس الطريقة التي نسمي بها المتغير تماما

Function Type تحديد نوع الدالة التي تقوم بارجاع قيمتها

Argument : هو المعامل الذي نمرر اليه قيم مثل الاجراء تماما

Return*Value* : هنا نقول ان الدالة يجب اعطاء قيمة نهائية في اخر الدالة ويجب ان تكون القيمة من نفس نوع الدالة ولتوضيح ذلك انظر المثال الاتي

سوف نقوم بعمل برنامج للجمع بين رقمين فقم بفتح كونسل جديد وكتب الدالة الاتي خارج الاجراء Main كلاتي

```
Private Function Addno(ByVal no1 As Integer, ByVal no2 As Integer) As Integer
Dim n1, n2 AsInteger
        n1 = no1
        n2 = no2
Return n1 + n2
End Function
```

في السطر الاول اعلنا عن وجود دالة تسمى Add ستقوم بارجاع قيمة رقمية وبها معاملين لادخال الرقمين وجمعهما

في السطر الثاني اعلنا عن وجود متغيرين يقومان باستقبال قيم المعاملين كمه هو في السطر الثالث و الرابع

في السطر الخامس الذي به كلمة Return تقوم الدالة برجاع القيمة النهائية من الجمع

عند استدعاء الدالة واخذ قيمتها منها فنقوم بكتابة الكود الاتي داخل الاجراء Main

```
1 Dim i1, i2 AsInteger
2 Console.WriteLine("Enter No 1 :")
3 i1 = Cint(Console.ReadLine)
4 Console.WriteLine("Enter No 2 :")
5 i2 = Cint(Console.ReadLine)
6 Dim Sum AsInteger
7 Sum = Addno(i1, i2)
8 Console.WriteLine("The Result is :" + Sum.ToString)
9 Console.ReadLine()
```

من السطر الاول الى السطر السادس اعلنا عن متغيرين رقميين لياخذا الرقمين الذان يدخلهما المستخدم وطبعا الدالة Read line هي الدالة نصية فيجب تحويل البيان النصي الى بيان رقمي عن طريق الدالة Cint

واذا اردنا ان نستدعي اي دالة كما هو في السطر السابع فاننا اعلنا عن متغير رقمي من نفس نوع الدالة ثم في السطر الثامن خصصنا له قيمة الدالة

ستقوم الدالة بارجاع قيمة الجمع بين المعاملين X,y الذان خصصنا لهما قيمة المتغيرين i1 , i2 وبعد اجرء عملية الجمع سترجع الدالة قيمة الجمع من خلال الكلمة Return فتخصص هذه القيمة وتخزن داخل المتغير Sum

في السطر التاسع طلبنا من الكونسل اظهار النتيجة للمتسخدم وعند تشغيل البرنامج سندخل قيمة الرقم الاول و الثاني ثم تجد ان البرنامج يظهر النتائج كما في الشكل الاتي

خلاصة ما سبق من الشرح

الفرق بين الدالة و الاجراء هو ان الدالة تقوم بارجاع القيمة للمتغير الذي ينشئه المبرمج اما الاجراء فهو ينفذ الامر دون الرجاع اي قيمة لاي متغير مثل الاجراء WriteLine موجود في الكنسل فالاجراء لا يرجع اي قيمة بينا الدالة Read line فتعيد القيمة التي يدخلها المستيخدم للمتغير ليتم معالجتها

كل ما اريده منك ان تقوم بقراءة هذا الفصل مرة اخرة واعد عمل الامثلة التي شرحتها مرة اخرى لكي يسهل عليك فهم هذا الفصل فهو فصل مهم جدا جدا جدا الى اقصى حد لانك اذا فهمته سوف يسهل عليك فهم الفصول القادمة

الفصل الرابع

التصنيفات و الكائنات

في الفصل السابق كنا قد شرحنا الاجراءات و الدوال على انها وحدات برمجية تساعد على بناء البرنامج حتى يمنع تكرار الكود ويسهيل اصلاحه

فاذا كتبان اجراءات كثيييييرة في البرنامج ووجدنا انفسنا امام كم هائل من الاجراءات فانة يتعين علينا ايجاد حل حتى يسهل علينا الوصول الى الاجراءات و الدوال ويسهل استدعاءها

هنا ياتي الحل في عمل تصنيف مشترك يجبع بين هذه الاجراءات ولكي وهذا التصنيف يعرف ب Class فيجمع بين الاجراءات و الدوال المشتركة في حل مشكلة معينة وعند استدعائه نقوم بعمل نسخة منه تسمى الكائن opjecte هذا الكائن يحمل كل الخصائص و الساليب الموجودة في التصنيف .

تأتي فكرة التصنيفات و الكئنات الى فكرة التصميم و التنفيذ بمعنى ان مهندس السيارات يضع كل مواصفات السيارة التي يريدها في ورقة ثم يسلمها للمصنع فتعتبر الورقه هي التصنيف الذي يضم كل الخصائص و الاساليب التي تخص السيارة فيقوم مدير المصنيع باعطاء امر تصميم سيارة على نفس المواصفات فتحول من سيارة على الورق الى سيارة ملموسة هذه السيارة هي الكائن مع تعديل في بعض القيم المعينة من الخصائص كلون السيارة وغيرها من المواصفات الفرعية التي يصممها المهندس لتكون قابلة للتعديل

مثال اخر اتذكر اننا شرحنا في الفصل الاول عن البيئة الرسومة او Windows Applicationوكنا كنا شرحنا امثلة عن صندوق النصوص او صندوق العناوين هذه الصناديق تعتبر وهي داخل صندوق الادوات عن تصنيفات وعند اخذها ووضعها على النموذج يكون اخذنا نسخة من التصنيف صندوق العنوان وعملنا منها كائن صندوق العناوين ونقوم بتغيير خصائصها بما يتلائم مع البرنامج

وسوف نشرح الان كيفية عمل التصنيفات و اضافة العناصر اليها وكيف استدعائها في هذا الفصل و الفصل القادم باذن الله

التصنيف

هو وحدة برمجية تقوم تجمع الاجرائات و الدوال والخصائص و الاحداث والتي هي تعتبر عناصر الكائن في البرمجة

ولكي نقوم بتصميم تصنيف علينا اتباع القاعدة الاتية

```
[Public , private]Class Class Name

Class Members

End Class
```

[Public , private] كما حددناها في الفصل السابق تحدد مدى الوصول اليها

Class Name اسم التصنيف وهي نفس شروط التسمية التي شرحنها في التغيرات

Class Members هي العناصر دخال التصنيف من الخصائص و الدوال و الاجراءات

وليكن الامر اكثر وضوحا دعنا نضع مثال بسيط عن التصنيفات تخيل لو انه طلب منك عمل برنامج عن مرتبات موظفين في شركة فبعد تحليلك قررت وضع قسم خاص في البرنامج لحساب مرتبات الموظفين فيتكون التصنيف من دوال لحساب المرتب الساسي و حساب عدد ايام الغياب ثم وحساب المكافئات

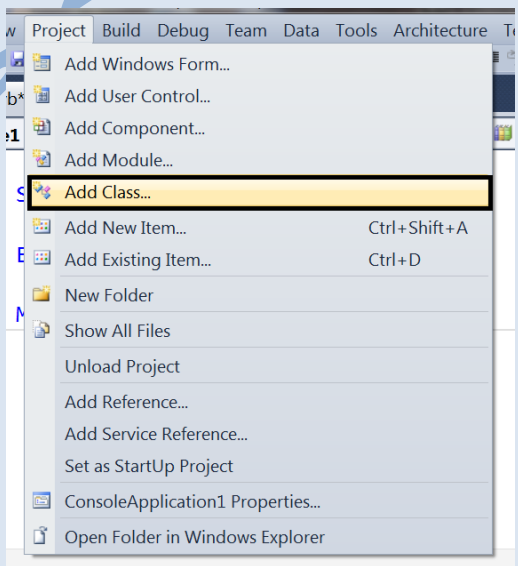
قبل ان نبدأ احب ان اوضح لك شيئ مهم بخصوص التطوير

ان عملية تطوير اي برنامج تتم على ثلاثة مراحل : **الاولى** هي اثناء انشاء التصنيفات ووضع العناصر بداخلها وهي يقوم بها مجموعة من المبرمجين المتخصصين

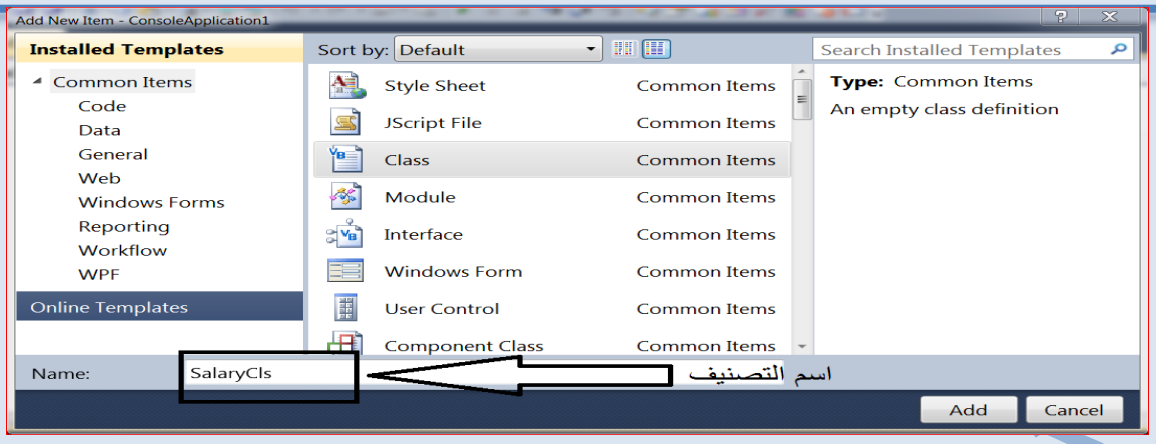
المرحلة الثانية هي مرحلة التطوير هي استدعاء هذه التصنيفات وانشاء الكائنات والتعامل مع العناصر التي بداخلها وتسمى هذه المرحلة Design time

المرحلة الثالثة هي مرحلة الاختبار ان نضع البرنامج في مرحلة الاختبار وتسمى Run time ونظر كيف يتعامل المستخدم العادي مع البرامج

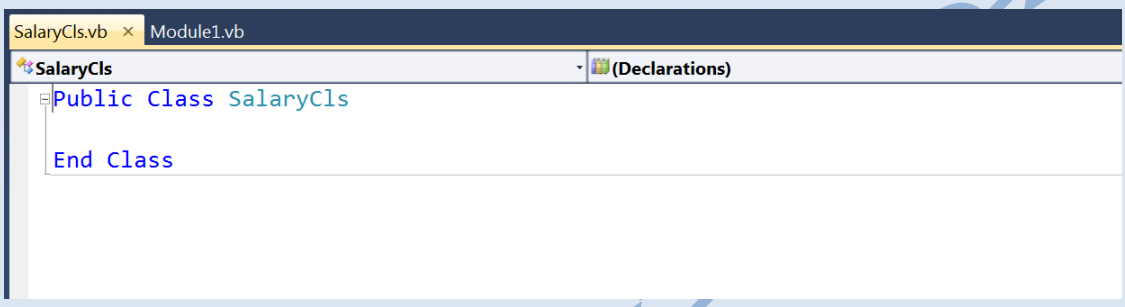
سنبدا الان بعمل التصنيف وسوف ننشئه ونختبره من خلال التطبيق كونسل فنفتح تطبيق كونسل جديد ثم من القائمة project نختار Add Calss كما هو موضح في الشكل الاتي



فيظهرلنا صندوق الحوار الخاص ليطلب منك اضافة اسم التصنيف فاكتب SalaryCls كما هو موضح في الشكل الاتي



فتظهر الشاشة الاتية



قبل ان نبداء بعمل التصنيف تعلى نحلل المشروع فنجد انه يتكون من مدخلات و مخراجات و معالجة

المدخلات

هي ما يقوم المستخدم بادخال ليتم حساب المرتب وهي

- المرتب المتعاقد عليه
- عدد ايام الغياب
- عدد مرات الخصومات حيث ان اذا ارتكب الموظف خطا ما فانه تحسب نقطة ليست في صالحه
- عدد مرات المكافئات هي عكس النقطة السابقة حيث انه تحسب نقط لصالح الموظف اذا قام بشئ يزيد من مكافئة الموظف

المخرجات

- صافي الدخل
- قيمة الخصومات
- قيمة المكافئات

المعالجة

- حساب المرتب بعد خصم ايام الغياب
- حساب قيمة الخصومات
- حساب قيمة المكافئات
- خساب صافي الدخل

وبناء على التحليل السابق فاننا نقول ان التصنيف مكون من العناصر الاتية

المدخلات	خاصية لادخال المرتب الذي تعاقد عليه الموظف مع الشركة
	خاصية لادخال عدد المخالفات التي ارتكبها الموظف
	خاصية لحساب عدد المكافئات التي حسبت للموظف
	عدد ايام الغياب
المعالجة	دالة لحساب المرتب الاساسي بحيث تطرح عدد ايام الغياب من المرتب المتعاقد عليه
	دالة لحساب قيمة المخالفات بحيث يتم ضرب قيمة المربت الاساسي نسبة الخصم وهي تساوي 3% وهي نسبة الخصم في لائحة الشركة ثم نضربها في عدد المخالفات
	دالة لحساب قيمة المكافئات المخالفات بحيث يتم ضرب قيمة المربت الاساسي نسبة الخصم وهي تساوي 3% وهي نسبة الخصم في لائحة الشركة ثم نضربها في عدد المكافئات
المخرجات	خاصية تحدد قيمة الخصومات
	خاصية تحدد قيمة المكافئات
	خاصية تحدد صافي الدخل

ومن العناصر السابق تبين ان لدينا في التصنيف اما يالي

1- خصائص وهي سوف نشرحها شرح كامل في السطور القادمة
2- اساليب وهي تتمثل في الدوال و الاجراءات

لنبدا العمل سوف نكتب الدالة الاولى اكتب الكود الاتي داخل التصنيف

```
Private Function MainSalary(ByVal Salary As Single, ByVal AppsDay As Integer) As Single
    Dim salPerDay As Single 'متغير يقوم بحساب قيمة عمل اليوم
    salPerDay = Salary / 30 ' قسمة المرتب على عدد الايام الشهر وهي ٣٠ يوم
    Dim DescountS As Single ' متغير لحساب الخصومات
    DescountS = salPerDay * AppsDay ' الراتب في اليوم مضروب في عدد ايام الغياب
    Return Salary - DescountS ' نطرح المخصوم عن الاساسي
End Function
```

والان نكتب الدالة الثانية وهي تخص حساب المخالفات

```
Private Function Descount(ByVal DescountCount As Integer, ByVal Salary As Single) As Single
    Const Percentage As Decimal = 0.03 ' نسبة الخصومات
    ' هو حساب الخصومات عن طريق ضرب الراتب الذي تم حسابة بعد خصم ايام الغياب
    'في نسبة الخصم ثم ضربه في عدد مرات المخالفة
    Return (Salary * Percentage) * DescountCount
End Function
```

لزيادة الايضاح في هذه الدالة هي تحسب خصومات المرتبات فنسبة الخصم هي نسبة افتراضية موجودة في اللائحة في الشركة التي نقوم بعمل البرنامج لها طبعا هذا **النظام تخيلي للفهم فقط**

فنفترض ان نسبة الخصم في اللائحة تساوي 3 في المائة

ثم نضرب المرتب الذي حسابناه في الدالة التي تحسب المرتب الاساسي في نسبة الخصم فيكون نسبة المرتب المخصوم في المخالفة الواحدة فنضربها في عدد الخصومات التي نتدخلها من المعامل DescountCount مع العلم ان المعامل Salary هو المرتب بعد خصم ايام الغياب

نكتب الان في الكود الاتي دالة المكافئات

```
Private Function Incentive(ByVal incentCount As Integer, ByVal Salary As Single) As Single
    Const Percentage As Decimal = 0.03 ' نسبة الخصومات
    ' هو حساب المكافئات عن طريق ضرب الراتب الذي تم حسابة بعد خصم ايام الغياب
    'في نسبة الزيادة ثم ضربه في عدد مرات المكافئة
    Return (Salary * Percentage) * incentCount

End Function
```

تكون نفس الطريقة و الان نحسب صافي الراتب بجمع دالة المكافئات مع المرتب الاساسي ثم طرحه من الخصومات ولكن قبل ان نبدا في هذه الدالة اريد ان اشرح ما هي الخاصية

الخاصية

عنصر من عناصر التصنيف تقوم بتخزين البيانات مثل المتغير ولكن تختلف الخاصية عن المتغير هي ان الخاصية تخزن البيانات ولكن تقوم بالتعديل البيانات قبل التخزينها او استخدامها فتخيل اننا سنخزن قيمة المرتب الاساسي المتعاقد عليه فيه فاننا سوف نقوم بالاعلان عن متغير كلاتي

```
Public MySal as Single
```

كلمة public هي كلمة محددة الوصول والتي قلنا اننا سنشرحها عندما نتكلم عن الاجراء او الدالة هذه الكملة تحدد مدى قدرتنا للوصول على المتغير او العنصر للقرائة من بيانات او التعديل عليها فكلمة public تعني اننا نستطيع الوصول الى المتغير من اي مكان في المشروع ولكن كلمة private تعني ان العنصر خاص فقط داخل التصنيف كما يوجد

كلمات اخرى غيرها سوف اشرحها ولكن بعد ان تتعلم مبادئ البرمجة الكائنية لانها مرتبطة بها اريدك ان تعرف فقط كلمتين هما private تعني ان العنصر عنصر خاص لا يجوز ان نصل اليه من خارج التصنيف ولكن نستطيع قرائته من داخل التصنيف الذي هو فيه وكلمة public تعنيا ننا نستطيع الوصول للعنصر من اي مكان في التصنيف و خارجه

نعود الان للمتغير الذي شرحناه وهو يقوم بتخزين البيانات و لكن افترض انك ادخلت رقم سالب اقل من وتقصد به هو بيان عن المرتب الاساسي المتعاقد عليه .... طبعا هذا البيان خاطئ وسينتج عنه نتائج خاطئة لذا يجب قبل ان نخزن البيان ان نجري فحص عليه قبل تخزينه ... هنا ياتي دور الخاصية فيه تقوم بهذا الدور .... واذا اردنا ان تتشئ خاصية علينا اتباع القاعدة الاتية

```
[Private – public] VariableName As Type

[public – private] property PropertyName (Agrument)As Type

Get

Statements

Return VariableName

End Get

Set(Value As Type)

Statements

VariableName = Value

Statements

End Set

End Property
```

[Private – public]توضح مدى الوصول للمتغير ويفضل ان يكون خاص

VariableName اسم المتغير الذي سنستخدمه في تخزين البيانات

Type النوع ويجب ان يكون من نفس نوع الخاصية

[public – private] توضح مدى الوصول للخاصية

Property Name اسم الخاصية المستخدمة

Agrument المعامل التمرير

Type نوع الخاصية

في القسم الخاص بـGet , EndGet فهو القسم الذي يشبه الدالة بحيث يقوم بعمل الحسابات ثم يرجع البيانات في كلمة Return وتلاحظ كلمة *VariableName* فهي تعني اننا نقراء القيمة التي المتغير الوسيط الذي اعلنا عنه قبل الاعلان عن الخاصية

في قسمSet , EndSet يشبه الاجراء ولكن يجب ان تخزن البيانات الموجودة باستخدام الامر

VariableName = Value بحيث ان كلمة Value هي عبارة ان المعامل الموجود دخال البلوك Set كلاتي Value As Type بحيث ان Type هو نفس النوع الخاص بالخاصية ولكي تستطيع فهم الخاصية بشكل اكثر تعالى نعود للمثال الذي نحن فيه فننشئ خاصية تخص المرتب الاساسي المتعاقد عليه فنكتب الخاصية على الشكل الاتي

```
Private SalContact As Single ' المتغير الذي نخزن فيه الخاصية
Public Property ContactSalary As Single
    Get
        Return SalContact  ' هنا ناخذ قيمة المرتب
    End Get
    Set(ByVal value As Single)
        If value > 0 Then 'اذا كان المرتب اكبر من الصفر
            SalContact = value ' يتم تخزين القيمة المدخلة على الخاصية
        Else
            SalContact = 0 'تكون القيمة تساوي الصفر اذا كان المتغير اصغر من الصفر
        End If
    End Set
End Property
```

هكذا يكون شكل الخاصية ولكن عند استدعاء القيمة التي في الخاصية فاننا نتعامل معها مثلما نتعامل مع المتغير مع العلم اننا نستعدي اسم الخاصية وليس اسم التغير الوسيط الذي نخزن فيه البيانات

يعني اننا نريد قراءة قيمة المرتب المتعاقد علي فاننا نستدعي الخاصية ContactSalary وليس المتغير SalContact ونستدعيه الخاصية مثل استدعاء المتغير تماما مع اختلاف بسيط هو لو كان للخاصية معامل *Agrument* فاننا يجب اثناء استدعاء الخاصية نقوم بادخال قيمة للمعامل

ولان نقوم بانشاء خاصية اخرى لاخذ قيمة المرتب بعد خصم ايام الغياب احب ان اوضح شئ هام اننا في بعض الاحيان نجد ان هناك خصائص لا تستطيع ادخال بيانات عليها فهي للقراءة فقط لذا فاننا نتسخدم الجزء الخاص فقطبGet لذا يجب استجدام كلمة readonly قبل كلمة property كما سنوضح في الكود الاتي عند انشاء خاصية لاعطاء قيمة المرتب بعد خصم ايام الغياب فنكتب الخاصية الاتية

```
Public ReadOnly Property MySalary As Single
    Get
        Return MainSalary(ContactSalary, Appse

    End Get
End Property
Public Appd As Integer
Public Property AppsentDay As Integer
    Get
        Return Appd
    End Get
    Set(ByVal value As Integer)
        Appd = value
    End Set
End Property
```

هنا تلاحظ اننا انشأنا خاصيتين واحد لادخال ايام الغياب واخرى للاخذ منها المرتب ولكن تلاحظ الخاصية التي تستخدم لتخزين ايام الغياب وهي الخاصية AppsentDay هي خاصية رقمية نستطيع ان نخزن عليها ايام الغياب ونستدعيها ونقرا منه القيمة المخزنة فيها بينما في الخاصية MySalary هذه الخاصية تعتبرللقراءة فقط يعني لا استطيع تخصيص اي قيمة لها وتلاحظ اننا قمنا باستخدام الدالة التي تحسب المرتب MainSalary و التي بها معاملين Salary الخاص بالمرتب المتعاقد عليه و ادخلنا عليه قيمة المرتب المتعاقد عليه وهي القيمة الماخوذة الخاصية ContactSalary و المعامل Appsday المستخدم لادخال يام الغياب قمنا بادخال عليه عدد ايام الغياب من الخاصية AppsentDay

وبذلك تلاحظ اننا نتستطيع ان نستدعي الدوال لاستخدام قيمتها التي تقوم بارجاعها وتخزينها داخل الخصائص

وكما يوجد خاصية للقراءة فقط يوجد خاصية للكتابة فقط وهي WriteOnly وهذه الخاصية تستخدم قفط Set كما هو موضح في الشكل الاتي مع العلم انه خارج المثال الذي نشرحه

```
Dim Var As Single
Public WriteOnly Property SSss As Single
    Set(ByVal value As Single)
        Var = value
    End Set
End Property
```

ولان نعود للمثال وندخل باقي العناصر .

سوف ندخل خاصيتين للقراءة فقط واحدة لاعطاء قيمة المستحقات او المكافئات و اخرى لخصومات

ثم نحسب صافي الدخل عن طريق دالة واعطائها لخاصية اخرى للقرائة فقط فيكون الكود كلاتي

```
Private descoun As Integer
Public Property DescountMores As Integer
    Get
        Return descoun
    End Get
    Set(ByVal value As Integer)
        descoun = value
    End Set
End Property
```

```
Public Property incentiveMores As Integer
    Get
        Return incentcoun
    End Get
    Set(ByVal value As Integer)
        incentcoun = value
    End Set
End Property
```

```
Public ReadOnly Property Descounts As Single
    Get
        Return Descount(DescountMores, MySalary)
    End Get
End Property
Public ReadOnly Property incentives As Single
    Get
        Return Incentive(incentiveMores, MySalary)
    End Get
End Property
Public ReadOnly Property Netincome
    Get
        Return (MySalary + incentives) - Descounts
    End Get
End Property
```

في الكود السابق لدينا 5 خصائص الخاصية الاولى تسمى discount mores وهي تستخدم في تخزين عدد مرات الاخطاء التي يخطئ فيها الموظف

اما incentiveMore فهي تستخدم في تخزين عدد مرات المكافئات

اما الخاصية Descounts فهي للقراءة فقط تستخدم في حساب قيمة الخصومات التي ستخصم من المرتب وذلك باستدعاءا الدالة Descount التي بها معاملين فقمنا بتخصيص في المعامل الاول عدد مرات المخالفة من الخاصية DescountMore اما المعامل الثاني الذي نستخدمه لادخال قيمة المرتب فقمنا بادخال قيمة الخاصية mysalary

اما الخاصية incentive فهي مثل الخاصية Descounts ولكن تختلف عنها في اننا استدعينا الدالة incentive التي تحسب المكافئات وقمنا تخصيص قيمة الخاصية incentiveMore للمعامل الاول و قيمة الخاصية Mysalary للمعامل الثاني

واخيرا الخاصية NetIncome وهي خاصية للقراءة فقط فتلاحظ اننا استخدمناها مثل الدالة حيث اننا قمنا بحساب قيمة صافي الدخال الذي سيقبضه الموظف فقمنا باضافة الحوافز من الخاصية incentive على المرتب الذي اخذنا قيمته من الخاصية mySalary ثم طرحنا منهما قيمة الخصومات الذي نستدعيها من الخاصية Descounts

ولن تجد هناك اختلاف بين الخاصية المستخدمة للقراءة فقط و الدالة من الناحية النظرية ولكن من الناحية العملية فهي تاتي على حسب تحليلنا للبرنامج هل سنحتاج للدالة ام للخاصية المستخدمة للقراءة فقط

وبذلك نكون قد انتهينا من عمل التصنيف واضفنا اليه العناصر يبقى علينا الان ان نستدعيه لاستخدام عناصره في عملنا البرمجي اريد ان اذكرك بشيء هام شرحناه في اول هذا الفصل وهو الفرق بين التصنيف و الكائن object *والتي تترجم في كتب اخرى بالهدف .....*

قلنا ان التصنيف يشبه رسم عناصر السيارة على ورق ولكن عندما يتم وضعها على ارض الواقع تتحول الى كائن له خصائص تختلف عن كائن اخر ينشئ من نفس التصنيف او من نفس الفئيه او الفصيله وكلها كلمات تترجم كلمة Class

ولكن في لغة البرمجة مثلا الزر صاحب اللون الاحمر ينشئ من نفس الفئة او التصنيف الذي ينشئ الالزر الازرق ولكي يتم عملية الانشاء فانه يوجد دالة تسمى دالة الانشاء و المعروفة بدالة New

دالة الانشاء NEW

عندما تقوم بعمل انشاء للدالة عليك اولا بالاعلان عن متغير يحمل نفس نوعه عن التصنيف

```
Dim I as SalaryCls
```

ثم نكتب الكود الاتي

```
I = new Salary ()
```

طبعا SalaryCls هو اسم التصنيف الذي كنا نكتبه في السطور السابقة وهو التصنيف الذي انشاءنا لحساب مرتبات الموظفين

يمكن عمل طريقة مختصرة وهي

```
Dim I as new SalaryCls()
```

وبذلك نكون قد انشانا الكائن يمكنك ان تجد العناصر المكتوبة داخل التصنيف ولكن كن على علم انه لن اظهر الا عناصر المكتوبة بكلمة الاستدعاء public لان كلمة private لا تسمح للوصول اليها الا من داخل التصنيف فقط

وللمزيد من الايضاء دعنا نعود للكنسل ونكتب الكود دخال الدالة Main

```
Dim sal As Single
Dim App, desc, incc As Integer
Console.WriteLine("Enter the Salary:")
sal = CSng(Console.ReadLine)
Console.WriteLine("Enter the Nomber of Appcent Day:")
sal = CInt(Console.ReadLine)
Console.WriteLine("Enter the Descount more:")
sal = CInt(Console.ReadLine)
Console.WriteLine("Enter the incentive more:")
sal = CInt(Console.ReadLine)
```

1

```
Dim salar As New SalaryCls
salar.ContactSalary = sal
salar.DescountMores = desc
salar.AppsentDay = App
salar.incentiveMores = incc
```

2

```
Console.WriteLine("main Salary :" + salar.MySalary.ToString)
Console.WriteLine("discounts :" + salar.Descounts.ToString)
Console.WriteLine("incentives :" + salar.incentives.ToString)
Console.WriteLine("net incom :" + salar.Netincome.ToString)
Console.ReadLine()
```

3

في الكود السابق الذي كتبناه في الدالة main ما هو الا كود لاستدعاء التصنيف وقراءة عناصره منه او التعديل فيها فتابع معي شرح الكود

من السطر الاول الى السطرالعاشر المربع رقم(1) : قمنا بعمل اعلان عن متغيرات ثم خصصنا القيم والتي يدخلا المستخدم ففي هذا المربع قمنا بالطلب من المستخدم بادخال قيم المرتب المتعاقد عليه وعدد ايام الغياب و عدد مرات المخالفات وعدد مرات المكافئات

من السطر الحادي عشر الى السطر السادس عشر (2): قمنا بانشاء الكائن SalaryCls باستخدام المتغير Salar ثم خصصنا القيم التي ادخلها المستخدم من قيمة المرتب المتعاقد عليه وعدد ايام الغياب و عدد مرات المخالفات وعدد مرات المكافئات الى خصائصها بحيث اننا استخدمنا المتغيرات التي اعلنا عنها في المربع الاول ثم استخدمناها بتخزين القيم التي يدخلها المستخدم ثم استدعينها في المربع الثاني لتخصيص القيم التي التي نريد ان نخزنها للخصائص التي تناسبها فمثلا في الكود السابق اعلنا عن المتغير Sal لتخزين قيمة المرتب المتعاقد عليه ثم استدعينها لكي نخصص قيمته في الخاصية ContactSalary و التي اسدعيناها من الكائن Salar

ولكي نستدعي اي عنصر من الكائن فاننا نستدعي المتغير الذي ينشئ االكائن ثم ننقر على النقط (.) ثم نستدعي العنصر الذي نريده سواء كان خاصية او دالة او اجراء كما هو موضح في الكود الاتي

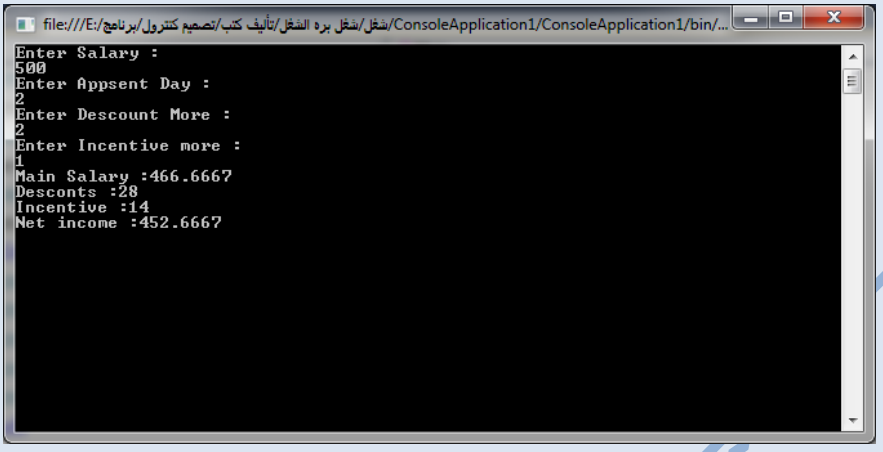
```
Salar.ContactSalary
```

Salar اسم المتغير استخدمناه في انشاء الكائن SalaryCls

ContactSalary مقصود بها العنصر الذي نريد الوصول اليه وهذه العنصر هو خاصية

اما باقي الاسطر من الكود قمنا باظهار النتائج من المرتب الاساسي بعدما طرحنا منه ايام الغياب من الخاصية MySalary واظهرنا اجمالي الخصوماتDescount واجمالي المكافئاتincetive و صافي الدخل

فاذا شغلنا البرنامج سنجد الشاشة تظهر على مرات لاستقبال البيانات ثم تظهر الناتج مرة واحد فيظهر الشكل النهائي كلاتي



اضافة قيم افتراضية لدوال الانشاء

يمكنك اضافة قيم افتراضية اثناء انشاء الكائن مثل الدوال فقم باضافة الدالة الاتية داخل التصنيف SalaryCls

```
Public Sub New(ByVal Sale As Single, ByVal App As Integer, ByVal DesC As Integer, ByVal inC As Integer)
    Me.ContactSalary = Sale
    Me.DescountMores = DesC
    Me.AppsentDay = App
    Me.incentiveMores = inC
End Sub
```

في هذه الدالة قمنا بكتابة دالة انشاء new وبها عدة معاملات التي تدخل قيم افتراضية للتصنيف وهي المرتب المتعاقد عليه و عدد ايام الغياب وغيرها من المدخلات التي ادخلناها في المثال ، داخل هذه الدالة اعطينا كل الخصائص القيم يجب ان تاخده ولكن ما معنى كلمة Me هي كلمة تعني نفس التصنيف الذي نستخدمه لانه لا يسمح ان نستخدم اسم النصنيف ثم اسم الدالة او الخاصية لكن يجب ان

عند انشاء هذا الكائن عن طريق دالة الانشاء هذه اكتب الكود الاتي في الاجراء main

```
Dim sal As Single
Dim apps, DesC, inscC As Integer
Console.WriteLine("Enter Salary :")
sal = CSng(Console.ReadLine)
Console.WriteLine("Enter Appsent Day :")
apps = CInt(Console.ReadLine)
Console.WriteLine("Enter Descount More :")
DesC = CInt(Console.ReadLine)
Console.WriteLine("Enter Incentive more :")
inscC = CInt(Console.ReadLine)

Dim salar As New SalaryCls(sal, apps, DesC, inscC)

Console.WriteLine("Main Salary :" + salar.MySalary.ToString)
Console.WriteLine("Desconts :" + salar.Descounts.ToString)
Console.WriteLine("Incentive :" + salar.incentives.ToString)
Console.WriteLine("Net income :" + salar.Netincome.ToString)
Console.ReadLine()
```

اختصرنا ه اسطر لسطر واحد

هذا الكود شرحنا سابقا ولكن في المربع الثاني اختصرنا كل الاسطر في المربع الثاني في سطر واحد بحيث اننا انشأنا كائن واستخدمنا الدالة new التي بها قيم افتراضية ثم استخدمنا في المتغيرات التي اعلنا عنا في المربع الاول والتي استخدمناها في تخزين البيانات التي ادخالها المستخدم في كل معامل بما يناسبه من قيم

ولكن اذا اردنا ان ننشئ الكائن بدون ادخال قيمة افتراضية مع ترك فرصة اخرى لانشاء الكائن مع ادخال القيم الافتراضية بمعنى ادق افترض اننا نريد انشاء كائنين احدهما ينشئ الكائن مع ادخال قيم الفتراضية واخر ينشئ بدون ادخال قيم افتراضية من نفس التصنيف لذا يجب علينا كتابة دالة نشاء اخرى وتكون فارغة كهذه الدالة ننشئها داخل التصنيف SalaryCls

```
Public Sub New()

End Sub
```

هنا ننشئ الدالة زائدة على دالة الانشاء *new* وهذا نسمية التحميل الزائد overload الذي سنشرحه بالتفصيل في الفصل القادم

متغيرات الحالة ومتغرات مشتركة

ملحوظه هامة جدا

العناصر التي تستدعى عن طريق انشاء متغير تسمى عناصر حالة يعني تتعامل مع حالة التغير التي تنشاء من خلاله

فلو انك انشات متغيرين x, y لعمل التصنيف SalaryCls وعطيت لكل عنصر مرتب مختلف عن الاخر فالمتغير X اخد مرتب 500 جنيه و Y 400 فان خاصية صافي المرتب التي تخص المتغير x ستختلف عن المتغير y وهذه تسمى عناصر الحالة

ولكن هناك عناصر مشتركة وهي عناصر لا تتغير قيمتها مهما كان حتى يتوقف البرنامج وتسمى بالعناصر المشتركة او الاستاتيكيه ولمزيد من الايضاح دعنا نعلن عن تصنيف اخر لعمل مرتبات **وانصحك بحفظ المثال الذى قمنا بشرحه سابقا لاننا سنحتاجه خلال الفصول القادمة**

ولكن سوف نشرح الان عن العناصر المشتركة ولكي تستطيع الفهم سوف ننشئ تصنيف جديد يسمى SaleCls ويتعامل مع المبيعات ونضيف اليه عناصر مشتركة و عناصر حالة في الشكل الكود الاتي

```
Public Function TotalSale(ByVal Sales As Integer) As Single
    Sales = Sales + 1
    nnsales = (Sales * 20) + nnsales

    Return Sales

End Function
Private Shared nnsales As Single
Public Shared Function GrossMargine() As Single
    Return nnsales - 100
End Function
```

هنا عملنا تصنيف لحساب المبيعات فالدالة الاولى دالة حالة مهمتها ان تحسب اجمالي قيمة المبيعات اما الدالة الثانية فهي الدالة المشتركة بحيث اننا نكتب كلمة Share قبل كلمة function لنعلن ان هذه الدلة دالة مشتركة ومهمتها هي ان تقوم بعمل حساب للارباح

ثم اعلنا عن متغير خاص وسيط بينهما وهو المتغير nnsale وتلاحظ انه متغير مشترك به كلمة Share قبل اسم المتغير

سنعود الى دالة الحالة total Sale لشرح محتوياتها فهي تحتوي على معامل رقمي اسمه Sales هذا المعامل استدعيناه في السطر الاول للدالة واضفنا اليه رقم 1

اما في السطر الثاني قمنا بعمل معادلة هي ان نضرب قيمة المتغير Sale في 20 ثم اضفنا اليه قيمة المتغير nnsale ثم خصصنا ناتج المعادلة للمتغير nnsale

ثم ارجعنا قيمة Sale باستخدام الدالة return

اما في الدالة المشتركة GrossMargine قمنا ارجاع قيمة المتغير nnsale مطروح منه 100

و الان سوف نستدعي الدالتين وننظر ما الفرق بينهما انتقل الى الدالة main لتطبيق كنسل واكتب الكود الاتي

```
Sub Main()
1   Dim sale1 As New SaleCls
2   Dim sale2 As New SaleCls
3   Console.WriteLine("Sale Variable From Sale 1 " + sale1.TotalSale(10).ToString())
4   Console.WriteLine("From Share Function :" + SaleCls.GrossMargine.ToString)
5   Console.WriteLine("Sale Variable From Sale 2 " + sale2.TotalSale(10).ToString())
6   Console.WriteLine("From Share Function :" + SaleCls.GrossMargine.ToString)
7   Console.ReadLine()

End Sub
```

في السطر الاول و الثاني اعلنا عن متغيرين لانشاء SaleCls

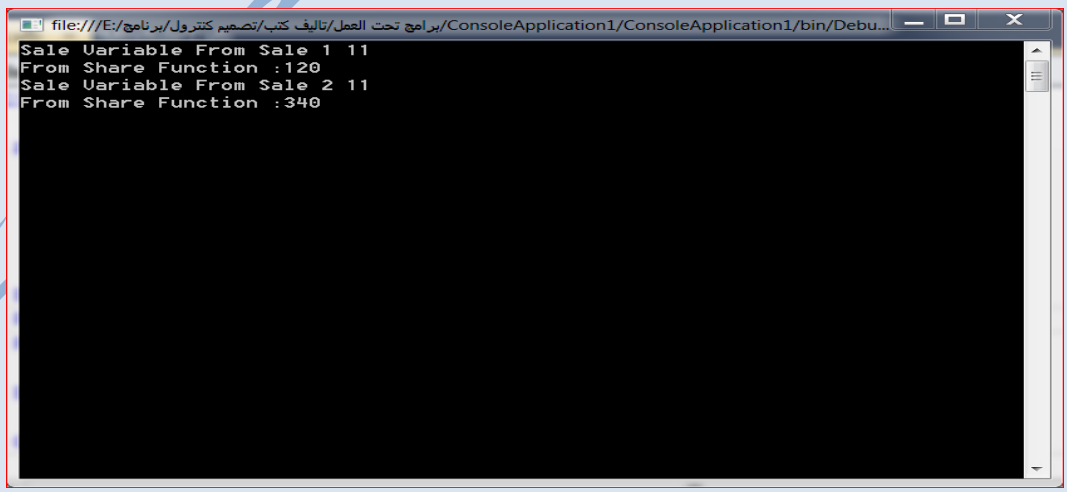
في السطر الرابع طلبنا من الكنسل اظهر نتيجة الدالة TotalSale من المتغير Sale1

في السطر الخامس طلبنا من الكنسل اظهر الناتج من الدالة المشتركة و التي تستدعى باسم التصنيف SaleCls

في السطر السادس طلبنا من الكنسل ظهار نتيجة الدالة TotalSale من المتغير Sale2

في السطر السابع طلبنا من الكنسل ظهار الناتج من الدالة المشتركة

اذا شغلنا المتغير سوف نجد الناتج كلاتي



تلاحظ ان دالة الحالة تفقد قيمتها اول تتغير قيمتها من متغير لاخر المتغير بينما المتغير المشترك و الدوال المشتركة لم تفقد قيمتها من متغير لاخر فتتغير قيمته من 120 الى 340

بعد ان انشاءا الكائن و تعلمنا كيف نستدعي منه العناصر بقي علينا ان ضيف على الكائنات عناصر اخرى مهمة جدا وهي الاحداث و التفويضات ومهمتها استدعاء الاجراءات و الدوال بطريقة مرنة

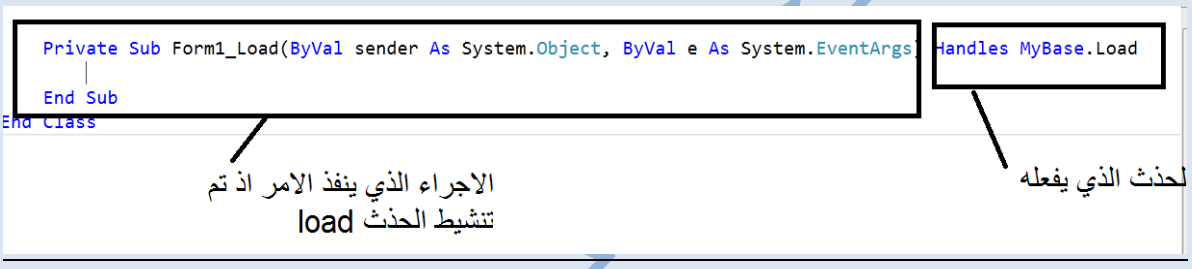
الحدث EVENT

تعلمنا في الفصل السابق ان انا لدينا نوعين من الاجراءات اجراء نستدعيه باسمه وهو الاجراء الفرعي و اجراء يتم استدعاءه بتنشيط حدث وهو ما نحن بصدده بحيث اذا تم تنشيط الحدث يستدعى وراءه الاجراء ولكن ما هو الحدث

الحدث هو ما يشبه الانذار او شرارة يتم من ورائها رد فعل واتخاذ اجراء معين ، تخيل معي الطائرة او السيارة اذا حدث عطل ما فانه يعطي انذار ينبه السائق فبالتالي ياخذ الاجراء المناسب لحل العطل

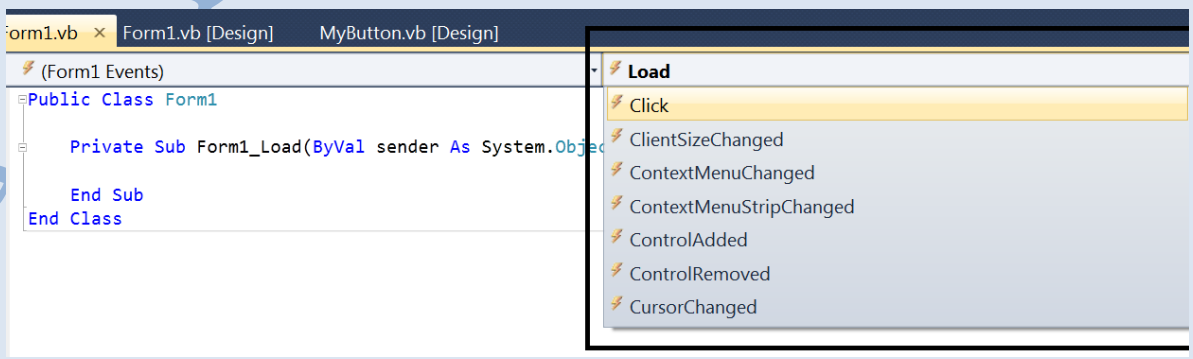
كذلك الحدث في البرمجة فهو شرارة ينتج من ورائها استدعاء اجراء معين فمثلا اذا قام شخص بالنقر على الماوس فانه يعطي اشارة نقر هذه الاشارة تعني الحدث و بالتالي اذا كان هناك اجراء معد ليكون رد فعل امام النقر فانه سيتم استدعاءه لنفيذ الامر كما تعمنا في الفصل الاول اننا اذا برمجنا الاجراء المحث لينفذ اوامر عنما ينقر المستخدم الماوس فانه ينفذ ما بداخله

افتح مشروع جديد ليكون من نوع Windows Application لنتذكر كيف يتم رد الفعل اذا نقرت مرتين على النموذج ستجد الشاشة الاتية



فاذا قام المستخدم بتحميل النافذة فانه يتم تنشيط الحدث فيفعل الحدث ويطلق شرار فيتم استدعاء الاجراء ليكون له رد فعل وينفذ الاوامر المطللوب تنفيذها

واذا اردت عمل اجراء اخر ليكون رد فعل عندما نقر المستخدم على النافذة فافتح القائمة التي اعلى يمني شاشة الكود واختر الحدث Click



فتجد الاجراء يكون جاهز كلاتي

```
Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click

End Sub
```

ولكن هذا لا نريد ان نتكلم عنه فهذا ما يقوم المطور الذي يستخدم التصنيف الذي قمت بتصميمه فتخيل انك تصمم التصنيف الخاص بالمرتبات واعطيته لزميلك المسئول عن ربطه في قاعدة البيانات فيكون زميلك هو مستخدم التصنيف فيقوم هو بانشاء الكائنات ويستدعي الاجرائات فنحن مطورين نستخدم النافذة ويندوز التي طورتها مايكروسوفت فنحن لم نقم بتصميم النوافذ ولكن نحن نعمل على تفعيل الحدث ليستدعي الاجراء الذي نريد تنفيذه ..... نحن نريد ان نتعلم كيف ننشئ الحدث ونعده لكي يقوم مستخدمي التصنيف بتنشيط الحدث ومنه يستدعي الاجراء الدي يريد ان ينفذه عند تنشيط الحدث

الان نقوم باضافة الحدث داخل التصنيف فما رايك بالعودة للتصنيف الخاص بالمرتبات Salarycls الذي انشأناه فنقوم باضافة حدث يتم تنشيطه عندما يكون صافي المرتب اقل من 10 % المرتب المتعاقد عليه

ولكي نضيف الحدث داخل التصنيف فنتبع الخطوات الاتية

1. الاعلان عن الحدث
2. وضع الحدث في المكان الذي تريده ان ينشط فيه
3. استدعاء الاجراء عند تنشيط الحدث

الخطوة الاولى

الاعلان عن الحدث

```
[Publc - private]Event Event Name (Argument ) AS Type
```

[*Publc - private*] : تحدد مدى الوصول الى الحدث

Event Name : اسم الحدث

Argument: المعاملات التي نستخدمها في استدعاء الاجراء او الدالة ويجب ان تاخذ في الاعتبار ان الحدث الذي به 3 معاملات لا يستدعي الى اجراء او دالة بها ثلاثة معاملات ويجب ان تكون كل انواع المعاملات مشابهة تماما لانواع المعاملات التي في الاجراء او الدالة بمعني لو اعلنا عن حدث به معامل رقمي ونص و منطقي يجب ان تكون المعاملات في الاجراء الذي نستدعيه بنفس النوع وبنفس الترتيب في الانواع فلا يجوز ان نستعدي اجراء به معاملات مختلف ترتيبها عن الترتيب الذي اعلنا عنه في الحدث

Type : نوع الحدث ويجب ان تعلم انك لا تستطيع استدعاء دالة نوعها نصي مع حدث من نوع رقمي او مع اجراء واذا اردت ان يكون مهمة الحدث استدعاء اجراء به معامل واحد فيجب ان يكون عند الاعلان عن الحدث به معامل واحد ولا تكتب اي نوع في الجزء الخاص بالنوع

للايضاح ادخل الكود الاتي في التصنيف

```
Public Event LawSalary()
```

فتجد اننا جعلنا الحدث lawSalary عام لنتمكن من الوصول اليه وهذا الحدث عندما ينشط سوف سيتدعي اجراء بدون معامل

الخطوة الثانية اعداد الحدث لينكون منشط

والان يجب ان يكون هناك تنشيط لهذا الحدث ليطلق الشرارة ولعمل ذلك يجب اتباع الخطوات الاتية

```
Raise Event Event Name ( Value)
```

Event Name: هنا نستدعي الحدث الذي نريد ان نشع الشرارة

Value اعطاء قيمة دخل في المعامل وسوف اشرح ذلك بالتفصيل

نطبق هذا على المثال فنفعلها في الخاصية NetIncome سوف نعدل الخاصية الى الكود الاتي

```
Public ReadOnly Property Netincome
    Get
        Dim ss As Single
        ss = (MySalary + incentives) - Descounts
        If ss < ContactSalary * 0.1 Then
            RaiseEvent LawSalary()
        End If
        Return ss
    End Get
End Property
```

في الكود السابق قمنا باعلان عن متغير لياخذ هو حسابات صافي الدخل

في السطر الذي يليه قمنا باختبار المتغير اذا كان اقل من 10 في المائة من الدخل المتعاقد عليه فاذا كان اقل فانه يتم نشيط الحدث

ثم قمنا بتخصيص قيمة SS الى الخاصية باستخدام الكلمة return

الخطوة الثالثة استدعاء الاجراء

بعد ان نشطنا الحدث يجب علينا انشاء الاجراء ويتم استدعائه عندما ينشط الحدث ولعمل ذلك هناك طريقتين لعمل ذلك

طريقة معروفة وهي الطريقة التي شرحناه في الاول في تطبيقات النوافذ والتي سنبدا بها الان ويتم ذلك من خلال خطوتين

الاولى هي الاعلان عن المتغيريقوم بانشاء كائن ولكن بطريقة مختلفة وهي كلاتي

```
[Public-private] WithEvent Varible Name AS Class Name(Value )
```

هذه القاعدة تشبه تماما قاعدة انشاء الكائن ولكن تختلف معها كلمة With Event هذه الكلمة تخبر انه عندما يتم انشاء الكائن واضافة عناصر الاحداث ليتم استدعاءها لانك اذا لم تكتب هذه الكلمة ستلاحظ انه لا يسمح لك استدعاء الحدث الا بوجود هذه الكلمة **وهنااااااك ملحوظه هاااااامة لا يسمح بهذا الاعلان داخل اي عنصر سواء كان اجراء او دالة او خاصية يعلن عنه في قسم الاعلانات العاملة وهو اي مكان دخال التصنيف وخارج العناصر**

```
Public WithEvents dd As New SalaryCls()
```

هذا هو الكود المطلوب للاعلان عن متغير ينشئ الكائن من نوع SalaryCls مع وضع الاحداث ضمن العناصر التي يمكن استدعاءها

الخطوة الثانية هي انشاء الاجراء الذي سوف نستدعيه عند تنشيط الحدث فاكتب الكود الاتي خارج الاجراء main

```
Sub mm()
    Console.BackgroundColor = ConsoleColor.White
    Console.ForegroundColor = ConsoleColor.Black
    Console.WriteLine("Low Salary")
End Sub
```

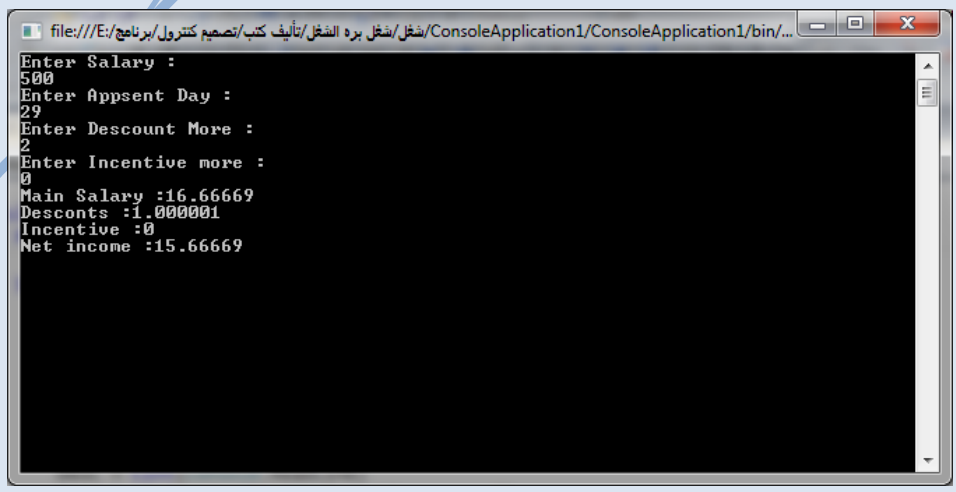
في الكود السابق قمنا في السطري الاول و الثاني بتغيير لون و خلفية الخط بحيث يظهر الانذار بلون مختلف

فيصبح الكود كله في قسم الكنسل هكذا

```
Module Module1
    Public WithEvents dd As New SalaryCls()

    Sub Main()
        Dim sal As Single
        Dim apps, DesC, inscC As Integer
        Console.WriteLine("Enter Salary :")
        sal = CSng(Console.ReadLine)
        Console.WriteLine("Enter Appsent Day :")
        apps = CInt(Console.ReadLine)
        Console.WriteLine("Enter Descount More :")
        DesC = CInt(Console.ReadLine)
        Console.WriteLine("Enter Incentive more :")
        inscC = CInt(Console.ReadLine)
        Dim salar As New SalaryCls(sal, apps, DesC, inscC)
        Console.WriteLine("Main Salary :" + salar.MySalary.ToString)
        Console.WriteLine("Desconts :" + salar.Descounts.ToString)
        Console.WriteLine("Incentive :" + salar.incentives.ToString)
        Console.WriteLine("Net income :" + salar.Netincome.ToString)
        Console.ReadLine()
    End Sub

    Sub mm()
        Console.BackgroundColor = ConsoleColor.White
        Console.ForegroundColor = ConsoleColor.Black
        Console.WriteLine("Low Salary")
    End Sub
End Module
```

الاجراء mm

لو شغلنا البرنامج فسوف نجد الاتي



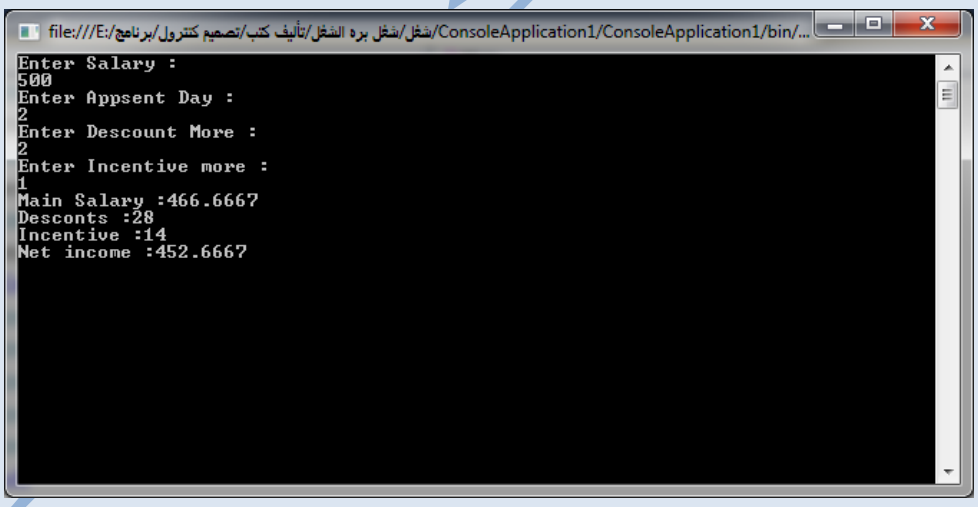
لكن تلاحظ ان الاجراء mm لم يستدعى بسبب اننا لم نستدعيه لاسمه ولم نجهزه ليكن مستعد عند تنشيط الحدث مع العلم اننا اثناء تشغيل البرنامج نعطي للبرنامج معطيات تجعل صافي الدخل اقل من 10% من المرتب الاساسي

لذا يجب علينا ان نجهز الاجراء ليكون جاهز في حالة تنشيط الحدث فيكون جاهز فقط قم بكتابة كلمة Handle ثم اسم الحدث بعد الاجراء هكذا

يكون جاهز عند تنشيط الحدث

```
Sub mm() Handles dd.LawSalary
    Console.BackgroundColor = ConsoleColor.White
    Console.ForegroundColor = ConsoleColor.Black
    Console.WriteLine("Low Salary")
End Sub
```

وبذلك نكون قد استدعينا الحدث عن طريق المتغير المعلن عنه الذي يعلن بتنشيط الحدث معه وعند تشغيل البرنامج ستلاحظ الاتي



في هذه الحالة ستجد نفسك غضبت ***كثير لماذا لم ينشط الحدث ؟؟؟*** و السبب لانك استدعيته عند تنشيط الكائن dd الذي من نوع SalaryCls و الذي ينشء الكائن مع الحدث وليس الكائن Salar الموجود داخل الاجراء main و الذي تم ادخال المعطيات التي نريد ان نحسب من خلالها صافي المرتب فتم تفعي الحدث في الكائن Salar ولم يفعل في الكائن dd الذي ربطنا من خلاله الاجراء mm وجعلنا مجهز ليتم استدعاء عندما نشط الحدث من الكائن dd وليس Salar !!!!! **ركز معي ..... انظر للكود الذي كتبناه**

```
Module Module1
    Public WithEvents dd As New SalaryCls()

    Sub Main()
        Dim sal As Single
        Dim apps, DesC, inscC As Integer
        Console.WriteLine("Enter Salary :")
        sal = CSng(Console.ReadLine)
        Console.WriteLine("Enter Appsent Day :")
        apps = CInt(Console.ReadLine)
        Console.WriteLine("Enter Descount More :")
        DesC = CInt(Console.ReadLine)
        Console.WriteLine("Enter Incentive more :")
        inscC = CInt(Console.ReadLine)
        Dim salar As New SalaryCls(sal, apps, DesC, inscC)
        Console.WriteLine("Main Salary :" + salar.MySalary.ToString)
        Console.WriteLine("Desconts :" + salar.Descounts.ToString)
        Console.WriteLine("Incentive :" + salar.incentives.ToString)
        Console.WriteLine("Net income :" + salar.Netincome.ToString)
        Console.ReadLine()
    End Sub

    Sub mm() Handles dd.LawSalary
        Console.BackgroundColor = ConsoleColor.White
        Console.ForegroundColor = ConsoleColor.Black
        Console.WriteLine("Low Salary")
    End Sub
```

هناك متغيران من نوع SalaryCls واحد كائن مفعل خارج الاجراء main ويستدعي الاحداث وهو المتغير dd الذي اعطناه الرقم 1

وهناك متغير اخر داخل الاجراء قمنا باعطاء قيم للخصائص واستخدماه في حساب المرتب ولكن المشكلة هنا انك تفعل الحدث الذي يستدعي الاجراء من كائن وتطلب وتدخل البيانات و الناتج من كائن اخر لذا يجب عليك ان تفعل الحدث lawSalary من دخال المتغير Salar المرمز اليه بالرقم 2 في هذه الحالة يجب ان نقوم بتعديل الكود كاملا حتى يمكن استداعاء الاجراء من خلال الحدث المنشط من المتغير Salar فسوف يتغير الكود ليصبح كلاتي

```
Sub Main()
    Dim sal As Single
    Dim apps, DesC, inscC As Integer
    Console.WriteLine("Enter Salary :")
    sal = CSng(Console.ReadLine)
    Console.WriteLine("Enter Appsent Day :")
    apps = CInt(Console.ReadLine)
    Console.WriteLine("Enter Descount More :")
    DesC = CInt(Console.ReadLine)
    Console.WriteLine("Enter Incentive more :")
    inscC = CInt(Console.ReadLine)
    Dim salar As New SalaryCls(sal, apps, DesC, inscC)
    dd = salar
    Console.WriteLine("Main Salary :" + dd.MySalary.ToString)
    Console.WriteLine("Desconts :" + dd.Descounts.ToString)
    Console.WriteLine("Incentive :" + dd.incentives.ToString)
    Console.WriteLine("Net income :" + dd.Netincome.ToString)
    Console.ReadLine()
End Sub
```

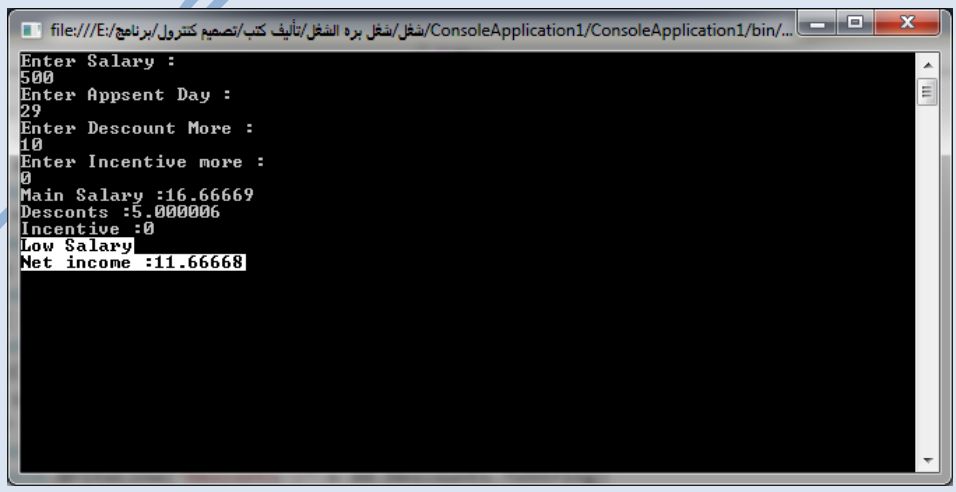
هنا عدلنا الكود

تلاحظ عند تعديل الكود اننا قمنا بالاتي

في السطر الاول من الكود المعدل قمنا باعطاء كل قيم المتغير Salar الى المتغير dd

ثم استدعينا الناتج من المتغير dd يعني بدلنا فبدل من اننا نستدعي الناتج النهائي وصافي الدخل من المتغير dd المرمز اليه بالرمز 1 في الشكل السابق بدلا من التغيرsalar المرمز اليه بالرمز 2

اذا شغلنا البرنامج ادخلن معطيات تعطي ان صافي الدخل اقل من 10 % من الدخل التعاقد عليه بحيث اننا نجعل ايام الغياب 29 يوم و عدد المخالفات تساوي 10 ستلاحظ الاتي



تم تفعيل الحدث وتم استدعاء الاجراء main

الطريقة الثانية في الاعلان عن الحدث

هناك طريقة اخرى لاستدعاء لاسدعاء الاجرء عند تنشيط الحدث ولكننا يجب ان تتعلم الطريقة الاولى لانها شائعة ، الطريقة الثانية هي نفس الخطوتين الاولى و الثانية في الطريقة الاولى ولكن في الخطوة الثالثة فاننا نستدعي دون الاعلان عن متغير يحمل الحدث يعني بدون ان نعلن عن المتغير dd سنعود لتعديل الكود ليكون كالكود الذي كتبناه اول مرة دون استدعاء أي حدث ونضيف اليه الاجراء mm ليكون كلاتي

```
Module Module1


    Sub Main()
        Dim sal As Single
        Dim apps, DesC, inscC As Integer
        Console.WriteLine("Enter Salary :")
        sal = CSng(Console.ReadLine)
        Console.WriteLine("Enter Appsent Day :")
        apps = CInt(Console.ReadLine)
        Console.WriteLine("Enter Descount More :")
        DesC = CInt(Console.ReadLine)
        Console.WriteLine("Enter Incentive more :")
        inscC = CInt(Console.ReadLine)
        Dim salar As New SalaryCls(sal, apps, DesC, inscC)

        Console.WriteLine("Main Salary :" + salar.MySalary.ToString)
        Console.WriteLine("Desconts :" + salar.Descounts.ToString)
        Console.WriteLine("Incentive :" + salar.incentives.ToString)
        Console.WriteLine("Net income :" + salar.Netincome.ToString)
        Console.ReadLine()
    End Sub
```

```
    Sub mm()
        Console.BackgroundColor = ConsoleColor.White
        Console.ForegroundColor = ConsoleColor.Black
        Console.WriteLine("Low Salary")
    End Sub
End Module
```

هكذا اصبح الكود ك الكود الاول نعود ونستدعي الحدث من المتغير salar فعيلك اتباع القاعدة الاتية

```
Add Handler Event Name , Addressof Function or Prosedure
```

Event Name تعني اسم الحدث الذي نريد ان يكون نشط

Function or Prosedure الدوال و الاجراءات التي نريد ان نستدعيها

دعنا نجرب هذا في الكود الذي كتبناه فنضع الكود الاتي في الحدث main

```
Dim salar As New SalaryCls(sal, apps, DesC, inscC)

Console.WriteLine("Main Salary :" + salar.MySalary.T
Console.WriteLine("Desconts :" + salar.Descounts.ToS
Console.WriteLine("Incentive :" + salar.incentives.T
AddHandler salar.LawSalary, AddressOf mm
Console.WriteLine("Net income :" + salar.Netincome.T
Console.ReadLine()
```

تلاحظ اننا قمنا باستدعاء الاجؤاء mm بحيث اذا تم تنشيط الحدث فانه يتم تفعيل الحدث

واريد ان انوه الى شيء مهم جدا وهو انك تكتب اسم الاجراء بعد كلمة Addressof بدون كتابة أي معامل اكرر بدون كتابة أي معاااااامل **وسوف اشرح كيف تتعامل المعالملات مع الاحداث**

شغل الان البرنامج وكرر البيانات التي ادخلتها ليكون صافي المرتب اقل من 10% وستجد ان الحجث ينشط ويستدعي الحدث

الان نقوم بانشاء الحدث ولكن مع وجود معاملات

سوف نعدل الكود بحيث اننا نستعمل المعاملات في الاحداث عدل الكود الخاص بالتصنيف SalaryCls بحيث نضيف اليه معامل نصي وهذا الكود في الخطوة الاولى

```
Public Event LawSalary(ByVal s As String)
```

في هذه لحالة سوف نستدعي الاجراء الذي يكون لديه معامل واحد فقط ويكون المعامل نصي وعند الانتقال للخطوة الثانية وهي اننا نقوم بعمل تنشيط للحدث سوف ندخل قيمة المعامل في الكود الاتي

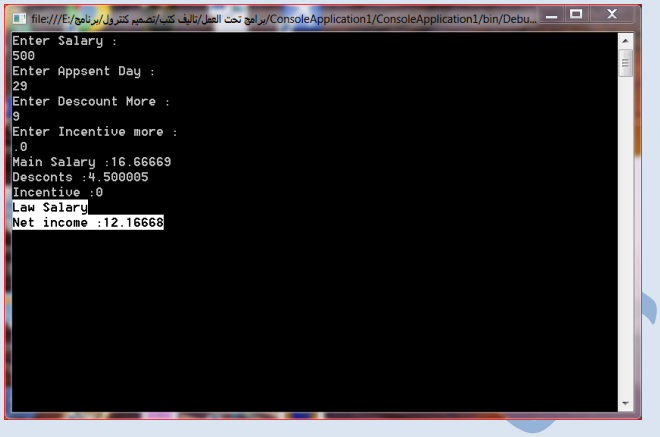
```
RaiseEvent LawSalary("law string")
```

و في الخطوة الثالثة كما كنا نطبق قاعدة التنشيط وعند استدعاء الاجراء سوف نعدل الاجراء mm الى ان يكون كلاتي

```
Sub mm(ByVal s As String)
    Console.BackgroundColor = ConsoleColor.White
    Console.ForegroundColor = ConsoleColor.Black
    Console.WriteLine(s)
End Sub
```

هنا عدلنا الاجراء ليكون به معامل نصي مثل الذي في الحدث ثم في السطر الاخير اعيطنا قيمة طلبنا من الكنسل اظهار قيمة المعامل التي طبعا ستكون قيمة الحدث عند تنشيطه و استدعاءه للاجراء جرب الان باستدعاء الاجراء بطريقة AddHandler

وستجد المعامل النصي في الحدث يظهر كما طلمبنا منه



كما يمكنك استخدام الطريقة الثاينة في استداعاء الحدث ستجد نفس النتيجة و احب ان وضح انه من الممكن عمل حدث مشترك

SHARE EVENT

في هذه الفقرة سوف ننشء حدث مشترك مثل الدوال و الاجراءا وسوف يكون الحدث موجود في التصنيف SalarCls وليكن الحدث OverDescount ويتم تفعيله عندما يكون عدد المخالفات اكبر من 10 فتعالى نقوم بتعديل صغير في التصنيف لكون في الشكل الاتي

سنعلن عن حدث مشترك في التصنيف

```
Public Shared Event OverDescount()
```

ثم نعدل الدالة Descount التي تحسب الخصومات الى الشكل الاتي

```
Private Function Descount(ByVal DescountCount As Integer, ByVal Salary As Single) As Single
    Const Percentage As Decimal = 0.03 ' نسبة الخصومات
    ' هو حساب الخصومات عن طريق ضرب الراتب الذي تم حسابة بعد خصم ايام الغياب
    'في نسبة الخصم ثم ضربه في عدد مرات المخالفة
    If DescountCount > 10 Then
        RaiseEvent OverDescount()
    End If
    Return (Salary * Percentage) * DescountCount
End Function
```

ادخلنا الشرط بحيث اذا كان عدد المخالفات اكبر من ١٠ يفعل الحدث

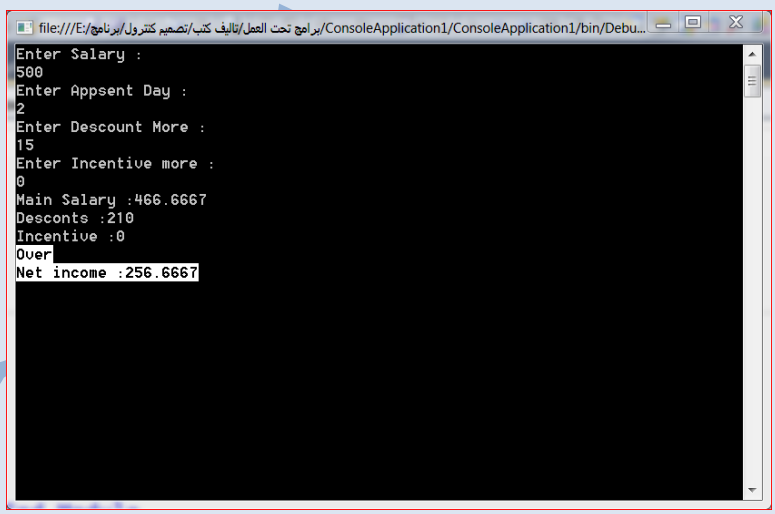
ثم في قسم الكنسل سوف نضيف اجراء جديد وليكن over يقوم بكتابة انذار عندما ينشط الحدث ويتم استدعاءه

```
Sub Over()
    Console.BackgroundColor = ConsoleColor.White
    Console.ForegroundColor = ConsoleColor.Black
    Console.WriteLine("Over")
End Sub
```

ونكتب في الدالة main الكود من خلاله يتم استدعاء الحدث عند تنشيط الحدث ولكن هذه المرة يجب استدعاءه من خلال اسم التصنيف وليس عن طريق متغير لانه حدث مشترك

```
Console.WriteLine("Main Salary :" + salar.MySalary.ToString)
Console.WriteLine("Desconts :" + salar.Descounts.ToString)
Console.WriteLine("Incentive :" + salar.incentives.ToString)
AddHandler salar.LawSalary, AddressOf mm
AddHandler SalaryCls.OverDescount, AddressOf Over
Console.WriteLine("Net income :" + salar.Netincome.ToString)

Console.ReadLine()
```

واذا شغلنا البرنامج واخلنا عدد المخالفات يساوي 15 فيظهر الشكل الاتي



بهذه نكون قد انتهينا من شرح الاحداث يقي لنا في هذا الفصل التفويضات وهي اسهل بكثير من الاحداث

التفويضات DELEGATES

التفويضات مثل الاحداث تقوم باستدعاء الاجراء او الدالة من داخل اجراء لتنفيذ الامر ولكن بدون عمل شرارة او حدث بل تكون استداء مثل اي استدعاء لدالة او اجراء فهي يمكنها استدعاء عدة دوال داخل اجراء واحد بدون ان نعيد تكرار الاجراء .....

مثال لو عندنا دالة تحسب عمولة الموظفين من مندوبي المبيعات و المحاسبين و عمال الانتاج فنلاحظ اننا سنجد ان نصف المعادلة التي تحسب العمولة متشابهه عند كل انواع الموظفين ولكن تجد انها تختلف في النصف الاخر فنضطر

اننا ننشئ دالة خاصة لحساب عمولة المحاسبين ودالة اخرى لحساب عمولة المندوبين ودالة ثالثة لحساب عمولة عمال الانتاج فنجد اننا ننشئ 3 دوال متشابهة في معظم اكوادها ومختلفة في جزء صغير فيها فلماذا لا يكون هناك دالة واحدة بها الجزء المشترك ثم نستدعي باقي المعادلة على شكل تفويضات

ولكي ننشئ التفويضات فاننا نتبع الخطوات الاتية

1- **الاعلان :** نعلن عن التفويضات باتباع القاعدة الاتية

```
[private - public] delegate function Or sub delegate Name (argument) As type
```

[private - public] : تحدد مدى الوصول الى تفويضة

function Or sub هنا نكتب احدى الجملتين اما تكون Function او Sub لتحديد نوع الاسلوب او method الذي نختاره

delegate Name اسم التفويض

argument المعامل المستخدم فهو يشبه المعامل في الحدث عند استدعاء الدالة يتم ادخال القيم من خلال المعامل مع العلم ان اي متغير من نوع تفويض لدالة رقمية و يحمل معاملين لا يستدعي الا دالة رقمية وتحمل معاملين من نفس النوع الخاص بالمعاملين

type نوع الدالة فهي لا تستخدم في الاجراء

2- **استدعاء المفوض:**

عندما نريد استدعاء

```
[ private -public] Variable Name as new Delegat Name(addressof Fumction name)
```

private -public] تحدد الوصول

Variable Name : اسم المتغير لكي يتم انشاء تفويضة

Delegat Name اسم التفويضة التي نريد استدعائها

addressof *Fumction name* كلمة Address of تستدعي الدالة او الاجراء مثل الاحدث وبدون ادخال اي معامل

3- **استدعاء الدالة**

في المرحلة السابقة حددنا الدالة التي نريد استدعاءها ولكن نريد تحديد متى يتم استدعاءها فتخيل عند اجراء تستخدم فيه عده اوامر وتريد ان تضع هذه التفويضة في وسط الكود بحيث انك تغير الدوال من وقت لاخر لذا فاننا عندما نريد ان نستدعي الدالة فاننا نتبع الخطوة الاتية

```
Name Variable as Delegate . invoke(Value)
```

Name Variable as Delegate . هنا يكون اسم المتغير الذي يحمل التفويضة

Invoke اجراء من خلاله يتم استدعاء الدالة او الاجراء الذي تم تحديده في المرحلة السابقة

Value قيمة المعامل

مثال

افتح مشروع كنسل ثم اكتب الكود الاتي وستجد نفسك تفهمه بسهولة

```
Private Delegate Function dlag() As Integer

Public Function mm() As Integer
    Return 546
End Function
Sub Main()

    Dim mms As New dlag(AddressOf mm)

    Console.WriteLine(mms.Invoke.ToString)

    Console.ReadLine()
```

١ ٢ ٣

في الكود السابق سوف نشرح كل رقم رقمته في الكود

1- هنا نعلن تفويضة نحدد من خلالها شكل الدالة او الاجراء وما هو نوعهما و ما شكل المعاملات
2- هنا الدالة التي سوف نستدعيها
3- متغير من نوع التفويضة التي اعلنا عنها في الرقم واحد ومن خلاله نحدد الدالة التي نريد استدعائها وكما يجب ان نوضح ان الدالة يجب ان تكون نفس النوع ونفس عدد المعاملات كما شرحنها في الاحداث
4- هنا نحدد مكان استدعاءا لدالة من خلال الاجراء invoke وندخل القيم التي تخص المعاملات ولكن في المثال كنا لم نعلن عن معاملات

قم بتشغيل البرنامج ولاحظ انه تم استدعاء الدالة عن طريق التفويضة

مثال 2

افتح كنسل جديد واكتب فيه هذه الكود

```
Module Module1

    Private Delegate Function dlag(i As Integer) As Integer

    Public Function DalgFuntio(i As Integer) As Integer
        Return i
    End Function

    Private Function Goo(delf As dlag) As String
        Return delf.Invoke(564)
    End Function

    Sub Main()
        Console.WriteLine(" value is " + Goo(AddressOf DalgFuntio))

        Console.ReadLine()
    End Sub


End Module
```

1 2 3 4

نشرح على اساس الارقم مرة اخرى

1- نفس المثال السابق اعلنا عن تفويضة وجعلنا لها دالة رقمية و لها معامل رقمي
2- الدالة التي نريد استدعائها ويجب ان تكون من نفس النوع ونفس المعامل
3- هي دالة نصية بها معامل من نوع التفويضة Dlag التي اعلنا عنها ...فنستخدم هذا المعامل في تفعيل الدالة فهو به عدة دوال سنستخدم منها الدالة invock التي من خلالها نفعل الدالة ولكننا لم نستدعي الدالة بعد
4- هنا نقوم باستدعاء الدالة الرقمية delfunc ونفعلها مرة واحدة عن طريق المعامل في الدالة النصية Goo حيث اننا نستدعي الدالة النصية ثم نمرر قيمة للمعامل الذي هو من نوع تفويضة dlag ولكي نمرر القيمة لاي معامل من نوع تفويضة فيكون عن طريق استخدام الدالة Addressof التي نستخدمها في اسدعاء الدالة ويجب ان نعرف انه نستدعي اسم الدالة **ولا نمرر عليها اي معامل**

وبهذه الطريقة عرفنا كيف تقوم استخدام المعاملات من نوع التفويضات في تمرير الدوال التي نريد استدعائها

انصحك ان تتدرب عليها بيدك مرة واخرى حتى تفهمها جيدا

بعد ان انتهينا من شرح التفويضات سننتقل الان الى مفاهيم برمجة الكائنات الموجهه وقبل ان ننتقل يجب عليك ان تتدرب مرة اخرة على كل الامثلة في هذا الفصل و الفصل السابق حتى تتاكد من فهمك منها ثم انتقل معي للفصل التالي نهي الباب الاول وننتهي من المبادئ العامة في البرمجة .

الفصل الخامس

برمجة الكائنات الموجهه

هذا الفصل هو اخر فصل في الباب الاول وبه ننهي المفاهيم التي يجب عليك كمبرمج محترف معرفتها في تصميم ادوات التحكم و حتى بناء مكتبات الDll و التي يمكن ان توفر لك وقت طويل في البرمجة وتجعل منك مبرمج محترف تستطيع ان تدخل سوق العمل بكل قوة

برمجة الكائنات الموجهةOOP

كنا شرحنا ما هي الكائنات في الفصل السابق و الان سوف نتناول اهم المزايا التي يتميز بها اسلوب برمجة الكائنات الموجهة ، فارجو من ان تركز جيدا وتقرا الفصل كاملا حتى لا يفوتك شيء وارجوا انك لم تمسح المثال السابق الخاص بالمرتاب لاننا سوف نبني عليه باقي المفاهيم في هذا الفصل

تتكون مبادئ برمجة الكائنات الموجهة من

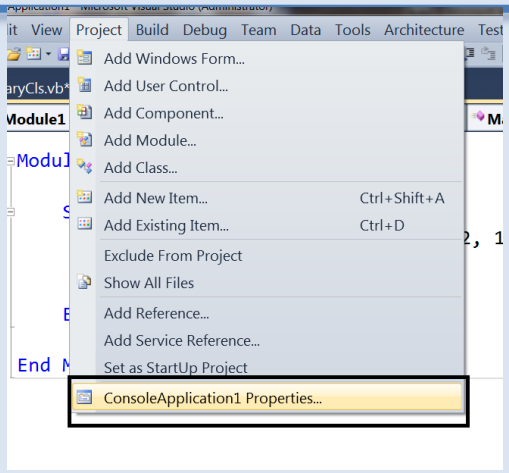
- الكبسلة Encapsulation
- الوراثة inherent
- تعدد الاوجه Polymorphism

1 – الكبسلة Encapsulation

لتوضيح معنى الكبسلة تخيل انك قمت بالاطلاع على احدى النظريات الفيزياءية لحل مشكلة ما ثم قمت تعمل برنامج ليطبق هذه النظرية واردت ان تبيعها لاحدى الشركات البرمجة لتطبيق هذا الكود في احدا المشاريع العملاقة فانك لا تريد ان تتطلع الشركة على سر مهنتك فتقوم باعطائها الكود مضغوط بحيث لا يمكن لاحد الاطلاع عليه ولا يتم التعديل في الكود المصدري بل فقط بتم انشاء الكائنات التي تريد ان ينشؤوها ويقوموا بتعديل قي خصائصها فقط

ولكي تعرف كيف يتم عمل ضغط وكبسلة للتصنيف الذي انشاناه بحيث نستطيع استدعائه في اي مشروع اخر اتبع الخطوات الاتية

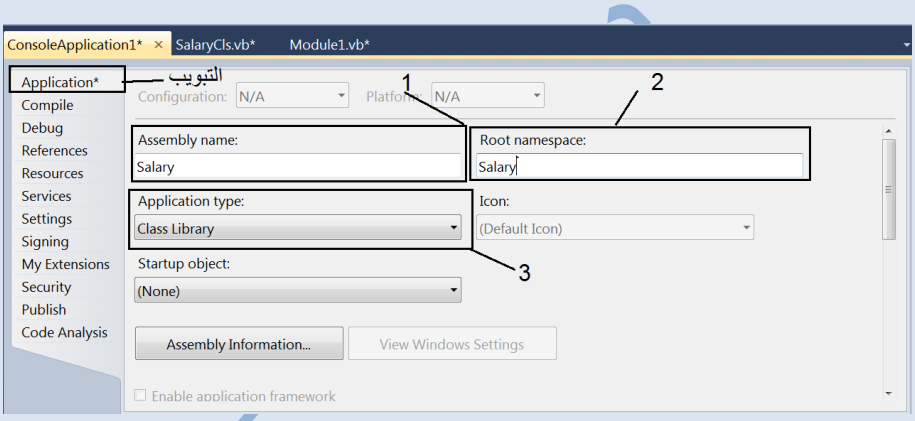
1- افتح قائمة property → project كما في الشكل الاتي وفيه سوف نعد خاصية المشروع من Consle الى Class liberary



2- ثم نقوم نعدل الخصائص الاتية داخل التبويب Application

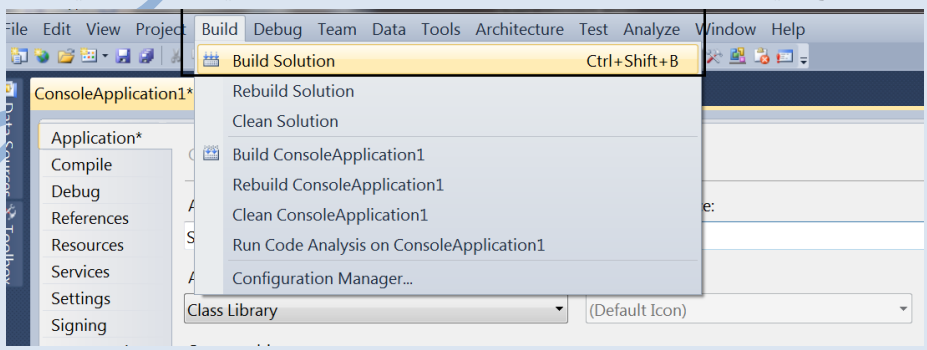
- تعديل اسم Assembly name الى Salary
- تعديل اسم Root Namespace الى Salary
- تعديل Application type الى Class library

كما في الشكل الاتي



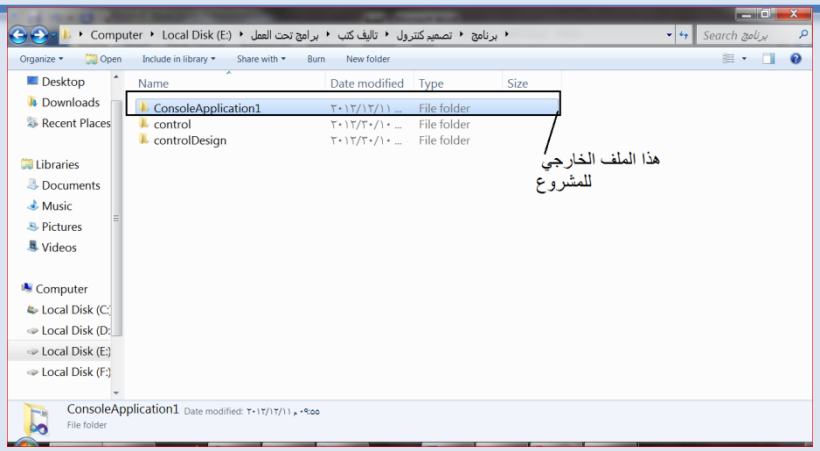
الارقام التي رقمتها هي نفس ترتيب الخطوات التي شرحتها قبل الصورة

3- ثم قوم بعمل بناء للمشروع في قائمة Build→ Build solution كما كنا في الشكل الاتي

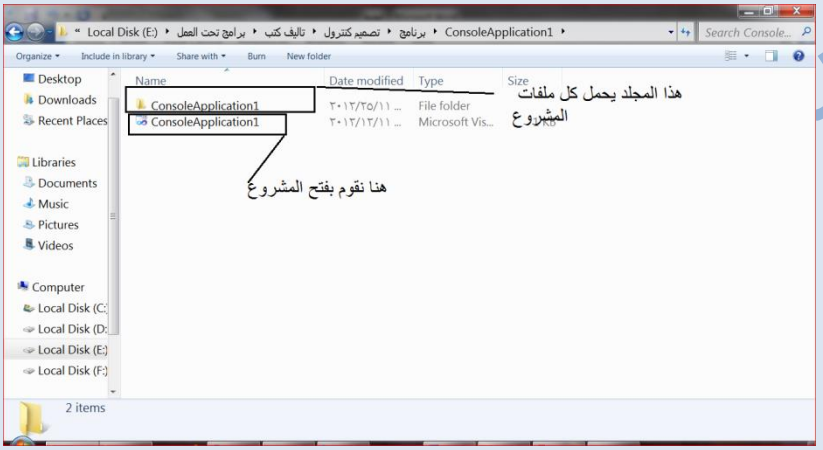


ثم انتقل الى الملف الاتي خارج برنامج فيجوال استوديو بحيث ان هذا المسار من داخل ملف الشروع هذا المسار هو-debug→ Bin

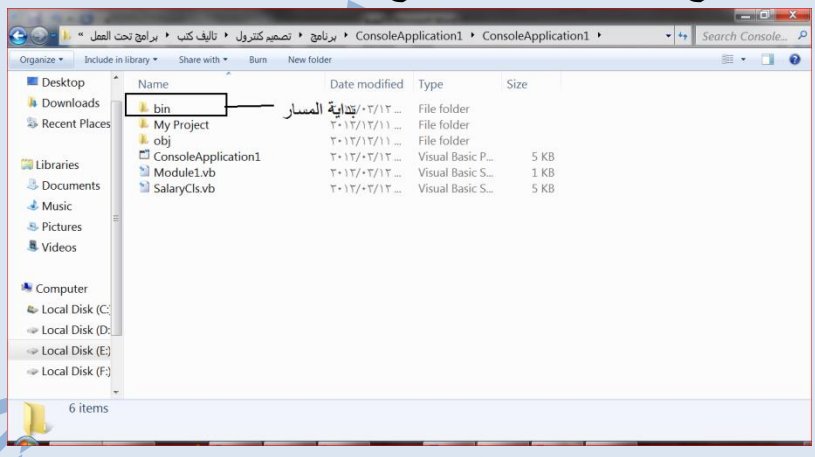
ولفهم ذلك تابع الشكل الاتي



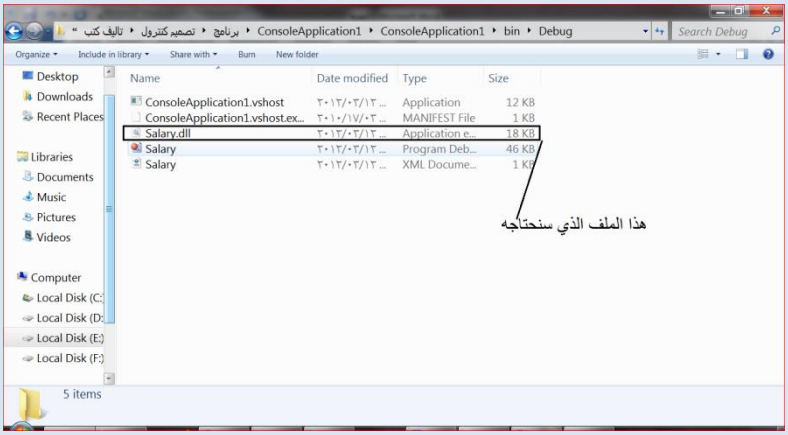
الشكل السابق هو الملف الخارجي للمشروع



في هذا الشكل مكونات المشروع ادخل الى مجلد المشروع

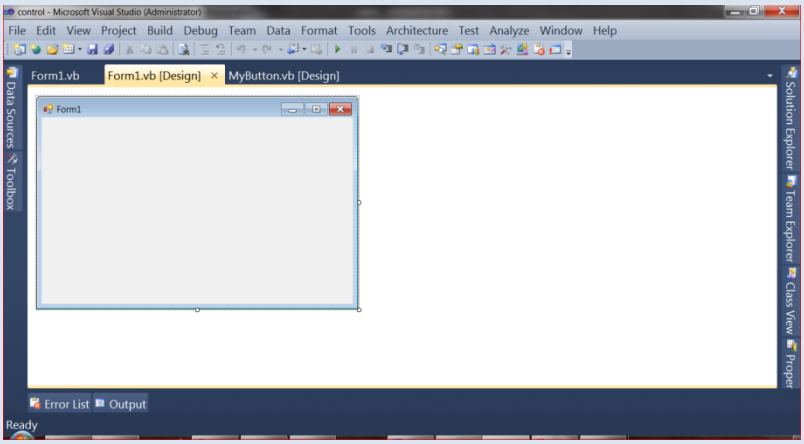


ولو اتباعن المسار الذي اشرنا اليه سوف نجد داخل مجلد Debug الشكلا الاتي

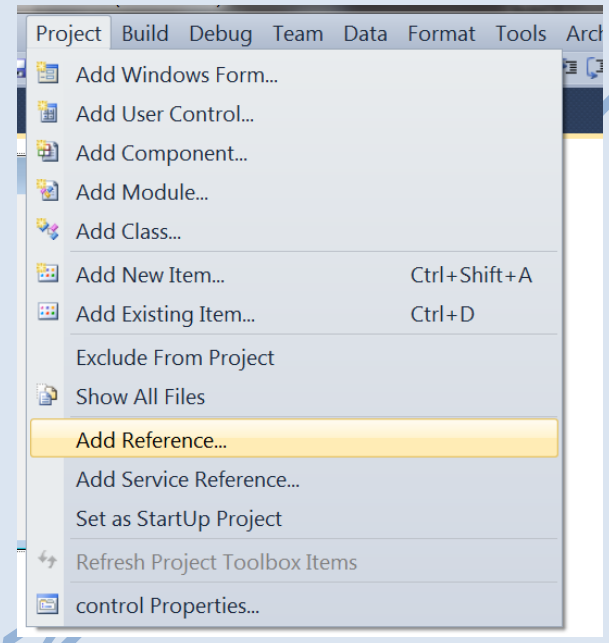


هذا هو الملف الذي تم فيه ضغط التصنيفات التي نريد استخدامها في اي مشروع خارجي

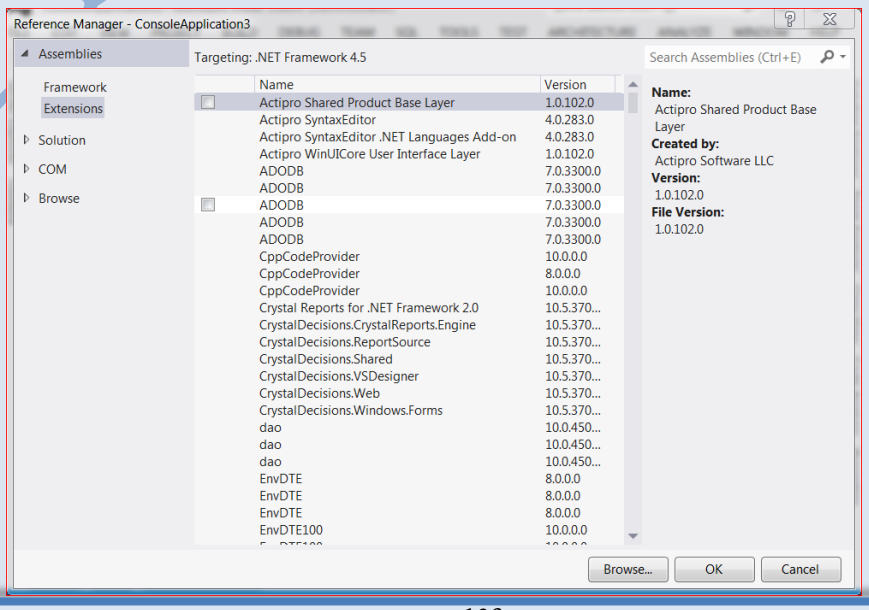
4- نقوم بفتح اي مشروع اخر وليكن من نوع تطبيقات Windows فيكون الشكل كلاتي



5- من قائمة project نختار Add reference



فتظهر لنا الشاشة الاتية



ستجد امامك عدة تبويبات

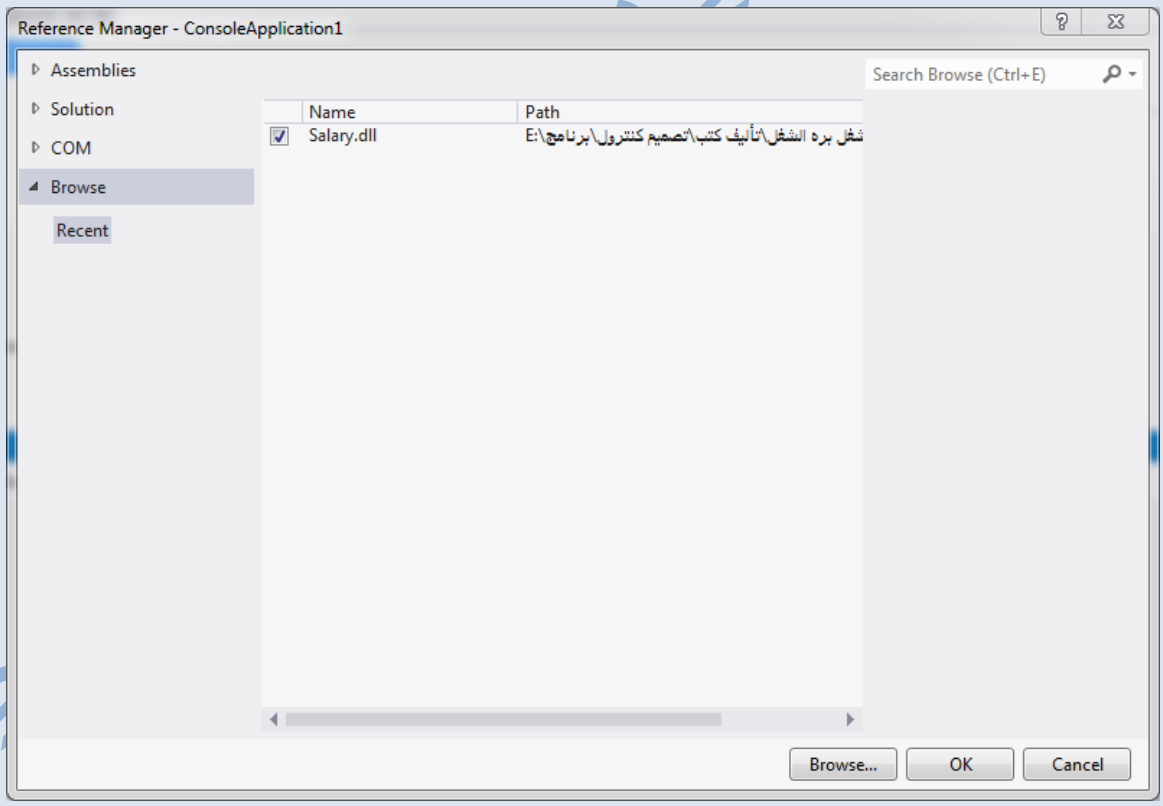
1— تبويب **Assemblies** يتكلم عن المكتبات التي تم تطويرها عن طريق لغات الدوت نت كC# او الفيجوال بيسك وغيرها والتي قامت شركة مايكروسوفت بتطويرها والتي لا يمكن فتحها الا من برنامج فيجوال استوديو ويحتوي على تبويبين هما FramWark بحيث انها المكتبة التي نتطور منها تطبيقاتنا Extensions هي المكتبات التي تم تطويرها من خلال الFramwark

2— تبويب Com يتكلم عن المكتبات التي تم تطويرها عن طريق لغات اخرى غير الدوت نت و التي شرحناها في اول البرمجة كال++C و فيجوال بيسك 6

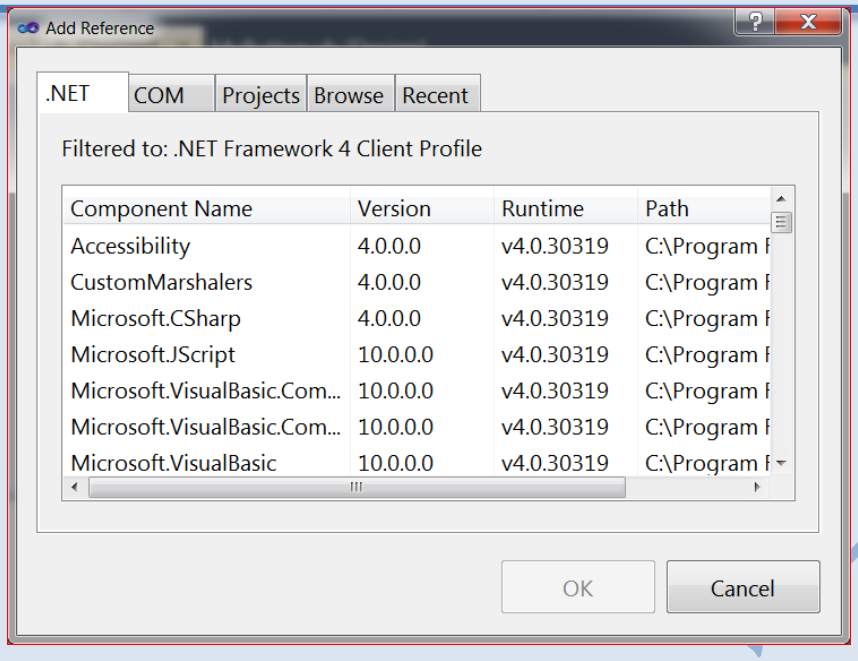
3—نبويب Solution استدعاء مشاريع خارجية

4— Browse هو المكتبات التي قمنا تطويرها وقمنا باستدعائها من بانفسنا كما سنشرح في السطور القادمة

فتقوم باستدعاء المكتبة عن طريق زر Browse ثم اتبع المسار الذي اشرنا اليه وعند اختيار المكتبة ستظهر لك الشاشة كلاتي



هذه النافذة موجودة فقط في فيجوال ستديو 2012 والى مستخدمي الاصدارات السابقة اليكم هذه الشاشة



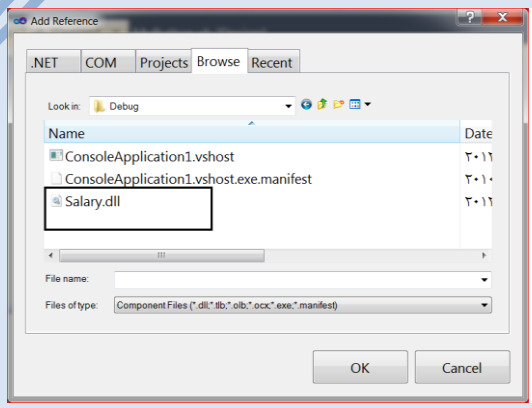
ستجد امامك عدة تبويبات

1— تبويب Net يتكلم عن المكتبات التي تم تطويرها عن طريق لغات الدوت نت كC# او الفيجوال بيسك وغيرها والتي قامت شركة مايكروسوفت بتطويرها والتي لا يمكن فتحها الا من برنامج فيجوال استوديو

2— تبويب Com يتكلم عن المكتبات التي تم تطويرها عن طريق لغات اخرى غير الدوت نت و التي شرحناها في اول البرمجة

3—نبويب project استدعاء مشاريع خارجية

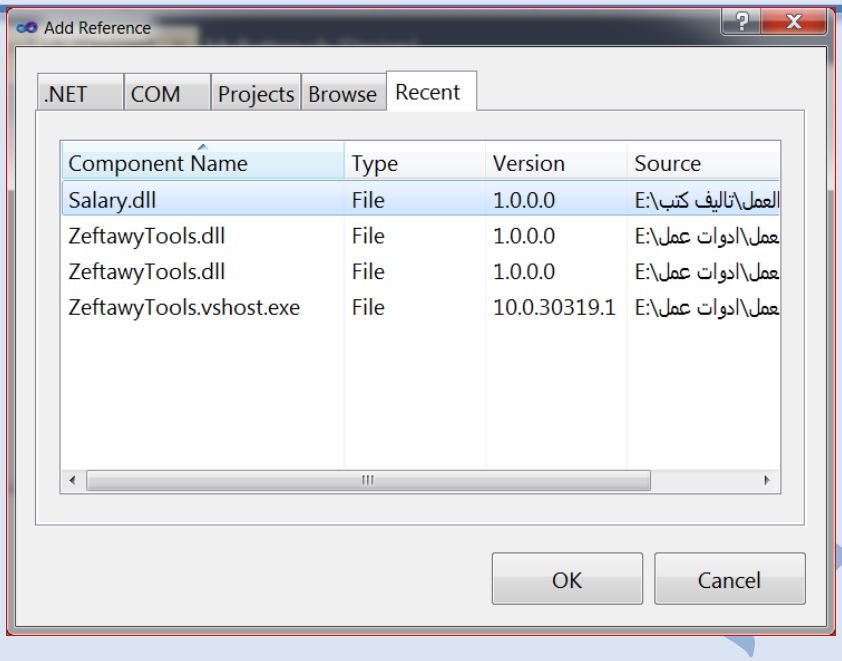
4— Browse هنا سيكون شرحنا بحيث انه سوف ننتقل الى المسار الذي اشرنا اليه ونختار الملف الذي تكلمنا عليه



هنا نختار المكتبة

5-- التبويب Recent يعرض المكتاب الخارجية التي انشأناها وادخلناها للمشروع

فاذا اخترنا المكتبة وضغطنا على ok ثم فتحنا هذه النافذة مرة اخرى ثم نقوم بفتح التبويب ستلاحظ المكتة تظهر في الشكل الاتي



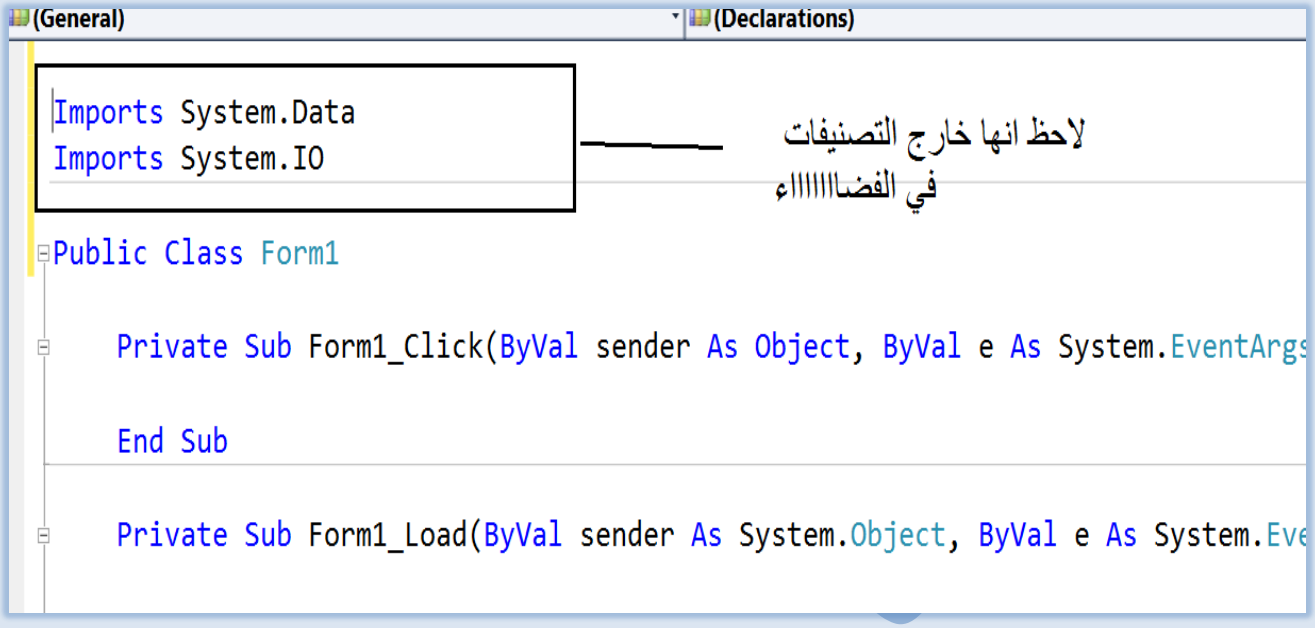
هذه الشاشة لمستخدمة اصدار 2010 و 2008

فضاءات الاسماء

فضاء الاسماء هو المشروع الكبير الذي يضم كل التصنيفات و يضم عناصر اخرى من حاويات الاكواد كالواجهات و الهياكل التي سوف نتناولها في اخر هذا الفصل ، ويعتبر فضاء الاسماء هو اسم المشروع ويمكن لفضاء الاسماء يحمل فضاء اسماء داخله كما هو مستخدم في فضاء الاسماء System المطور من مكتبة net framework التي تتحكم بالفيجوال ستوديو ولا يمكن ان نتمكن من انشاء تطبيقات بدون هذا الفضاء System واليك نبذة عن فضاء الاسماء System

فضاء الاسماء	الاستخدام
system	هو فضاء اسماء عام يحمل كل الفضاءات اسماءالاخرى ويحمل التصنيفات التي تكون انواع البيانات (integer, string) كما يتعامل مع الاستثناءات وغيرها
System.Data	هذا الفضاء يتعامل مع قواعد البيانات ويربط قواعد البيانات بالتطبيقات التي نطورها
System.IO	هذا الفضاء يتعامل مع الملفات بفتح ملفات وينسخ ويمسح الملفات
System.Draw	هذا الفضاء يقوم مسئول عن الرسم و التعامل مع الصور
System.Math	هذا الفضاء يحتوي على المعادلات الرياضية كالدوال المثلثية و حساب الوسط الحسابي

وغيرها من هذه الفضاءات التي تدخل بشكل مباشر في تطوير التطبيقات ولكي نستطيع استخدامها يجب علينا انقوم باستيراده ونستوردها في اعلى شاشة الكود بكلمة imports كما هو موضح في الشكل الاتي



بهذه الطريقة سوف نستورد المكتبة التي انشاناها والتي اسمها Salary

```
Imports System.Data
Imports System.IO

Imports Salary

Public Class Form1

    Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.Event
```

والان قم بعمل كائن وادخل عليه المدخلات سوف تجد انك لا تستطيع رؤية الخصائص او الدوال الخاصة و التي تبدا بكلمة private وانما العامة فقط وهذه هي الكبسلة يعني انك لا تستطيع رؤية كواليس البرنامج فقط تتعامل مع الكائنات وتعدل في خصائصها وتخرج الناتج

2 – الوراثة inherent

من اهم المبادئ التي تيسر عليك كتابة الاكواد وتقلل الاخطاء وتسهل الوصول الى مواضع الخطأ او التعديل فيه بسهولة .. يعتمد مبدى الوراثة على ان الابن يرث كل الصفات و الخصائص من والده وتسمى ايضا بالاشتقاق لان التصنيف الابن ياخد كل خصائص الاب ثم يستطيع التعديل عليها او اضافة خضائص اخرى

تخيل انك قمت بعمل التصنيف الخاص بالمرتبات SalaryCls ولديك 10 تصنيفات اخرى بها نفس الكود ولا تريد ان تقوم بانشاء كائن داخل التصنيفات فانك تضطر الى استخدام اسلوب النسخ و اللصق ... اذا فكرت بهذه الطريقة فاني اقول لك انك تفكر باسلوب خاطئ تماما لانك بهذه الطريقة لو حدث اي خطاء في التصنيف الاب وهو Salarycls فمثلا تريد تعديل الدالة التي تحيسب قيمة الخصومات فانه يتعين عليك تعديل الدالة في كل التصنيفات العشرة الاخرى ..... هنا ياتي دور التوريث حيث انك تعطي امر التوريث للتصنيفات العشرة فتجد ان كل التصنيفات العشرة بها نفس الخصائص و الاساليب التي في التصنيف الاب واذا عدلت الكود في التصنيف الاب ستجد انه يعدل في كل التصنيفات التي ترث منه بدون تدخل منك

ولكي تقوم بعمل وراثة لاي تصنيف فاتبع الخطوات الاتية

- افتح مشروع جديد وقم باستيراد المكتبة Salary.dll باستخدم كلمة import ثم قم بانشاء تصنيف Saler أو يمكن من نفس المشروع الذي فيه التصنيف SalarlyCls تستطيع عمل عملية التوريث
- قم بعمل وراثة للتصنيف SalaryCls كما في الكود الاتي

```
Imports Salary
Public Class Saler
    Inherits SalaryCls

End Class
```

هذا هو كود الوراثة

وعند الاتسيراد ستلاحظ اننا اتبعنا القاعدة التية

```
Inherits Class Name
```

حيث ان Class Name هي اسم التصنيف الذي نريد الوراثة منه

- ملاحظة هاااااام لا يمكن بان يتم الوراثة من تصنيفين واحد فقط ولكن يمكن ان عدة تصنيفات تورث من تصنيف واحد يعني انك لا تستطيع ان تكتب كود الوراثة مرتين
- ولكي نتاكد التصنيف الابن Salaer قد ورث من التصنيف الاب قم بعمل كائن من نوع Saler وستجد انه قد اخذ كل الخصائص و التصنيفات ولكن طبعا انك لن تستطيع رؤية الدوال الخاصة private

محددات الوصول

الان سوف نشرح ما هي محددات الوصول : هي عبارة عن كلمات مفتاحية تقوم بتحديد مدى العمر الافتراضي الي عنصر من عناصر فضاء الاسماء من تصنيفات او خصائص او دوال او حتى متغيرات بحيث يحدد مدى قدرة الوصول الى العنصر كما يحدد قدرة العنصر بالحتفاظ بقيمته مثل متغير عام يحفظ بقيمته طول فترة التشغيل على عكس المتغير الخاص الذي يتهي عمره عند الانتهاء من العصر الذي يحتويه

ولكي تستطيع الفهم اكثرسوف نعرف الكلمات المفتاحية ونحدد مدى الوصول اليها

1- Public

هذه الكلمة تعلن عن انه يمكن رؤية العنصر الذي تعلن عنه سواء كان تصنيف او عناصر التصنيف من خواص و دوال ومتغيرات بحيث يمكن رؤيتها في اي مكان سواء

- يمكن رؤيتها من داخل التصنيف نفسه
- يمكن رؤيتها من اي تصنيف اخر اذا قمت بانشاء كائن من داخل فضاء الاسماء او المشروع و خارجه
- يمكن رؤيتها من اي تصنيف مشتق من التصنيف الاساسي سواء داخل المشروع او خارجه

2- Private

هذه الكلمة تمنح العنصر المعلن عنه بها بامكانية رؤيته من داخل التصنيف فقط ولا يسمح برؤيتها من اي مكان اخر

3- Friend

يمكن رؤية العنصر من خلال الاتي

- من داخل التصنيف التي الموجودة فيه
- من اي تصنيف اخر ادا قمت بانشاء كائن ولكن من داخل المشروع او فضاء الاسماء
- من اي تصنيف مشتق ولكن من داخل المشروع او فضاء الاسماء

4- Protection

- يمكنك رؤيتها من داخل التصنيف
- يمكنك رؤيتها من التصنيف المشتق من داخل فضاء الاسماء فقط

لا يمكن رؤيتها من اي مكان اخر

5- Protection Friend

- يمكن رؤيتها من داخل التصنيف
- يمكن رؤيتها من داخل التصنيف المشتق من داخل فضاء الاسماء
- يمكن رؤيتها اذا قمت بانشاء كائن من التصنيف المشتق ولا تستطيع رؤيتها اذا انشات كائن من التصنيف الاب او الاساسي

يعني لو اننا لدينا تصنيفين A ,B و التصنيف B يرث من التصنيف A و العنصر C عنصر صديق محمي Protection Friend فانك اذا انشاء كائن من التصنيف A فانك لا تستطيع رؤية العنصر C بينما لو انشات الكائن من التصنيف B فانك تستطيع رؤية العنصر c

ينمكن تلخيص ما سبق في الجدول الاتي

	رؤيته من كائن من خارج فضاء الاسماء	رؤيته من التصنيف المشتق من خارج فضاء الاسماء	رؤيته من كائن من داخل فضاء الاسماء	رؤيته من التصنيف المشتق من داخل فضاء الاسماء	يمكن رؤيته من داخل التصنيف نفسه	يمكن رؤيته من داخل كائن من التصنيف المشتق من داخل فضاء الاسماء
public	نعم	نعم	نعم	نعم	نعم	نعم

private	لا	لا	لا	نعم	نعم	لا
Friend	لا	لا	نعم	نعم	نعم	نعم
protected	لا	لا	لا	نعم	نعم	لا
Protected Friend	لا	لا	لا	لا	نعم	نعم

ملحوظة : كلمة Dim التي تستخدم لاعلان المتغيرات كانت تستخدم في لغة الفيجوال بيسك 6 لذا تم ادراجها في لغة الفيجوال بيسك دوت نت وتستخدم للاعلان الخاص فقط private ان كان لاعلان خارج الدوال و الاجراءات وداخل التصنيف ولا تستخدم خارج التصنيف وتستخدم للاعلان عن متغير داخل عناصر التصنيف مثل الدوال و الاجراءات ولا يسمح باستخدام كلمة اخرى غير Dim للاعلان داخل العناصر كالدوال والاجراءات

نعود الان لمبدأ التوريث لدراسة بعض انواع التصنيفات من حيث تعاملها مع التوريث

تصنيفات غير قابلة للتوريث

هي التصنيفات التي لا تسمح لاي تصنيف ان يرث منها عناصرها ويتم ذلك باتباع القاعدة الاتية

```
[public – private] notinheritable Class ClassName
```

ستلاحظ ان قاعدة انشاء التصنيف غير قابل للتوريث نفس القاعدة انشاء التصنيف العادي لكن ستجد اننا نضيف الكلمة المفتاحية NotInheritable

فلو كنا نريد ان نجعل التصنيف SalaryCls غير قابل للتوريث فاننا سوف نكتب التصنيف كلاتي

```
Public Notinhertable Class SalaryCls

العناصر التي نريد ادخالها

End Class
```

واذا حاولت ان ترث هذا التصنيف ستجده يرفض التوريث وتظهر علامة خطا وهو خط ازرق كما في الشكل الاتي

```
Inherits SalaryCls
'SaleCls' cannot inherit from class 'SalaryCls' because 'SalaryCls' is declared 'NotInheritable
```

لكن تستطيع انشاء كائن منه باستخدام الدالة new

تصنيفات اجبارية التوريث (التصنيفات المجردة)ABSITRACT

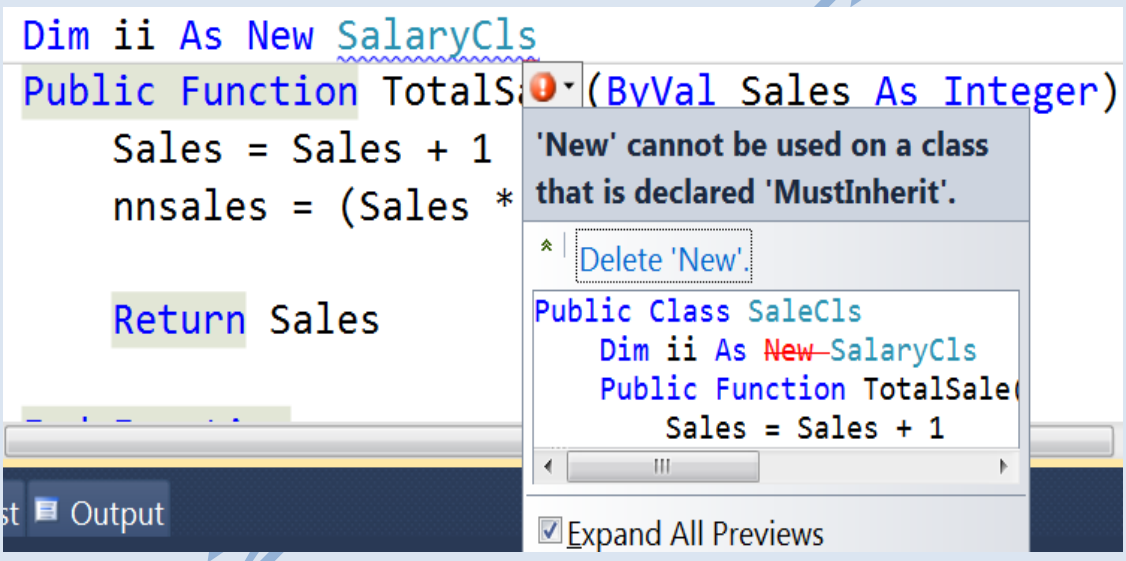
هي عبارة عن تصنيفات لا تسمح بانشاء اي كائن وتجبر المستخدم على ان يرث فقط التصنيف ويتم عمل ذلك على القاعدة الاتية

```
[public – private] Mustinherit Class ClassName
```

ستلاحظ ان قاعدة انشاء التصنيف المجرد نفس القاعدة انشاء التصنيف العادي لكن ستجد اننا نضيف الكلمة المفتاحية MustInherit كما هو موضح في الشكل الاتي

```
Public  MustInherit Class SalaryCls

العناصر التي نريد ادخالها

End Class
```

واذا حاولت ان ان تنشئ كائن ستجده يرفض انشاء الكائن وتظهر علامة خطا وهو خط ازرق كما في الشكل الاتي



3 – مبدا تعدد الاشكال Polymorphism

تعدد الاشكال تعني تعدد العنصار التي تحمل اسم واحد يعني ان يكون هناك دالتان تحملان نفس الاسم وليكن CostAcc ستجد ان هناك دالتان تحملان نفس الاسم ولكن كل دالة لها طريقة عمل مختلفة عن الاخرى . وهناك نوعان في عملية تعدد الاشكال

1- تعدد الدوال داخل التصنيف ويسمى التحميل الزائد (OVERLOAD)

مبدا التحميل الزائد هو وجود اكثر من دالة تحمل نفس الاسم داخل التصنيف ولكن لكي تتمكن من تطبيق هذا المبدأ يجب عليك ان تراعي الشرط الاتي

- يجب ان تختلف عدد المعاملات وانواعها من دالة لاخرى
- ان يتختلف نوع دالة عن دالة اخرى في حالة تشابه في عدد المعاملات و انواعها

- وجود كلمة overload قبل الدالة

ولكي تستطيع فهم ذلك قم بعمل اي دالة في التصنيف الذي تريده وليكن SalaryCls الذي انشاناه سلفا ولتكن الدالة هي Commsion سنقوم بعمل الدالة على الشكل الاتي

```
Private Sub Commsion()

End Sub

Private Function Commsion(ByVal i As Integer)

End Function
Private Function Commsion() As Integer

End Function
```

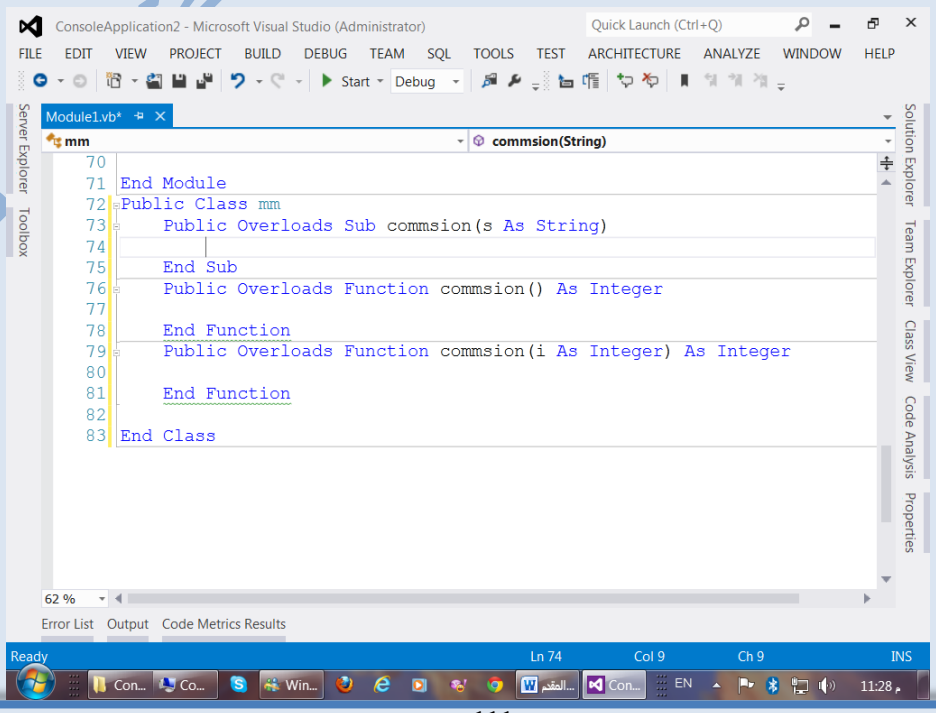
'Private Sub Commsion()' and 'Private Function Commsion() As Integer' cannot overloa each other because they differ only by return types.

في الكود السابق اعلنا عن ثلاثة دوال من نفس الاسم ولكن يجب ان تكتب الكلمة OverLoad قبل كل دالة والا يسظهر لك ويظهر الخط الازرق كما في الشكل السابق يجب اضافة كلمة

فيكون الكود كلاتي

```
Public overload function Commsion()As string

End function
```

فيكون الكود كلاتي



تلاحظ في في الكود السابق يجب ان تحتوي كل دالة او اجراء على معامل يختلف عن الاخر بحيث اما تحتوي كل دالة على معامل يختلف في نوعه عن الاخر او عدد معاملات في اي دالة تختلف عن دالة اخرى

وعند استدعاء اي دالة ستجد ظهور الكود في الصورة الاتية

على المربع المعلم عليه بالرقم 1 هو عبارة عن عداد يحدد عدد الدوال التي تحمل نفس الاسم ويتيح لك اختيار اي دالة تريد لاستدعاءها و استخدامها فيتيح لك عدة خيارات من الدوال التي تختلف في عدد المعاملات او نوعها

2- تعدد الدوال من نصنيف مشتق من اخر وتسمى اعادة القيادة(OVERRIDLY)

هذا المبدأ هو عبارة عن وجود اكثر من عنصر متاشهه في نفس الاسم ولكن ليس في تصنيف واحد ولكن في تصنيف مشتق عن تصنيف اخر مع العلم انه من المسموح ان تكون العناصر متشابهة في نفس الاسم و النوع و عدد المعاملات و انواعها ولكي تستطيع ان تفرق بين العنصر الموجودة في التصنيف المشتق عن التصنيف الاساسي في حالة واحدة هي انك اذا انشأت متغير يحمل نوع التصنيف الاساسي فانه يستخدم العنصر الموجودة فيه وكذلك في التصنيف المشتق نستخدم العنصر الموجود فيه

مثال لو اننا لدينا تصنيفان A,B و التصنيف B مشتق من التصنيف A ويوجد دالة رقمية C في التصنيف A فقمت بعمل اعادة قيادة لها في التصنيف B بحيث ان يعطي قيمة مختلفة وتريد ان تسخدم الدالة C في التصنيف B فانك تنشئ كائن من النصيف B واذا كنت تريد الدالة C التي في التصنيف A فانك تنشئ كائن من التصنيف A

ولكن يجب ان تحقق الشرط الاتي

- استخدام كلمة overridable في العنصر الموجود في التصنيف الاساسي
- استخدام كلمة Overrides في العنصر الموجود في التصنيف المشتق

ولكي تستطيع الفهم اكثر تابع معي المثال الاتي

افتح مشروع وسنضيف تصنيفين واحد اساسي واسمه A واخر مشتق وسيكون B واستخدام العنصر وليكن دالة واسمها C وسوف نتأكد من الدالة المستخدمة عن طريق الدالة main وسوف تستطيع فهم ذلك من متابعة الكود الاتي :

```
Module Module1
    Sub Main()
        Dim a As New A
        Dim b As New B
        Console.WriteLine("Function C : From A " + a.C.ToString)
        Console.WriteLine("Function C: From B " + b.c.ToString)
        Console.ReadLine()
    End Sub
    Public Class A
        Public Overridable Function C() As Integer
            Return 555
        End Function
    End Class
    Public Class B
        Inherits A
        Public Overrides Function c() As Integer
            Return 22
        End Function
    End Class
End Module
```

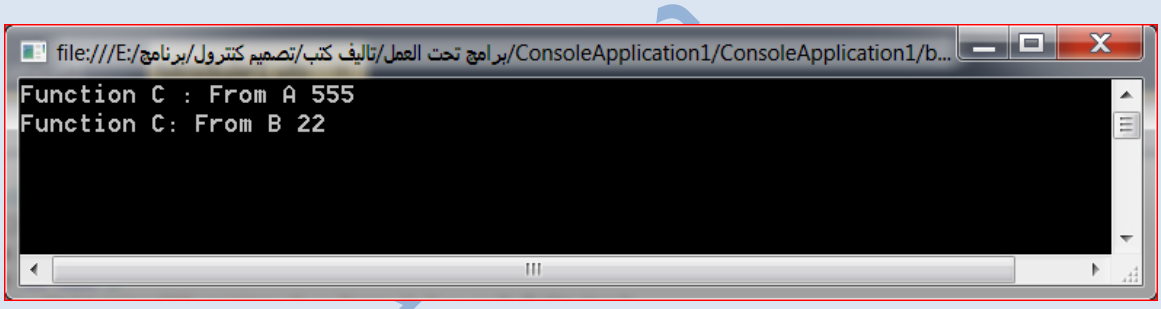
استدعاء الدالة C من التصنيفين B, A

الدالة C من التصنيف الاساسي A

الدالة C من التصنيف الفرعي B

و

وعند تشغيل البرنامج تلاحظ الاتي



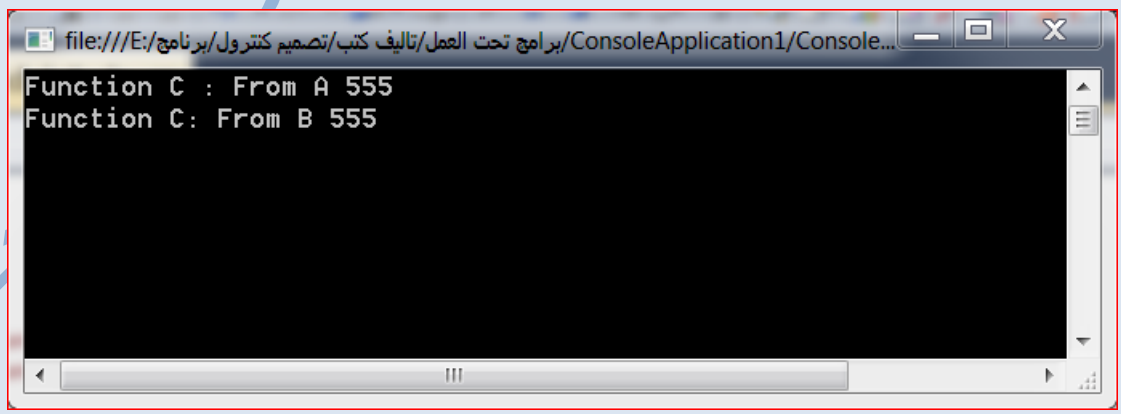
تلاحظ اننا انشائنا كائنين كائن من التصنيف A واخر لB واذا استخدمنا الدوال فاننا نجد كل كائن يستخدم الدالة C التي بداخله

واذا الغيت الدالة c من التصنيف B واعدت تشغيل البرنامج ستلاحظ الاتي

```
Module Module1
    Sub Main()
        Dim a As New A
        Dim b As New B
        Console.WriteLine("Function C : From A " + a.C.ToString)
        Console.WriteLine("Function C: From B " + b.c.ToString)
        Console.ReadLine()
    End Sub
    Public Class A
        Public Overridable Function C() As Integer
            Return 555
        End Function
    End Class
    Public Class B
        Inherits A

    End Class
End Module
```

في الكود السابق هو نفس الكود الذي كتبناه في اول المثال ولكن مسحنا الدالة التي في التصنيف B

كما ستلاحظ في الشكل الاتي اننا الشكل الاتي



ستجد اننا استخدمنا نفس الدالة التي في التصنيف A و التصنيف B ولكننا نجد ان الناتج من الدالة في التصنيف A وذلك لان B ترث من A والعنصر C في A هو نفسه العنصر في B لاننا لم نقم بعمل اعادة قيادة للعنصر في التصنيف B من التصنيف A

هناك كلمتين نستخدمهما بدل من overridable للعنصر داخل التصنيف الاساسي بكلمة Mustoverride هي عبارة عن كلمة تجبر عند اشتقاق التصنيف الاساسي والذي يجب ان يكون تصنيف مجرد mustinherit بحيث بعد اشتقاق الاتصنيف يجب ان تقوم باعادقة قيادة للدالة

كما يوجد كلمة اخرى هي NotOverridable هي تعني ان العنصر غير قابل للقيادة ولكن تكتب هذه الكلمة في التصنيف المشتق وتكتب قبل كلمة overrides فيكون شكل العنصر كلاتي

```
Public NotOverridable Overrides Property m
```

عناصر الظلSHADOW

نستخدم كلمة Shadow عندما نريد ان نستخدم عنصر سواء كان دالة او اجراء او خاصية بحيث نعيد قيادتها ولكن نكتشف ان مصمم التصنيف الاساسي لم يعطها قابلية اعادة الفيادة بمعانى ادق لم يكتب اثناء الاعلان عن العنصر كلمة overridable وتركها خالية نضطر ان نكتب كلمة shadow فمن المثال السابق

لو كانت الدالة C الموجودة في التصنيف A لم يكتب قبلها Overridable ونريد ان نعيد قيادتها في التصنيف B فاننا نعلن عن الدالة C كلاتي

```
Public Shadows Function C()  As integer

End function
```

انشاء التصنيف الاساسى من التصنيف الفرعى

في الامثلة السابقة التي شرحنها في التصنيف A, التصنيف B نفترض اننا نريد ان ننشئ التصنيف A من خلال التصنيف B بحيث نستخدم الدوال overridable التي تخص التصنيف A من خلال التصنيف B فاننا سنستخدم الكملة myBase كما ستلاحظ في الكود الاتي

```
Public Class A
    Public Function c() As Integer
        Return 55
    End Function
    Sub New()

    End Sub
End Class
Public Class B
    Inherits A
    Sub New()
        MyBase.New()
    End Sub
End Class
```

انشاء كائن التصنيف الاساسي من داخل التصنيف الفرعي

وبذلك نكون قد انتهينا من شرح مبادئ البرمجة بالكائنات الموجهة وسوف ننهي الباب الاول بشرح بعض الحاويات البرمجية كالواجهات و الهياكل

الهياكل STRUCTURE

الهياكل تشبه تمام الشبه بالصنيفات بحيث انها تحتوي على عناصر الكائن كالخصائص و الاجراءات و الدول ويشهر عنه بحمل المتغيرات فهومثل التصنيفات حتى وقت الاستدعاء يمكن انشاء كائن واعطاء قيام افتراضية ... ولكن يختلف عن التصنيفات في النقاط الاتية

- انه لا يرث و لا يورث لاي تصنيف او حتى الهيكل نفسه
- لا تستطيع استدعائه خارج فضاء الاسماء حتى لو كان يحمل كلمة استدعاء public
- لا يمكن انشاء اي كائن داخله
- لا يمكن اعضاء قيم افتراضية لاي متغير مثل Dim I as integer = 5 هذا الكود غير مسموح به في الهيكل لكن يجب ان تعلن عن المتغير بدون اعطاء قيمة Dim I as integer

لتوضيح ذلك قم بعمل هيكل في اي مشروع جديد ويتم عمل ذلك بالتباع القاعدة الاتية وطبعا تشبه التصينفات

```
[Public – private] Structure Structure Name

Members


End structure
```

Public – private تحدد مدى الوصول للهيكل

Structure Name اسم اليهكل

Members العناصر داخل الهيكل

ولتوضيح ذلك قم بعمل هيكل عن طريق اتباع الخطوات الاتية

اكتب الكود الاتي في اي مكان في التصنيف او خارجه وانا ساكتبه خارج التصنيف لكي تتعرف اكثر عليه

```
Public Structure mmd
    Sub New(ByVal ii As Integer)

    End Sub
    Private Sub mm()

    End Sub
    Public Property jj As Integer

End Structure
```

انظر قمنا بكتابة هيكل بشكل مجازي سوف نقوم بتطبيق النقاط التي ذكرناها في تعريف الهيكل فافتحي اي تصنيف واكتب الكود الاتي

```
Public MustInherit Class SalaryCls
    Inherits mmd
    Public Su
```

Classes can inherit only from other classes.

ستجد علامة الخطا توضح انه التصنيف لا يرث الا من التصنيفات مثلها .... حسنا قم بعمل توريث لهيكل اخر فانشئ هيكل اخر وحاول التوريث

```
Structure NewStruct
    in
    Private Sub mm()
```

Interface
MustInherit
NotInheritable

Interface statement
Declares the name of an interface and the definitions of the members of the interface.
Note: Tab twice to insert the 'Interface' snippet.

لاحظ كلمة inherit لا تظهر في قائمة المساعدات فيدل على انها غير موجودة

قم الان بانشاء هيكل داخل اي تصنيف فستجد انه يرفضه ويرفض ايضا اي اضافة قيم افتراضية في الهيكل

وعند استدعاء الهيكل تستطيع انشاء أي كائن باستخدام الدالة new او باعلان عن أي متغير يحمل نوع الهيكل كما هو موضح في الكود الاتي

```
Public Class AA
    Private Sub ffd()
        Dim m As New BB
        Dim e As BB
    End Sub
End Class
Public Structure BB
    Dim i As Integer
    Private Sub c()
    End Sub
End Structure
```

استدعاء الهيكل

الهيكل

INTERFACE الواجهات

ذكرنا في قسم التوريث اننا لا نتسطيع عمل توريث من تصنيفين فان الدوت نت لا يدعمها لذا ظهرت فكرة عمل الواجهات و الواجهات ما هي الا عن حاوية كود لكن لا تحمل الاكواد لكن تحمل ما يسمى توقيعات العناصر بمعنى انه تعطيك فقط الشكل الخارجي للعناصر مثلا تعطي اسم الخاصية و المعاملات التي تدخل فيها ولكن لا تحمل اي كود داخلها كما هو موضح في الكود الاتي

```
Public Interface AlBaraa
    Property BooksName As String
    Function BookSerial(ByVal BookNo As Integer) As Integer
End Interface
```

تلاحظ اننا اتبعنا القاعدة الاتية

```
[Public - private] interface  interface_Name
Members
End interface
```

Public - private تحدد مدى الوصول الواجهة

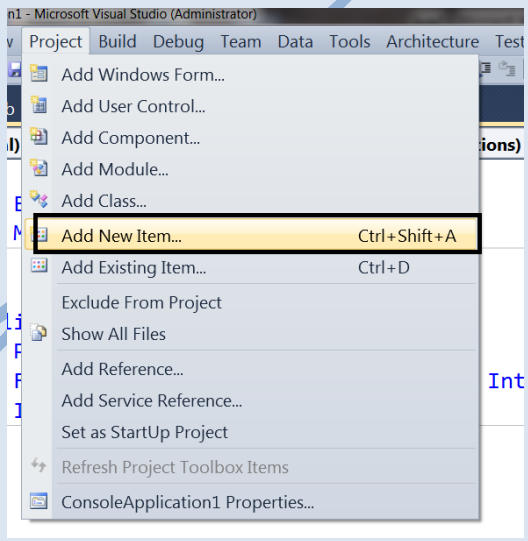
Interface Name اسم الواجهة

Members العناصر داخل الواجهة لكن تلاحظ في الكود السابق اننا لان نضع محددات الوصول يعني لا يحدد فيها اذا كان العنصر خاص او عام او صديق لكن نكتب ما هو العنصر و اسمه و نوع البيانات و المعاملات فقط

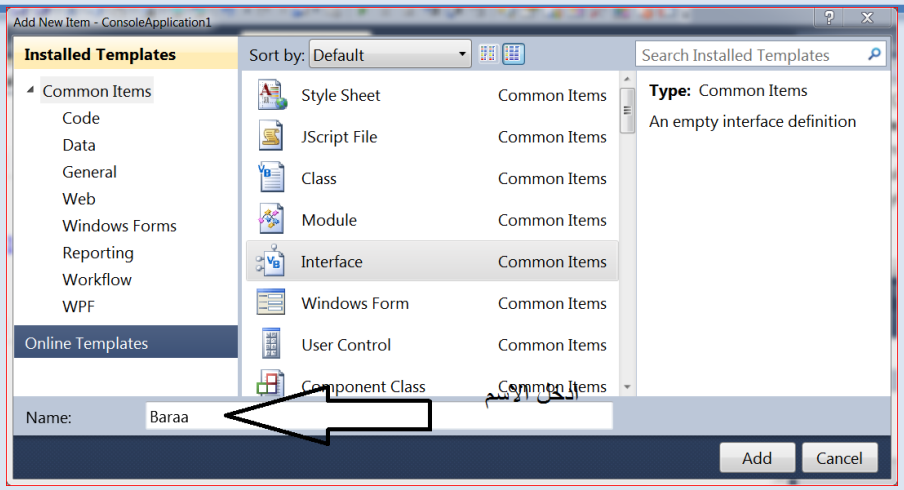
ويمكن الاعلان عن الواجهات بعدة طرق اهما كتابة الكود بشكل مباشر في اي مكان داخل او خارج التصنيف

او يمكن عمل ملف مستقل داخل المشروع كما هو موضح في الخطوات الاتي

- فتح قائمة Project ← add new iterm كما هو موضح في الشكل الاتي



- ثم ننتقل لمربع النص ونحدد الواجهة ونكتب اسم الواجهة كما هو موضح في الشكل الاتي



-

فاذا اردنا استدعاء الواجهة فعلينا فقط كتابة كلمة implement ثم اسم الواجهة وسوف تجد انها قامت بكتابة عناصر الواجهة داخل التصنيف او الهيكل كما هو موضح في الشكل الاتي

```
Public Class NewClass
    Implements AlBaraa
    Public Function BookSerial(ByVal BookNo As Integer) As Integer Implements AlBaraa.BookSerial

    End Function

    Public Property BooksName As String Implements AlBaraa.BooksName
        Get

        End Get
        Set(ByVal value As String)

        End Set
    End Property
End Class
```

تطبيق الواجهة

2

2

1

في الكود السابق :

كلمة تطبيق الواجهة التي في الرسمة ، توضح كيف يتم استدعاء الواجهة ويمكن استدعاء واجهات اخرى

في الرقم 1 : يوضح ان العناصر التي تم ادراجها داخل التصنيف تابعة للواجهة التي تم تطبيقها داخل التصنيف لانه يمكن ادخال عدة واجهات في تصنيف واحد.

في الرقم 2 : يوضح ان العناصر التي تم ادراجها داخل التصنيف .

التعدد ENUM

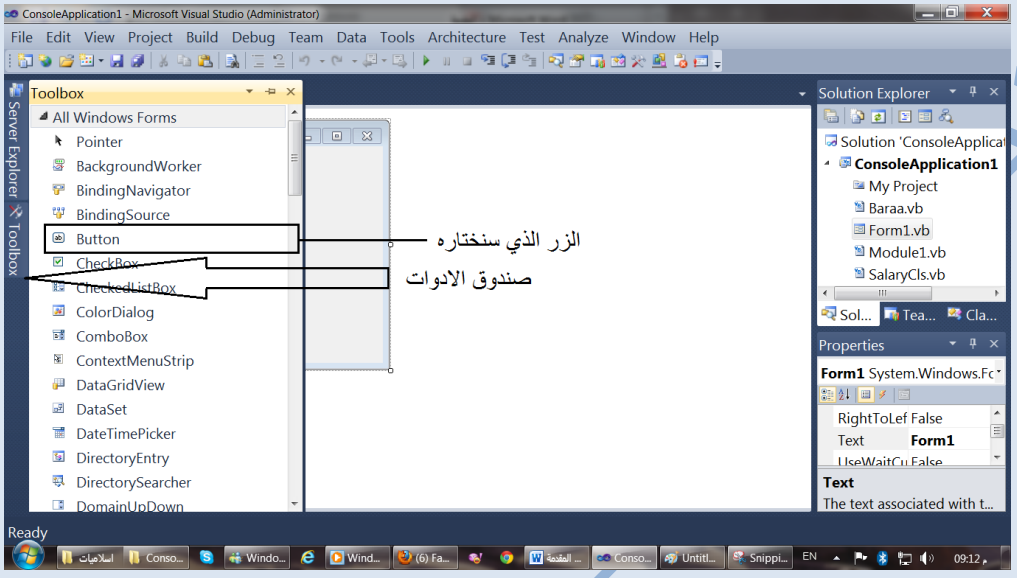
التعدد هو عبارة عن حاوية كود لكن لا تحتوي على عناصر لكن مسميات تتيح للمبرمج الاختيار فيها ..طبعا لا تفهم شيء تابع معي المثال الاتي .

هل تذكر الخيارات التي نختارها في جملة الاختبار Select case التي شرحناها في جمل التدفق بالاخص جمل الشرط فكنا نقول في حالة اذا كان المتغير X =1 سيكون كذا وكذا واذا كان X = 2

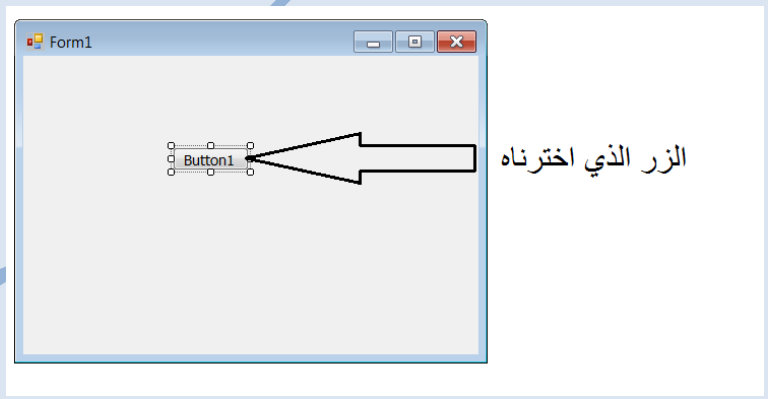
وهكذا فبدل من ان نضع ارقام لاننا قد ننسى الارقام على ماذا ترمز فيتم وضع رموز مكانها حتى لا ننسى الهدف من وضع الارقام ويمكن توضيح ذلك من المثال الاتي

سوف نقوم بعمل اختبار عن التقدير الذي اخذه الطالب فلو كان ممتاز ستظهر رسالة تقول انت ممتاز ، فلو كان جيد جيد تظهر رسالة تقول شد حيلك ، فلو كان التقدير جيد تظهر الرسالة تقول تقدير شعبي ، فلو كان التقدير مقبول تظهر الرسالة لست جيدا .

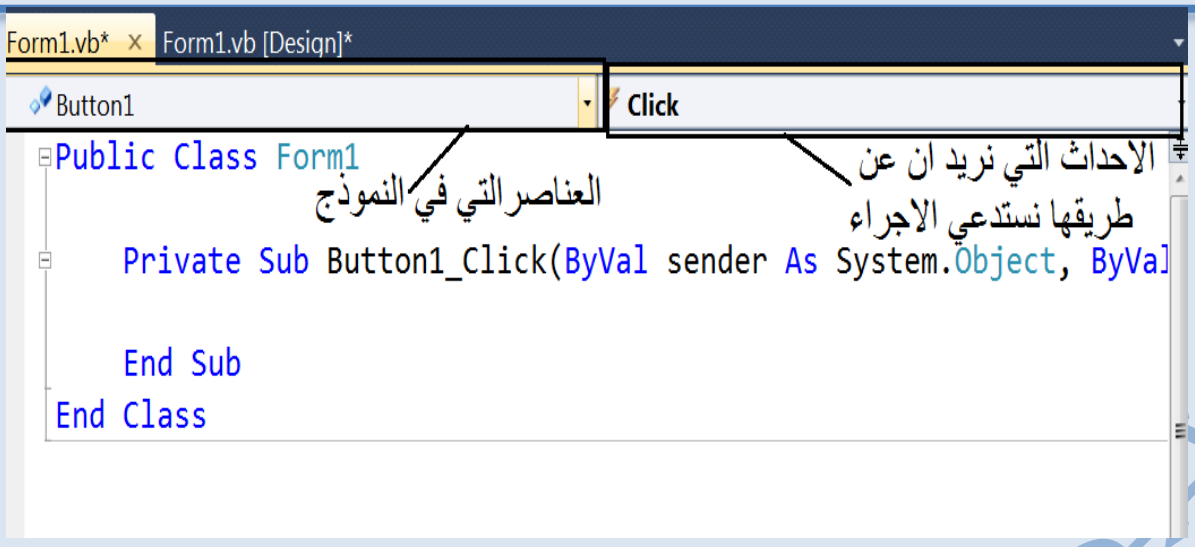
قم بفتح مشروع من نوع Windows Form واضف اليه الزر من صندوق الادوات سوف كما هو موضح في الشكل الاتي



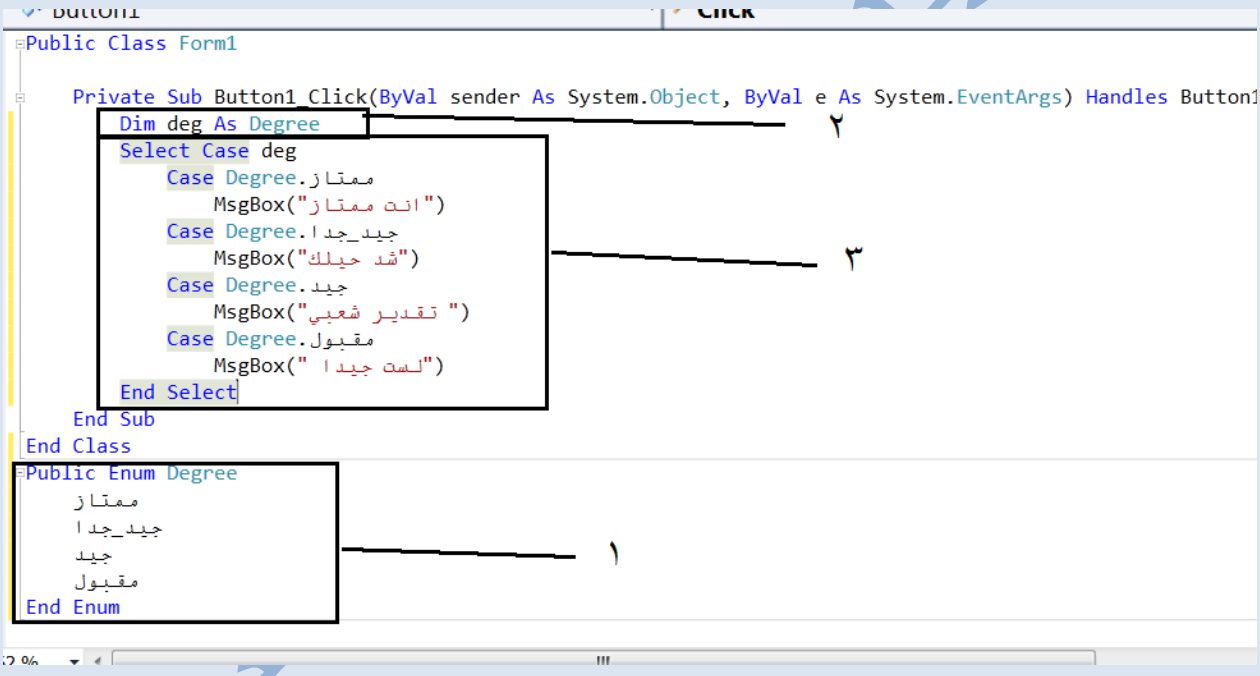
طبعا سنوضح بيئة تطبيقات الويندوز سوف نشرحها في التفصيل في الفصل القادم ولكن اريد من ان تعرف فقط اننا كيف نختار الادوات ونختار الاداة التي نريدها وننقر عليها مرتين ليتم اختيار الحدث Click كما هو موضع في الشكل الاتي



ننتقل الان الى شاشة الكود عن طريق النقر على الزر مرتين فتظهر لنا شاشة الكود



في الشكل السابق يوضح ان هناك قائمتين قائمة التي تختار العناصر الذي توجد في النموذج كالزر الذي نريده ، اما القائمة التي بجانبها تغبر عن الاحداث التي سوف نستدعي من خلالها الاجراءات وكما قلنا سنشرحها بالتفصيل في الفصل القادم ولكن اؤهلك كي تستطيع استيعاب فقط الان سوف نكتب الكود الاتي في شاشة الكود



هذا الكود كتبناه دخال الاجراء button1_Click الذي يتم استدعائه عندما ينشط الحدث Cliek الذي نشط عند ضغط المستخدم للماوس .

في الشكل السابق هناك 3 ارقام سنوضحها كلاتي

1 هذه اداة الترقيم الذي تضع فيه العناصر كما نوضح يمكن *كتابته* عناصر باللغة الغربيه ولكن عند كتابة المسافة نستبدلها بعلامة (_) الذي يسمح فقط عند التسمية كما وضحنا سابقا في فصل المتغيرات وعند انشاء التعدد يمكن اتباع القاعدة الاتية

```
[public - private] Enum Enum Name
Members
End Enum
```

[public - private] محددات الوصول للتعدد

Enum Name اسم التعدد

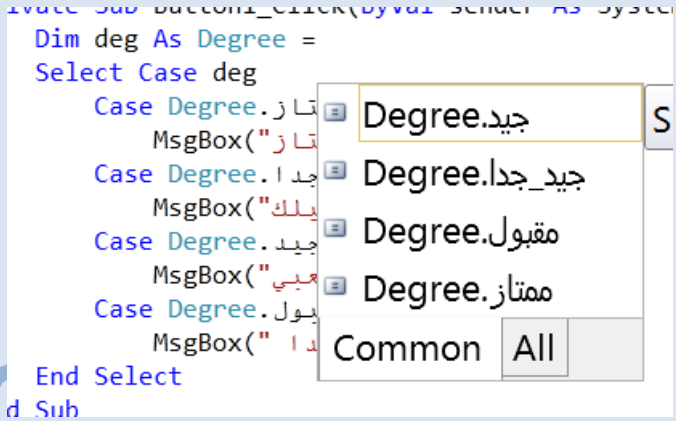
Members هي العناصر توضح الاختيار وكما وضحنا في المثال السابق نكتب فقط المسميات

2 هي عملية استدعاء الكود وهي عن طريق اعلان عن متغير يحمل نوع التعدد

```
Dim deg as Degree
```

حيث ان Deg هي اسم المتغير و Degree هي التعدد التي نختارها

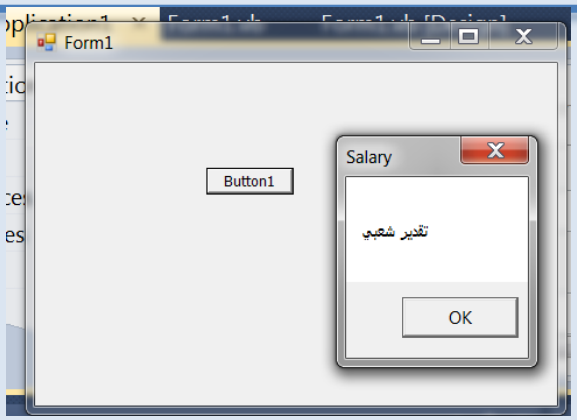
3 هنا نكتب الكود الذي يقوم بالاختبار وعند كتابة الكود نلاحظ ان قائمة المساعدات تعطيك الاختيارات وهي العناصر التي توجد في الترقيم كما هو موضح في الشكل الاتي



طبعا اذا شغلنا البرنامج فلن تظهر اي رسالة فيجب اعطاء قيمة للمتغير Deg ولتكن القيمة (جيد) كما هو موضح في الكود الاتي

```
Dim deg as Degree = Degree.جيد
```

وعند تشغيل البرنامج تظهر لك الرسالة كما هو موضح في الشكل الاتي



الى هنا قد انتهينا من الباب الاول الذي يشرح مبادئ البرمجة سننتقل الى الباب الثاني الذي يكون متقدما قليلا لشرح كيفية تصميم ادوات التحكم مثل الزر الذي ادرجناه على النموذج في المثال السابق

البدايات المضيئة

الباب الثاني

تصميم ادوات التحكم

بعدما استعرضنا عن مبادئ البرمجة سنقوم الان باستعراض كامل عن كيفية عمل تصميم ادوات التحكم حتى تتمكن من عمل ادوات تساعدك في تطوير تطبيقاتك وتسهل عليك عمل تطبيق قوي و يعمل بكفائة ففي هذا الباب سوف نستعرض الفصول الاتية

- الفصل السادس : ما هي الواجهة الرسومية وما هي اهم الادوات في النموذج
- الفصل السابع : تصميم ادوات التحكم
- الفصل الثامن : تحسين اداء ادوات التحكم
- الفصل التاسع : تصميم زر

الفصل العاشر : اكتشاف الاخطاء وتصحيحها

الفصل الحادي عشر : اكود يجب عليك معرفتها

البدايات المضيئة

الفصل السادس

مقدمة في الواجهة الرسومية و تطبيقات النوافذ

وبعد ان استعرضنا مبادئ البرمجة تعال معي نستعرض ثمرة ما تم وضعه من مبدائ برمجة الكائمنات الموجة فاننا سوف نستعرض بيئة تطوير التطبيقات النوافذ من برنامج الفيجوال استوديو 2012 , ولو كنت مستخدم فيجوال ستوديو 2010 او 2008 لن تجد اختلاف كثير واي اختلاف سوف اوضحه لك

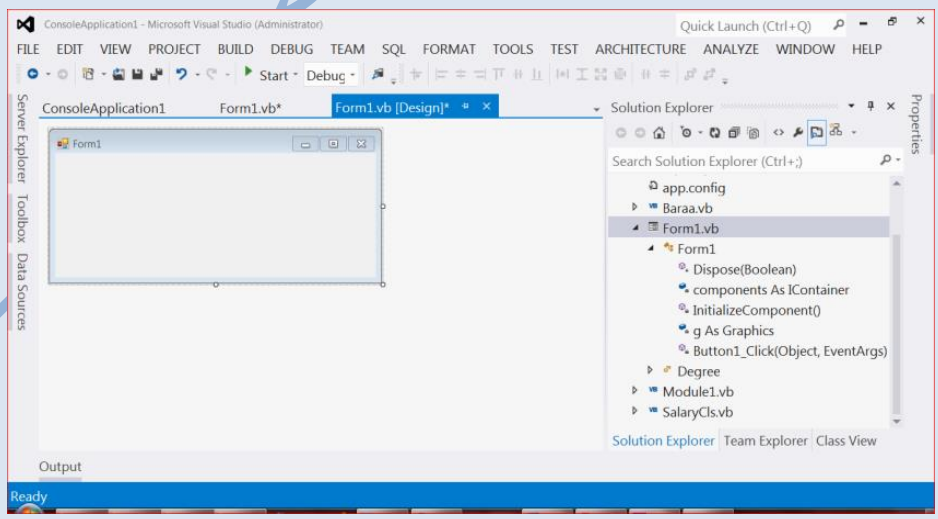
كنا اعطينا في الفصل الاول نبذة عامة عن الواجهة في تطبيقات windows ففي هذا الفصل سوف نخوض في تفاصيل الواجهة وسوف نبدا النموذج لنتعرف على مكوناته

النموذج FORM

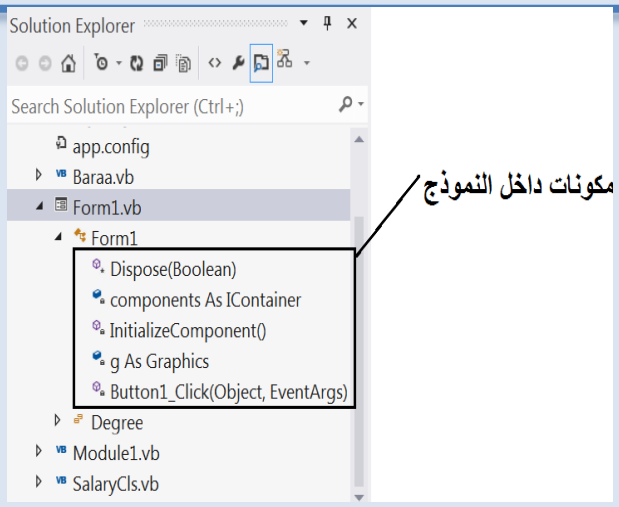
في الفصل الاول تكلمنا عن المكتبة التي طورتها شركة microSoft التي استخدمت في تطوير كل تطبيقاتها وهي مكتبة netFramWork و التي تساعد كل المطورين بلغة الفيجول استوديو الدوت نت في تطوير تطبيقاتهم وكنا تكلمنا بعض فضاءات الاسماء التي مكتبة netFramwork من اهما الفضاء System و الذي يندرج تحته فضاء اسماء اسمه windows و يندرج تحته الفضاء Windows فضاء اسماء اسمه Forms من هذا الفضاء سنجد التصنيف Formيقوم بتطوير كل تطبيقات WindowsApplication التي نشرحها الان فيكون المسار كلاتي

```
System.Windows.Forms
```

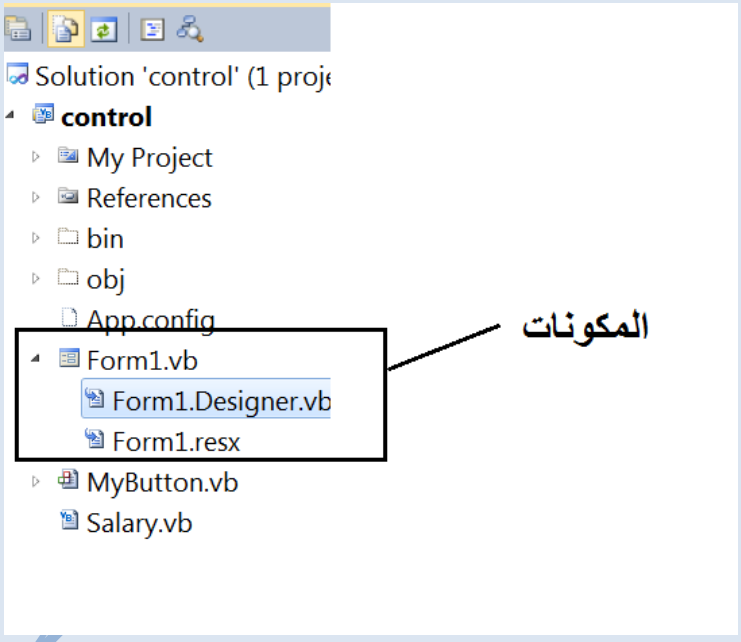
لكي نتعرف اكثر على النموذج اكثر سوف نفتح مشروع جيدد من نوع Windos Application ليكون الشكل الاتي



اريدك ان تنظر الى نافذة البحث داخل المشروع Seluton Explorer لاحظ الصورة الاتية



هذه النافذة تختلف في الشكل الخارجي عن نوافذ المتصفح الموجودة في الاصدارات السابق كما هي موضحة في الشكل الاتي الذي يوضح نافذة المتصفحة في الفيجوال استوديو 2010



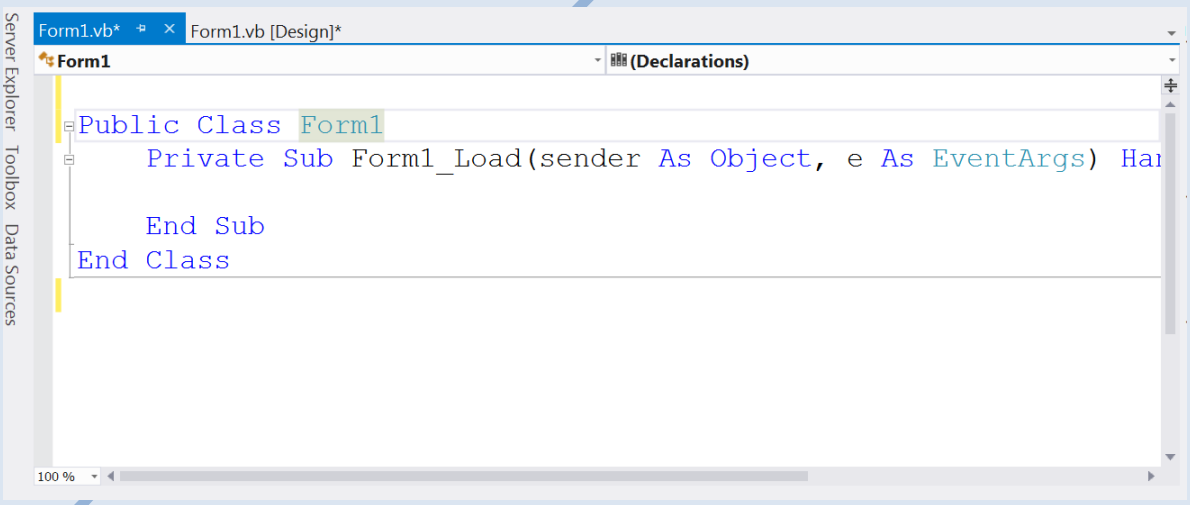
هو اختلاف في الشكل لكن تريد ان تتعرف ستجد في التصنيف Form سوف تجد ان هناك تصنيفين التصنيف الاول هو Designer كما هو في الشكل السابق وهو يقوم بكتابة تلقائية للكود المسؤول عن تصميم النموذج وتصميم اشكال الادوات على النموذج في وقت التصميم

قم باضافة زر على النموذج ستلاحظ انه وضع الزر تلاقئية في المنوذج وكتب الكود الاتي داخل Desgner

```
'NOTE: The following procedure is required by the Windows Form Design
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.SuspendLayout()
    '
    'Button1
    '
    Me.Button1.Location = New System.Drawing.Point(233, 89)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(75, 23)
    Me.Button1.TabIndex = 0
    Me.Button1.Text = "Button1"
    Me.Button1.UseVisualStyleBackColor = True
    '
    'Form1
    '
```

ستجد بشكل تلاقئي اضافة الاكواد اللازمة لتصميم النموذج في الملف الخاص بالتصميم Designer

ولكن اذا اردت برمجة الزر ستجد نافذة كود اخرى وهيا لنافذة التي نستخدمها لبرمجة ادواتنا تطوير تبطيقاتنا وهي تستطيع الوصول اليه من خلال الضغط على النموذج مرتين فتظهر كما في الشكل الاتي



اما الشكل الخارجي للشاشة الكود المسؤول عن اتصميم Designer النموذج فهي يمكن ان تراها في الشكل الاتي

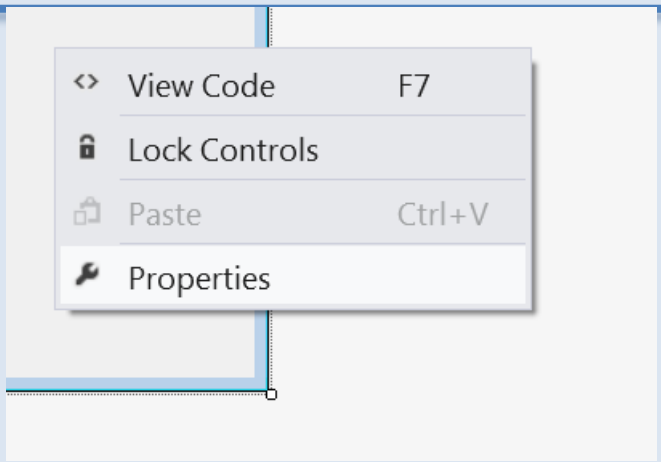
```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Form1
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        '
        'Form1
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(7.0!, 16.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(552, 240)
        Me.Name = "Form1"
        Me.Text = "Form1"
        Me.ResumeLayout(False)

    End Sub
```

استيراد فضاء الاسماء

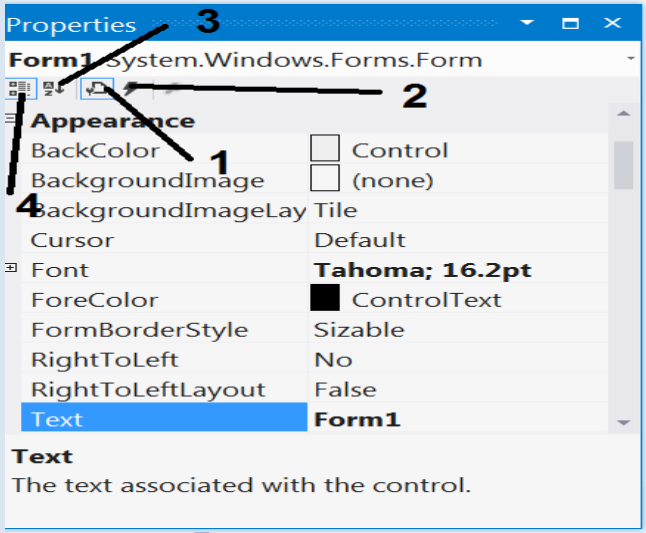
هذا شكل الكود من داخل الملف Designer للنموذج قبل اضافة اي ادوات وقبل تغيير اي خصائص للنموذج ستجد ان النموذج عبارة عن تصنيف Class يرث من التصنيف الاب Form الموجود داخل فضاء الاسماء Forms كما هو في اعلى التصنيف

خلاصة القول في كل ما سبق هو اننا عند تطبيقات النموذج فان لدينا ملفين الكود ملف امتداده هو Designer.vb مهمته كتابة الكود المسؤول عن تصميم النموذج ... وملف اخر يكون امتداده هو resx في هذا الملف نقوم نحن بكتابة الكود الذي نريده كتابته

سنقوم الان باستعراض اهم الخصائص التي في النموذج ولكن قبل ان نتكلم في الخصائص اريدك ان ترى كيف نظهر الخصائص وتعديلها في بيئة التصميم في الفيجوال ستوديو فلديك عده خيارات لكي تظهر نافذة الخصائص لديك عدة طرق لاظهار النافذة وهي موجودة في كل اصدارات الفيجوال استوديو دوت نت بحيث تنقر بالزر الايمن للماوس ثم تختار الخيار الاخير propireies كما هو موضح في الشكل الاتي



او تضغط على الزر F4 ستظهر لك النافذة الاتية



سوف نشرح هذه الادوات التي قمت بترقيمها

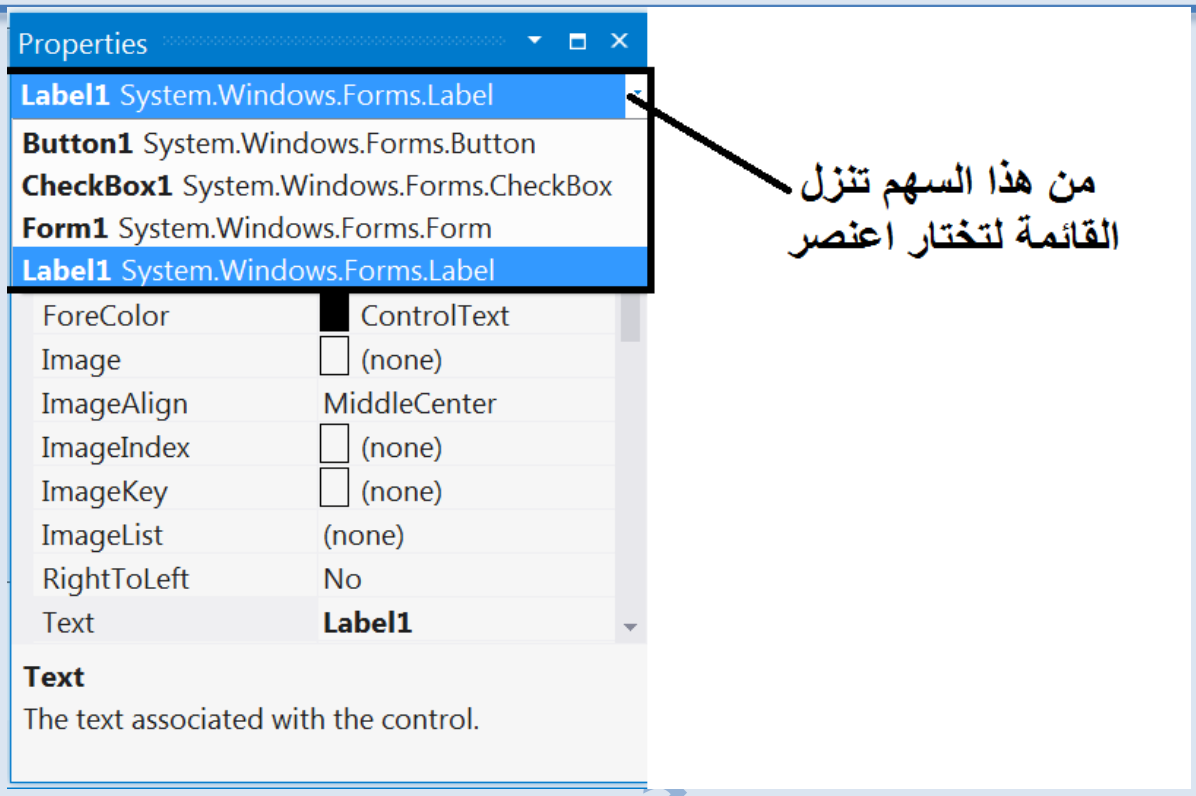
رقم 1 هي تظهر الخصائص التي تخص النموذج داخل نافذة الخصائص فتجد في العمود الاول الخصائص و العمود الثاني القيم التي نريد تعديلها

رقم 2 هي تظهر الاحداث التي تخص النموذج داخل النافذة فتجد في العمود الاول الاحداث و في العمود الثاني الاجرائات التي تستدعى اثناء تنشيط الحدث

رقم 3 تقوم بترتيب الخصائص او الاحداث على حسب الترتيب الابجدي

رقم 4 ترتب الاخصائص و الاحداث على اساس ترتيب بالمجموعات يعني الخصائص التي تخص التعديل في تصميم النموذج توضع في مجموعة واحد وتسمى مثلا Designe وتسمى المجموعة ب Category

لو اردت ان تنتقل من عنصر لاخر للتعديل في بعض خصائصه فانك اما ان تختار العنصر ثم تفتح نافذة الخصائص او من خلال قائمة العناصر في النافذة وهي في الشكل الاتي



والان نعود الى اهم خصائص في النموذج وهي تتمثل في الاتي

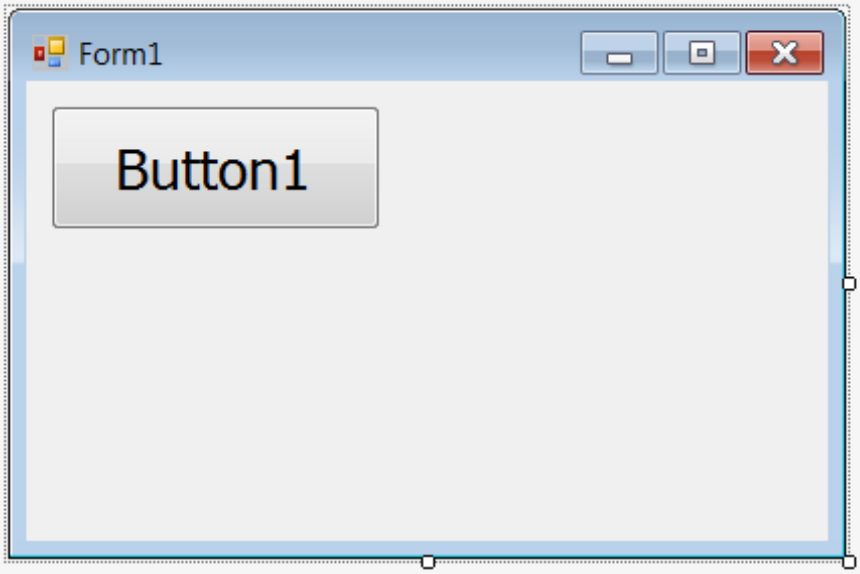
الخاصية	النوع
Name	هو اسم النموذج الذي يتم استدعائه في عملية البرمجة مثل اسم المتغيرات تتم التسمية بنفس الشرووط التي ذكرناها في تسمية المتغيرات
Text	هي كتابة اسم بحيث يظهر للمستخدم ولا يشترط فيه استخدام الشروط فهي تعتبر اسم الشهرة فلا تستخدم في البرمجة وهي تظهر للمتسخدم وتظهر قيمة هذه الخاصية في شريط العنوان على النموذج
BackColor	لعمل خلفية للنموذج
ForeColor	تغديل لون الخط على النموذج
Font	تعديل الخط على النموذج
RightToleft	قلب شكل المنوذج من اليسار لليمني اذا كانت القيمة True و من اليمين لليسار اذا كانت False
cursor	تحدد شكل الماوس على النموذج
Enable	هل يعمل النموذج
Visable	هل تظهر النموذج ام لا

WindowState	هذه الخاصية تحدد وضع النموذج هل هو في وضع تكبير ام تصغير عم الشكل العادي
StartPassion	بها خيارات تحدد بداية ظهور النموذج هل يقع في وسط الشاشة ام على يمينها وغير ذلك
FormBorderStyle	تحدد شكل النموذج يسمح بان يكون قابل تغيير الاطوال ام يكون بال شريط العنوان او يظهر بشريط العنوان ولكن لا يسمح بتغيير الاطوال
Opacity	هذه الخاصية رائع النها تتحكم بشفافية النموذج
BackgroundImage	هذه الخاصية تظهر خلفية النموذج ولكن تظهر صورة وليس لون
Size	هذه الخاصية تحدد طول وعرض الصفحة وتندرج تحتها خاصيتين هما height تحدد خاصية الطول و Width تحدد خاصية العرض
Location	هذه الخاصية تحدد موقع النموذج على الشاشة بحيث يندرج تحتها خاصيتين هما X تحدد مكان النقطة على يسار النموذج و y تحدد النقطة على النموذج كما يوجد خاصيتين تحددان موقع النموذج هي lift تحدد يسار النموذج و Top يحدد اعلى النموذج

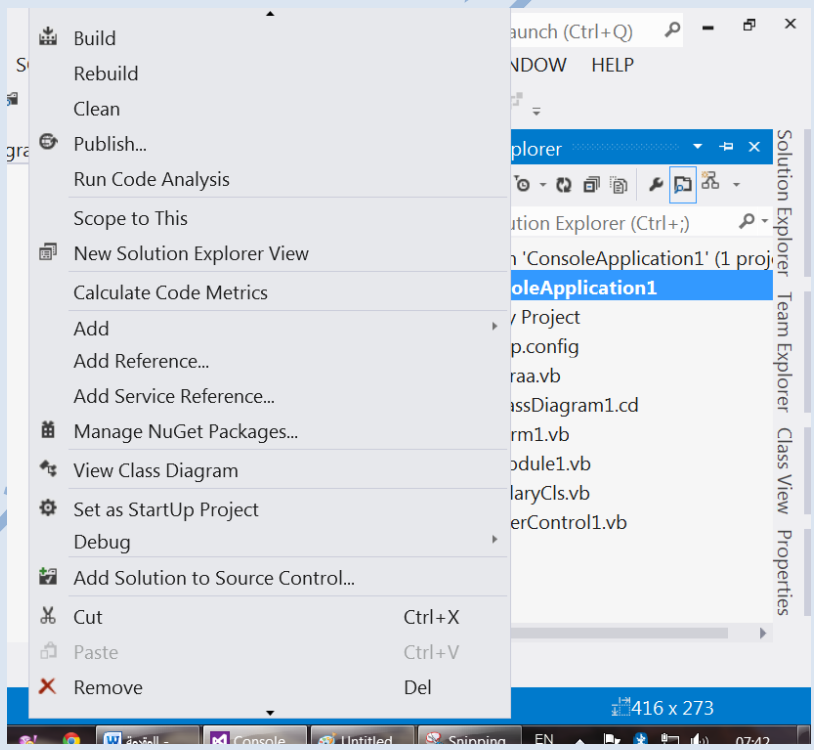
وغيرها من الخصائص ولكني استعرضت الخصائص التي يستخدمها كل مبرمجي تطبيقات الويندوز وقبل ان ننتقل الى الادوات التحكم يجب ان تعرف اهم الاجراءات التي نستخدمها في النموذخ والتي نستخدمها في كل تطبيقاتنا وهي

Shaw	اظهر النموذج وتحميله
ShowDailog	تظهر النموذج وتحميله ولكن تقوم بمنع المستخدم ان يستخدم اي نافذة اخرى غير النافذة النشطة
Close	اغلاق النموذج ومسحه من الذاكرة
CenterToScreen	امر بجعل النموذخ يتوسط الشاشة
Hide	اخفاء النموذج لكن يظل في الذاكرة

ولكي تستطيع فهم الفرق بين الاجرائين hide او Closse قم بفتح مشورع وندوز جديد ثم قم باضافة زر على النموذج حتى يكون شكله كلاتي



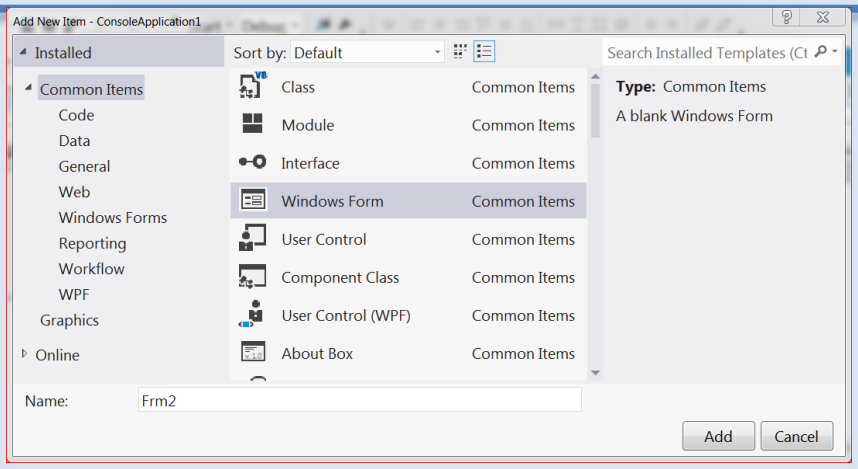
ثم نضيف نموذج اخر ولكي نصيف نموذج اخار اما ان نختار Add Windows Form من قائمة Porject او تقوم بالضغط بالزر الايمن على المشروع في نافذة التصفح Selution Explorer كما هو موضح في الشكل الاتي



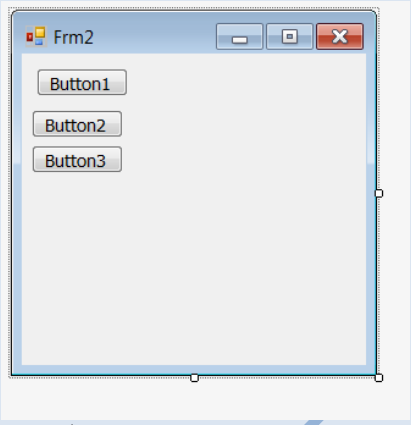
تلاحظ بعد الضغط تظهر قائمة اختصار هذه القائمة بها كل الادوات و الاوامر التي نستخدمها دائما في تطبيقاتنها سوف نخترا منها الاختيار Add فيظهر خيارات هذه نختار من هذه الخيارات Windows Form

سوف نشرح اهم الخيارات الي في القائمة بعد ان ننتهي من المثال

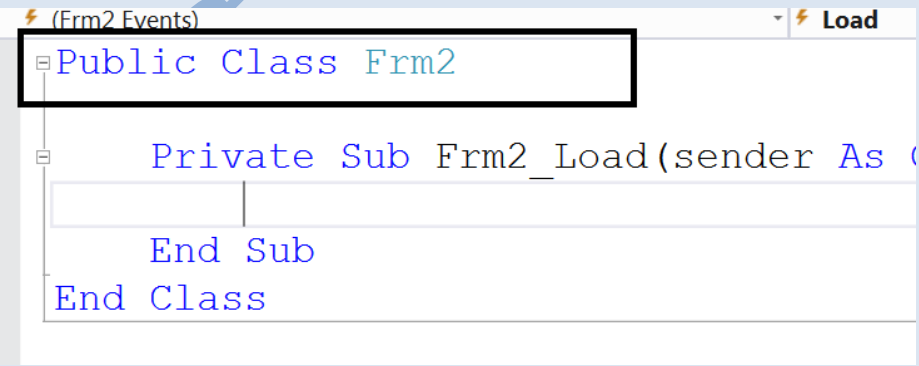
سيظهر لك صندوق حوار كنا قد شرحناه من قبل به فاكتب اسم النموذج الدي نريده وليكن Frm2 كما هو موضح في الشكل الاتي



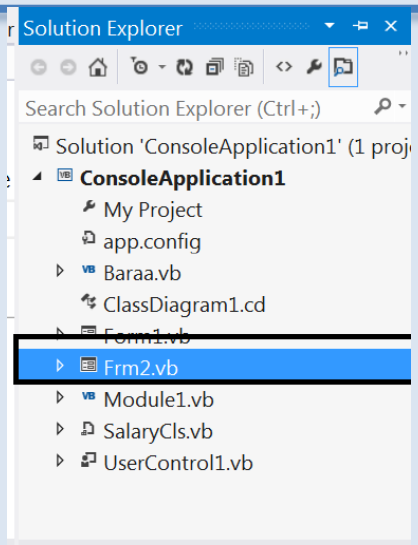
سنقوم بعض التضغط على الزر ok لتظهر لكن النافذة الثانية اضف اليها ثلاثة ازرار كما هو موضح في الشكل الاتي



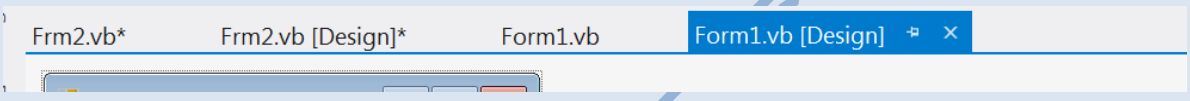
لو فتحت الخصائص التي تخص النموذج ستلاحظ ان الخاصية name هي Frm2 التي كتبتها وتلاحظ ان اسم التصنيف الخاص بالنموذج ايضا Frm2 كما هو موضح في الشكل الاتي



كما تلاحظ في نافذة التصفح وجود النافذة كما هو موضح في الشكل الاتي



انني اوضح لك كل هذا لانك سوف تحتاجها وانت تطور تطبيقاتك بحيث انك تريد ان تنتقل من Form1 الى Frm2 يمكن ان ننتقل خلال شريط اللسان Tabe كما هو موضح في الشكل الاتي



سنعود للمثال لكي نفرق بين الاجراءين هما Hide و Close

عد الى انافذة الاولى واكتب الكود الاتي في الحدثButton1_Click بحيث انك تضغط مرتين على الزر واكتب الكود الاتي

```
Private Sub Button1_Click(sender As Object, e As EventArgs)
    Frm2.Show()

End Sub
```

ثث

ثم ننتقل الى Frm2 ثم تكتب في الزر الاول الكود الاتي

```
Private Sub Button1_Click(sender As Object, e As EventArgs)
    Me.BackColor = Drawing.Color.Red

End Sub
```

ولكن ما رايك لو قمت بتعديل عده خصائص لاي تصنيف فانه من الافضل تستخدم جملة With التي يمكن كتابتها باتباع القاعدة الاتية

```
With ClassName
```

```
.ClassPoparty
End With
```

ClassName اسم التصنيف الذي نريد استخدامه

ClassPoparty _ خصائص التصنيف التي نريد تعديلها

فنقوم بتعديل خصائص النموذج باستخدام الجملة كما هو في الكود الاتية

```
With Me
    .BackColor = Drawing.Color.Red
    .Height = 222
    .Width = 222
End With
```

في الكود السابق عدلنا الكود الاسبق بحيث قمنا بتعديل الكود باستخدام كلمة With واستخدمنا النموذج الذي نحن فيه وبما اننا نستكلم عن التصنيف الذي نكن نكتب فيه فانه يجب ان نكتب كلمة Me بمعنى لو اننا نكتب كود داخل التصنيف (النموذج) اسمه Frm2 ونريد ان نعدل بعض خصائصه او نستدعي دوال او اجرائات تم انشاءها في داخله لذا يجب كتابة كلمة Me وليس Frm2

اما في السطر الثاني و الثالث و الرابع في الكود اعطينا خصائص النموذج وقمنا بتعديلها

ننتقل للزر الثاني ونكتب الكود الاتي

```
Private Sub Button2_Click(sender As Object, e As EventArgs)
    Me.Hide
End Sub
```

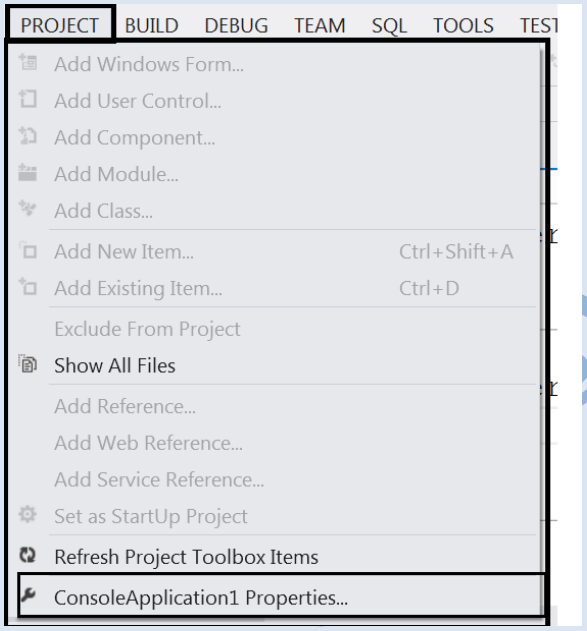
ثم ننتقل للزر الثالث ونكتب الاتي

```
Private Sub Button3_Click(sender As Object, e As EventArgs)
    Me.Close
End Sub
```

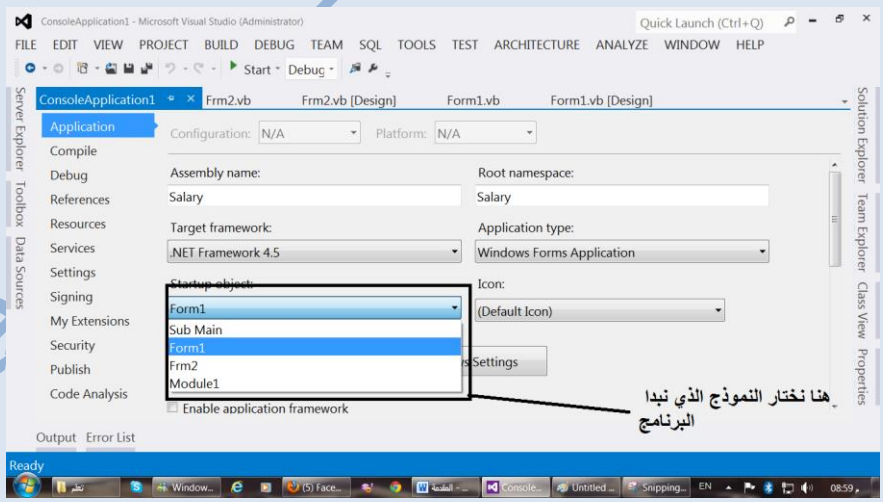
ننتقل الى تشغيل البرنامج ولكن قبل ان نشغل البرنامج نوضع اننا قمنا بتصميم البرنامج لكي نتاكد هل النموذج frm2 ونرمز له رمز 2 اننا اذا ضغطنا على الزر الاول فننا نغير خصائص النموذج ثم اذا ضغطنا على الزر الثاني الفنموذج

2 يختفي وتظل الخصائص متغيرة بسبب ان النموذج ما زال محتفظ بقيمته في الذاكرة اما الزر الثالث فانه يغلق النموذج يفقد تماما من الذاكرة

ولكي نقوم بشغيل البرنامج يجب ان يبدا البرنامج بالنموذج Form1 وهو النموذج الذي يحتوي على زر واحد لاظهار النموذج Frm2 ولكي نتاكد من ان النموذج هو النموذج الذي نبدا به في البرنامج ولكي نتاكد من ذلك قم بفتح القائمة Project ثم نضغط على الخيار الاخير الذي يحتوي على كلمة property كما هو موضح في الشكل الاتي



فتظهر لك النافذة الاتي فتقوم بتعديل لكي تكون النافذة الاولى هي التي تبدا بالبرنامج ونعلم ان اسم النموذج الاول هو Form1



سنقوم بتشغيل البرنامج ونتبع الخطوات الاتية

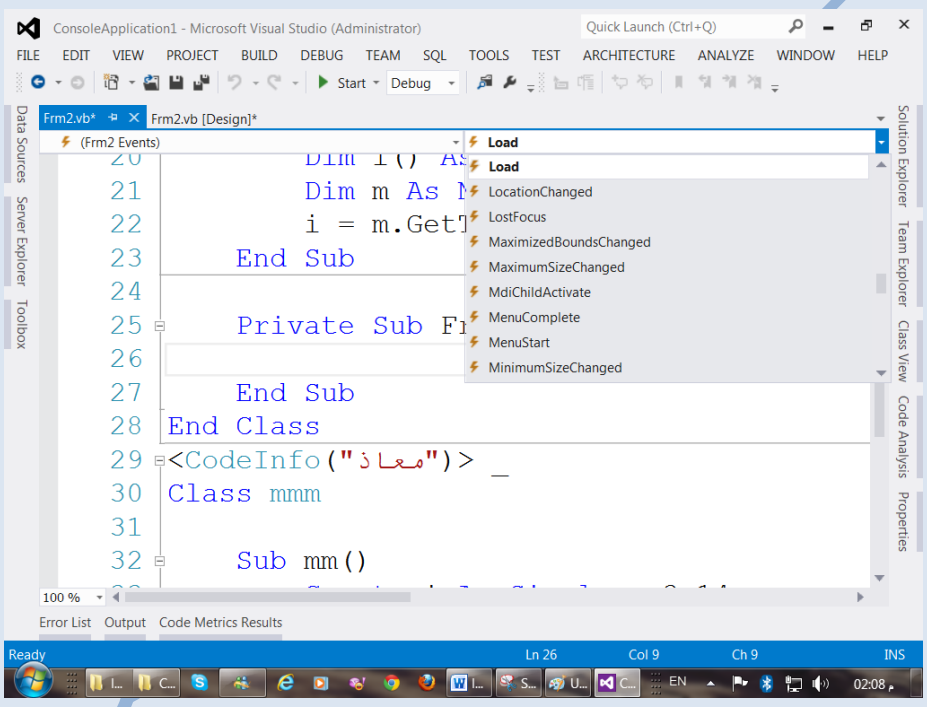
1- سنشغل البرنامج ونضغط فيظهر النموذج Form1 على الزر فيظهر لك النموذج الثاني Frm2
2- اضغط على الزر الاول فتلاحظ تغير الخصائص النموذج التي كتبناها
3- اذا ضغطنا على الزر الثاني الذي به الامر hide سيختفي النموذج الثاني وظل النموذج الاول Form1ظاهر معك فنضغط على الزر فيظهر النموذج الثانيFrm2 محتفظ بخصائصه

4- اذا ضغطنا على الزر الثالث الذي به الامر Close سيختفي النموذج الثاني Frm2 وظل النموذج الاول ظاهر معك فعند الضغط على الزر فيظهر النموذج الثاني Frim2 بدون الخصائص المعدلة ويكون كما كان في الاول بحيث انه خسر خصائصه المعدلة

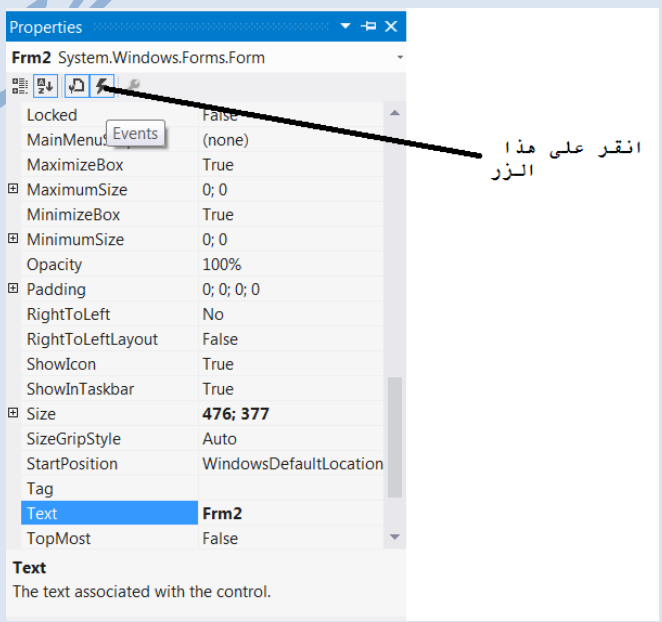
بهذا نكون قد عرفنا ما هو الفرق بين الدالة hide و Close

الاحداث في النموذج

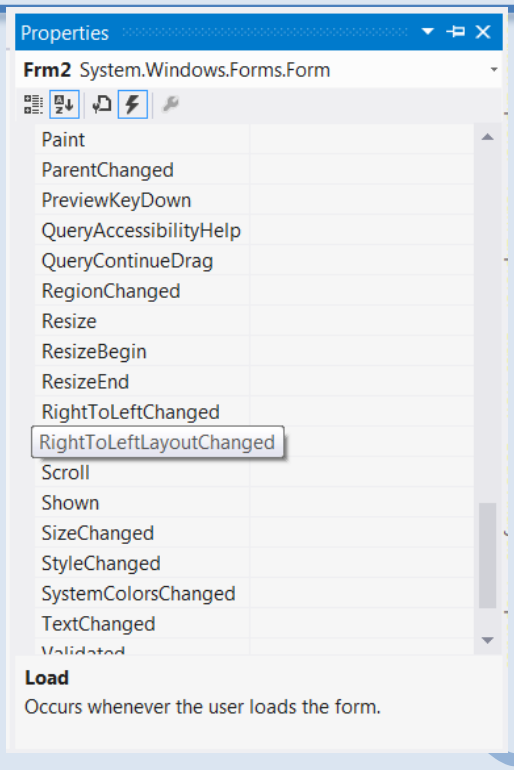
سوف نستعرض اهم الاحداث التي نستخدمها مع النموذج ولكي تربط الاحداث بالاجراءات لديك طريقتين الطريقة الاولى هي الطريقة التي كنا نتسخدمها من نافذة الاكواد ونختار الاحداث التي نريدها كما هو في الشكل الاتي



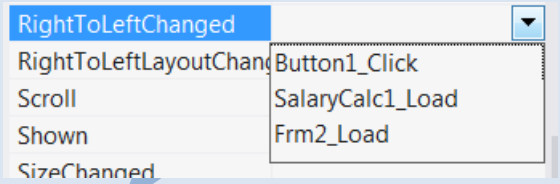
الطريقة الثانية هي من نافذة الخصائص



اذا نقرنا على الزر الخاص بالحادث تظهر قائمة الاحداث كما في الشكل الاتي



اذا اردت اربط الحدث باي اجراء قم النقر على الزر فيظهر كل الزر السهم فيتدلى لك كل الاجراءات التي كتبتها داخل الفورم على الشكل الاتي فاختر منها واحدة



واذا اردت ان تنشئ اجراء مرتبط بالحدث مثل الطريقة الاولى انقر مرتين على الحدث فتظهر لك شاشة الكود منشئة الاجراء مثل Form1_Click مثل الطريقة الاولى لاختيار الحدث

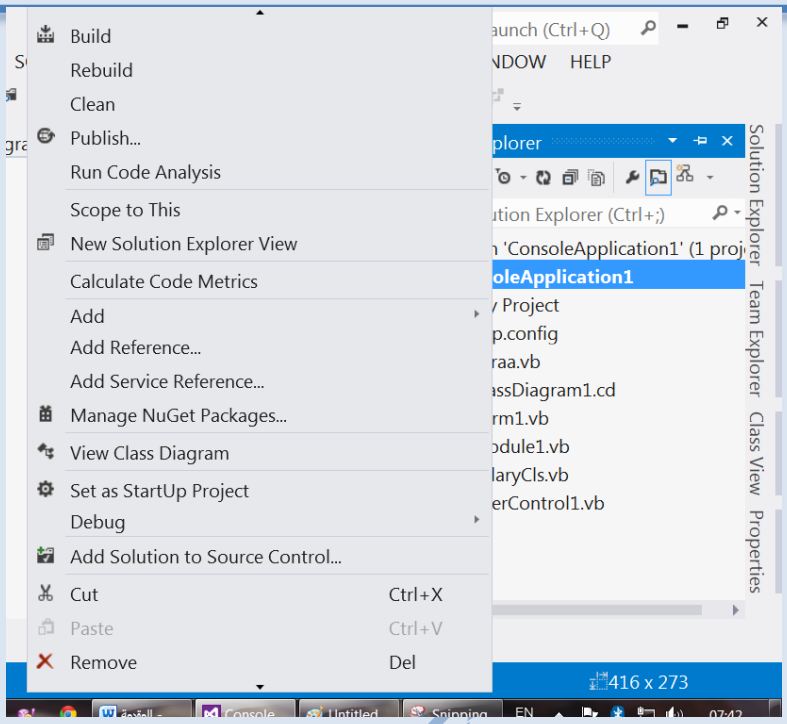
واليك الان جدول يوضح اهم الاحداث من تصنيف النموذج التي نستخدمها في تطبيقاتنا

Load	عندما يحمل النموذج باستخدام الاجراء Show فانك تنشط هذا الحدث فيتم استدعاء الاجراء الذي نريده بتنفيذ الاوامر التي نريدها اثناء تحميل النموذج
Click	ينشط هذا الحدث عندما ينقر بالماوس على النموذج
MouseMove	ينشط الحدث عنما يتحرك الماوس على النموذج
MouseDown	عندما نقر المستخدم على النموذج ويكون الزر في حالة الضغط
MouseUp	عندما نقر المستخدم على النموذج ويكون الزر في حالة

	الرفع اليد عن الزر
MoseHover	عندما يمر الماوس على النموذج ويتوقف لمدة نصف ثانية
MouseLeave	عندما يخرج الماوس من النموذج
MouseEnter	عندما يدخل الماوس على نموذج
Activated	عندما ينشط النموذج
Deactivate	عندما لا ينشط النموذج ويتم اختيار نموذج اخر
Resize	عندما نغير حجم النموذج
Paint	هذا الحدث مهم جدا في عملية الرسم بحيث يتم من خلاله رسم النموذج
BackColorChange	عند تغيير لون الخلفية
BackgroundImageChange	عندما نغيرالخلفية وتصبح صورة
FontChange	عندما نغير شكل الخط
ForeColorChange	عندما نغير لون الخط
RightToLeftLayoutChanged	عندما نغير الخاصية RightToLeft
TextChanged	عندما يحدث تغيير في الخاصيةtext
KeyDown	عند الضغط بالوحة المفاتيح ويكون وضع الزر مضغوط
KeyUp	عند الضغط بللوحة المفاتيح ويكون وضع الزر بعد الضغط
keyPress	عندما ينتهي النقر على الزر

القائمة المساعدات المختصرة

كنت قد اخبرتك انني سوف اشرح القائمة المختصرة التي تظهر عند الضغط على الزر اليمين عند الوفوف على المشروع في نافذة المتصفح .... دعني اذكرك بشكلها



هذه القائمة المختصرة ... هذها لقائمة من الفيجوال الستوديو 2012 اما القائمة في الاصدارات السابقة ستختلف عنها بعض الشيء وسوف ساستعرض عناصر القائمة في الجدول الاتي

Build	لبناء المشروع بحيث تقوم بنقل البيانات الى المسار الاتي Bin/Debug و كنت شرحت هذه المسار في الفصل السابق في مبادء البرمجة الموجهة في مبدا الكبسلة
Rebuild	اعادة بناء المشروع بحيث انك لو قمت ببناء المشروع ووجدت ان هناك ادوات لم تصل فانك تعيد بناءه حتى يتم نقل كل الادوات للمسار
Clean	مسح كل الملفات في bin/Debug
Publish	يقوم باعداد المشروع بحيث تضعه في ملف مضغوط ويتم فكه عن طريق عمل Setup
View Class Designe	هذه الاداه هي واحدة من افضل الادوات التي تساعد في تحليل الكود بحيث يظهر التصنيف و الواجهات الى رسمات بيانية تساعدك في تحليل الكود الذي كتبته والوصول اليه
Add	هنا يندرج منها قائمة فرعية تعطيك ما الذي تريد اضافته للمشروع فيظهر لك عدة خيارات وهي *New item* عنصر جديد بحيث يظهر لك صندوق حوار يعطيك عدة خيارات لكي تضيفها الى المشروع *Existing item* اي عنصر من مشروع اخر *New Folder* عمل مجلد جديد في المشروع *Class* اضافة تصنيف

	Windos Form نموذج جديد *User Control* اداة تحكم وسوف نشرحها بالتفصيل في الفصل القادم *Combonant* اداة تحكم اخرى وسوف نشرحها بالتفصيل الفصل بعد القادم *Module* الموديول
Add Refrence	هي اصافة مكتبات الDll للمشروع وكنا شرحنها سابقا
Debug	هي تشغيل البرنامج F5
Cut	قص الملف
Rename	اعادة التسمية
Remove	حذف المشروع
Properties	الخصائص

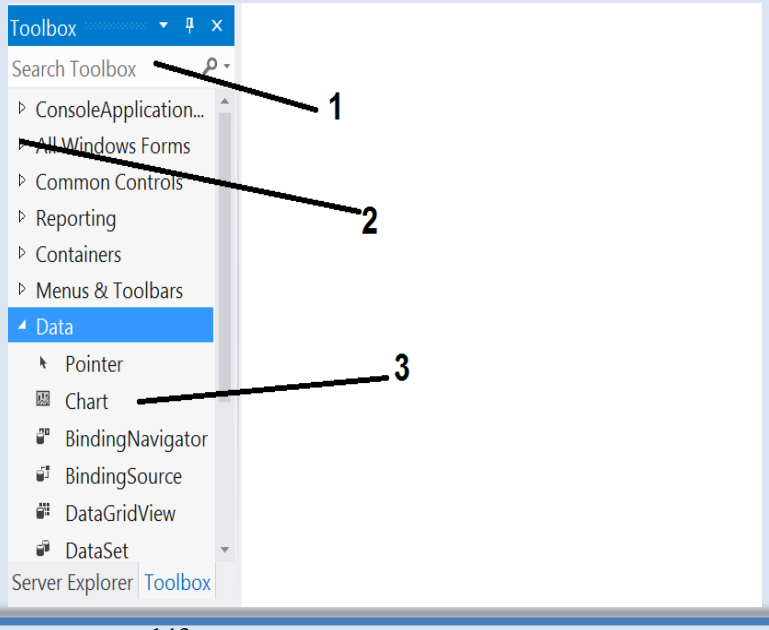
هذه اهم ما نحتاجه اثناء عملنا وهذه الادوات موجودة في كل الاصدارات للفيجوال ستوديو

ادوات التحكم

سننتقل الى التعرف على الادوات التي تكون تطبيقات الويندوز وقبل ان ننتقل الى دراسة الادوات يجب ان نتوقف قليلا عند صندوق الادوات ToolBox

صندوق الادوات او الToolBox هو عبارة مثل حقيبة الدوات النجار او الكهربائي فهي تساعد المطور في استخدام الادوات في تطوير تطبيقاته وسوف نستعرض اهم ما في الصندوق

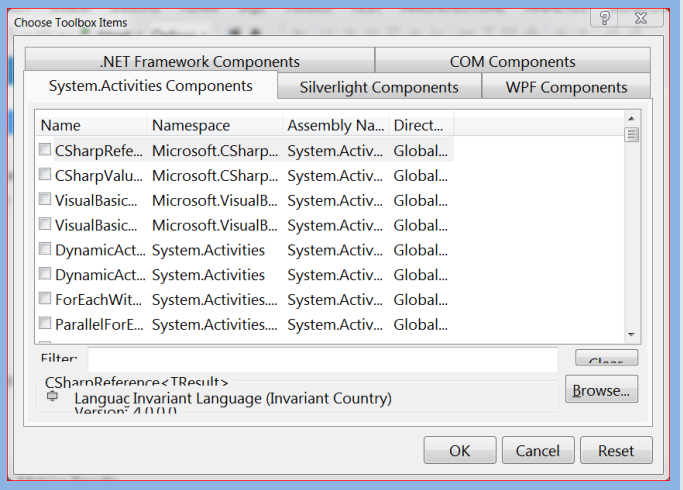
انظر معي الشكل الاتي



في الشكل السابق هذه صورة لصندوق الادوات من الفيجوال ستوديو 2012 وهي تختلف بعض الشيء عن الصناديق السابقة وقمنا بترقيم افسامها الى 3 اقسام

1- هذه الخاصية التي لن تجدها في الاصدارات السابقة في فيجوال ستوديو 2010 و 2008 وهي تقوم ببحث عن الادوات و تسهل عليك الوصول الى الادوات التي تريدها عند كتابة اسمها فقط
2- هذه خاصية تسمى Tape وهي تقوم بتجميع الادوات المتشابهة في مجموعة واحدة وتعطيها اسم معين لتسهيل الوصول اليها وهذه الخاصية متوفر في جميع الاصدارات
3- الادوات التي نضعها على النموذج

لو نقرت بالماوس على صندوق الادوت بالزر الايمن ستندرج لك قائمة مختصر هذه القئمة تتكون من العناصر الاتية

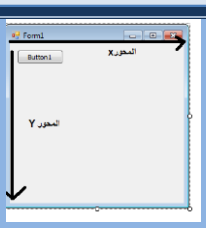
View list	يظهر الدوت بشكل تتابعي الواحدة فوق الاخرى على شكل قائمة اذا كانت الاختيار بها علامة صح واذا لم تظهر العلامة ستظهر بجانب بعضها
Show All	اظهار كل الادوات
Choose items	ادخال ادوات جديدة الى صندوق الادوات بحيث انك اذا اخترتها تظهر كل نافذة كالاتي  اختر الاداة التي تريدها ثم انقر ok فتظهر على صندوق الادوات ستجد بها عدة السن (Tabs) او اقسام على اساس نوع الاكواد *System Activities : هي فضاءات الاسماء التي نستخدمها في برامجنا* *NetFramWark : الادوات المتطورة باستخدام الnetframewark* *Com هي الادوات المتطورة من لغات اخرى كفيجوال بيسك 6 و C++*

	Silverlight هي ادوات مطورة تطبيقات *Silverlight* *Wpf* تطوير ادوات من نوع *Wpf* اما عن الزر المكتوب عليه كلمة Browse هي تتيح لك فرصة ان تاخذ ادوات من خارج المشروع هذه النافذة موجوده بهذا الشكل وهذا التقسيم فقط في الفيجوال استوديو 2012 ولكن النوافذ في الاصدارات الاخرى فهي شبيهة بها ولكن بترتيب اخر
Sort item Alphabitically	ترتيب الادوات ترتيب ابجدي
Restart ToolBox	اعداة صندوق الادوات على ما هي عليه بعد ننصيب البرنامج

سوف نتناول الان اهم الادوات التي نستخدمها في تطوير تطبيقاتنا

قبل ذلك سوف نستعرض اهم الخصائص المشتركة في الادوات لان كل الادوات مشتقة من تصنيف Control الموجود في فضاء الاسماء Forms واليك الخصائص التي في هذا التصنيف

Name	اسم الاداة التي نستدعيها في وقت البرمجة ويتم تسمية الاداه بنفس الشروط التي شرحناها في شروط المتغيرات
Size	تحدد حجم الاده وبها خاصتين الطول و العرض التي تحدد طول و عرض الاداه
Location	تحدد مكان الاداة على الاداة التي تحتضنها مثل Form , GroupBox , Panal بحيث انها تحدد مكان الاداة عن طريق خاصيتين هما X,Y بحيث X تحدد يسار الاداة و الخاصية y التي تحدد اعلى الاداه كما توجد الخاصيتين *left , Top* التان تحددان يسار و اعلى الاداة ..... احب ان اوضح ان هذه الخصائص من نوع integer بحيث ان لو زادت قيمة الخاصية left فان الاداة تتحرك من اليسار الى اليمين واذا زادت قيمة Top فان الاداة تتحرك من الاعلى الى الاسفل اريد ان اوضح لك شيء هاااام جدا ان النقطة التي تقع على يسار النموذج او GroupBox دائما تكون النقطة (0,0) بحيث انك لو وضع الاداة اقصى يسار النموذج فان قيمة الخاصية left او الخاصية X تساوي 0 وكذلك في الخاصيتين Top ,y كما هو موضح في الشكل الاتي

	
Dock	في هذه الخاصية بها عدة خيارات بحيث تتيح لك اختيارات تحدد من خلالها موضع الاداه بالنسبة للنموذج او اي اداة محتضنة للادوات مثل GroupBox الخيارات هي *Bottom* تقع الاداة في اسفل النموذج *Left* تقع الاداة في يسار النموذج *Reight* تقع الاداة في يمين النموذج *Top* تقع الاداة في اعلى النموذج *Fill* تقع الاداة في كل النموذج *none* لا يوجد
Anchor	هذه الخاصية رائعة بحيث تربط موقع الاداة بالنسبة الاداة الحاضنة للادوات ولكن احب ان تجرب هذه الاداه بنفسك لاني لو قمت بشرحها فلن تصل المعلومة كما يجب لذا ساعطيك تجربة قم بفتح مشروع جديد وضع عدة ازرار في شتى اماكن النوذج ثم قم تعديل الخاصية لاحد الازرار بحيث تكون قيمتها Right ثم قم بتغيير حجم النوذج بحيث تغير عرض النوذج ستلاحظ ان الزر يتحرك لليمين ولكن اذا غيرت قيمة الخاصية Anchor الى Left , Reight فان حجمه من العرض هو الذي يتغير ونفس الكلام في قيمتين Top , Botton
BackColor	تغير لون الاداة
BackgroundImage	تغير خلفية الادة بحيث تضع صورة
BackgroundImageLayout	تحديد موقع الصورة في الخلفية
Enable	هل الاداة نشطة ويمكن استخدامها
Visable	هل الاداة ظاهرة على النموذج
Text	هذه الخاصية مثل الخاصية Text في النموذج بحيث تتيح للمستخدم اسم الاداة التي يتعامل معها مع العلم انها من نوع String يعني انك تكتب بها كما تشاء ليس بها شروط التسمية وهي شبية الخاصية Caption

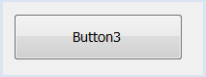
	الموجودة في فيجوال بيسك 6
Font	تعدل شكل الخط على الاداة
ForeColor	تحدد لون الخط على الاداة

هذه اهم الخصائص التي استعرضناها وسوف نستعرض اهم الاحداث المشتركة

Load	عندما يحمل الاداة مع تحميل النموذج الذي هي تقع فيه
Click	ينشط هذا الحدث عندما ينقر بالماوس على الاداة
DoubleClick	عندما نقر المستخدم مرتين
MouseMove	ينشط الحدث عنما يتحرك الماوس على الاداة
MouseDown	عندما نقر المستخدم على الاداة ويكون الزر في حالة الضغط
MouseUp	عندما نقر المستخدم على الاداة ويكون الزر في حالة الرفع عن الزر
MoseHover	عندما يمر الماوس على الاداة ويتوقف لمدة نصف ثانية
MouseLeave	عندما يخرج الماوس من الاداة
MouseEnter	عندما يدخل الماوس على الاداة
Size Change	عندما نغير حجم النموذج
Paint	هذا الحدث مهم جدا في عملية الرسم بحيث يتم من خلاله رسم النموذج
BackColorChange	عند تغيير لون الخلفية
BackgroundImageChange	عندما نغيرالخلفية وتصبح صورة
FontChange	عندما نغير شكل الخط
ForeColorChange	عندما نغير لون الخط
TextChanged	عندما يحدث تغيير في الخاصيةtext
KeyDown	عند الضغط بالوحة المفاتيح ويكون وضع الزر مضغوط

KeyUp	عند الضغط بللوحة المفاتيح ويكون وضع الزر بعد الضغط
keyPress	عندما ينتهي النقر على الزر
VisableChange	عندما تغير الخاصية Visable
EnableChage	عندما يغير الخاصية Enable

سوف نستعرض اهم الادوات واهم خصائصها



Button

هذه الاداه كانت تسمى Command في الفيجوال بيسك 6 وهي عندما يقوم المستخدم بالضغط عليها فتقوم بتنفيذ الامر وهذه الاداه مشهورة جدا في تطبيقات النوافذ وكنا قد استخدمناها في امثلة سابقة وسوف نستعرض لك اهم الخصائص التي تختلف عن باقي الادوات لانها مشتقة من التصنيف Control ولكن تم اضافة بعض الخصائص كما هو موضح في الجدول الاتي

FlatStayle	هذه الخاصية تحدد مظهر الزر
TextAlign	هذه الخاصية احدد مكان النص على ازر هل يقع يمين الزر ام يساره ام في الوسط
cursor	يحدد شكل الماوس عند المرور على الزر

ستجد الامر سهل جدا

TextBox

هذه الاداة تتيح للمستخدم ان يكتب على الاداة بحيث انه يقوم بادخال البيانات المطلوبة للتعامل معها مثل انه يدخل الاسم و العمر فهي تشبه الدالة Readlineالتي في الكونسل مع العلم الخاصية Text هي التي تستقبل البيانات التي يدخلها المستخدم واليك اهم الخصائص التي تميز الاداة عن غيرها

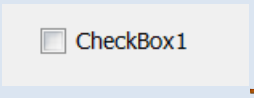
PassWordChar	هذه الخاصية تحدد شكل الحرف التي تخفي بيانات كلمة السر يعني ان اذا كانت قيمة الخاصية هي * فان المستخدم اذا كتب على الاداة كلمة السر مثال بسم الله فانها تستبكل كل حرف بهذه القيمة فتظهر على صندوق النص بهذا الشكل ********

TextAlign	هذه الخاصية احدد مكان النص على ازر هل يقع يمين الزر ام يساره ام في الوسط
cursor	يحدد شكل الماوس عند المرور على الزر
ReadOnly	هذه الخاصية تمنع المتسخدم الكتابة او التعديل على الاداة
Multiline	هذه الخاصية تتيح للاداة بتعدد الاسطر بحيث انك تكتب عدة اسطر في الاداة
ScrollBars	اذا كانت الاداة بها عدة استطر فانك تحتاج الى هذه الاداة بحيث تضع لك شريط التمرير الذي يساعدك على التحرك داخل الادة
BorderStyle	هل يكون شكل صندوق انص ثلاثي الابعاد ام مستوى وغيرها من الاشكال

Lable

هذه الاداة تظهر فقط نصوص توجيه بحيث توجه المستخدم كيف يتعامل مع التطبيق مثل انكت تطلب من المتسخدم ادخال الاسم و رقم البطاقة فتضع صندوق العنوان يقوم بتوجيه المسخدم بوضع الاسم و رقم البطاقة ومكان الذي يضع فيها

ستجد ان كل الخصائص التي في هذه الاداة قد شرحنها

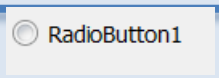


CheckBox

هذه الاداة تتيح للمتسخدم ان يضع علامة صح على الاداة بحيث يعطي للكمبيوتر ان المتسخدم قد اختار هذا الاختيار بحيث تعطي للمستخدم عدة خيارات و المستخدم يختار منها ما يريد فيضع علامة صح بجانب على الاداة ... واليك اهم الخصائص

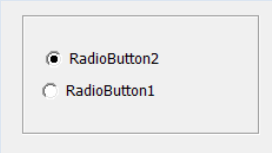
Appearance	تحدد مظهر الاداة بحيث هل يكون شكل هى على شكل زر ام على شكله العادي
Checked	هل تم اختيار هذه الاداة ام لا يعنى هل وضعت علامة صح على الاداة ام لا
CheckState	تحدد وضع الاداة عن تحميل النموذج او تشغيل البرنامج هل تكون معلمه ام لا
TextAligne	تحدد مكان النص على الاداة هل يمين الاداة ام يسارها ام اعلى يمينها ام العى يسارها ام وسطها ام اعلى وسطها وغير ذلك

واهم الاحداث هو CheckChanged ينشط عندما تغير الخاصية Checked

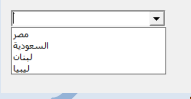


RedioBox

هذه الاداه شديدة الشبه الاداة CheckBox ولكن تختلف انه يجب ان يتم اختيار اداة واحدة من عدة ادوات من نفس النوع وموجودة في نفس الادة الحاوي كما هو موضح في الشكل الاتي

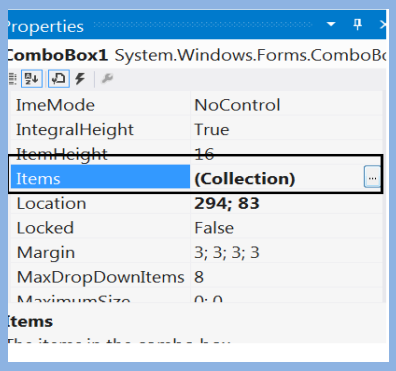
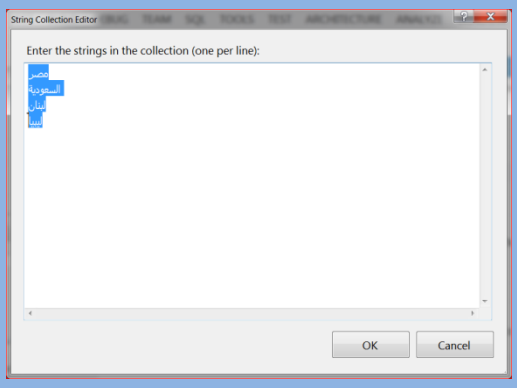


ولها نفس الخصائص التي في CheckBox



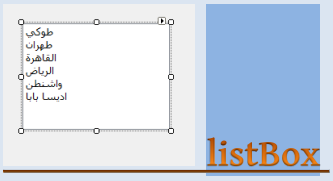
ComboBox

هذه االاداة تتيح لك اختيار واحد من عدة خيارات متاحة من داخل الاداة كما هو موضح في الصورة السابقة وهذه الاداة بها خصائص تختلف عن باقي الادوات وهي

items	هذه الاداة تتيح لك ادخال العناصر فلو انك تريد ادخال اي عنصر اذهب للخاصية itemsثم اضغط على زر كما هو موضح في الشكل الاتي  فتظهر لك النافذة ادخل بها العناصر التي تريدها 
DropDownStayle	هذه الخاصية تتحكم في طريقة عرض العناصر داخل الصندوق بجيث يتم عرضها بشكل منسدل ام بشكل قائمة ثابتة وهذه الخاصية بها عدة خيارات

	Simple تصبح على شكل قائمة ثابتة **DropDown** الشكل التقليدي للانسدال ولكن عند الضغط بالماوس على زر السهم **DropDownList** الشكل التقليدة ولكن عن الضغط على اي مكان في الاداة

SelectedIndexChanged هو من اهم الاحداث وينشط عندما يتغير اختيرا العنصر كما يوجد احداث مرتبطة بعندما تنزل القائمة لعض الاحداث مثل DrobDown و DargDrop



listBox

هي شبيهة تماما بالComboBox ولكن تختلفي عنه بانها تعرض العناصر بشكل ثابت كما بها بعض الخصائص تختلف عنه بعض الشيء

SelectionMode	في هذه الاداة يمكن اختيار اكثر من عنصر بحيث يمكنتك اختيرالعنصر طوكيو و القاهرة و وشنطن وذلك باستخدمام المفاتحين Shift لتحديد العناصرا لمتتابعة و Ctrl لتعديد عدة عناصر ولكن يجب ان تحدد ذلك بعدة خيارات وهي One اختيار واجد MultiSimple عدة خيارات MuliExtended عدة خيارات ولكن يجب ان تكون متتابعة
Items	هي الخاصية التي تدخال الادوات داخل القائمة
Sorted	هذه الخاصية توجد ايضا في الComboBox وتقوم بترتيب العناصرترتيب ابجدي
MutliColumn	هل يمكن عمل عدة اعمدة
ColumnWidth	ما هو عرض العمود الواحد

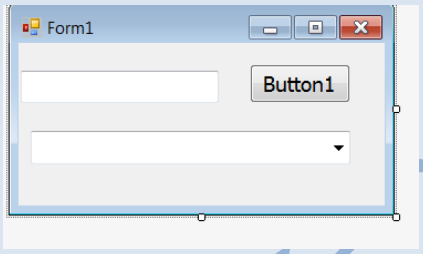
كما ان هذه الاداة بها نفس الاحداث التي في الاداة ComboBox مع اختلاف الحدث ItemChange وهي عندما يتغير العنصر الذي نختاره

مثال على ما سبق

سنقدم مثال تطبيقي على الاداتين ComboBox و listBox بحيث يمكنك ان تقوم باضافة عناصر لهما اثناء وقت التشغيل . قم بفتح مشروع جديد واضف اليه العناصرالاتي

الاداة	الخاصية	القيمة
Textbox	Name	Txt
Combobox	Name	Comb
Button	Name	btn

بحيث يصبح شكل النموذج كلاتي

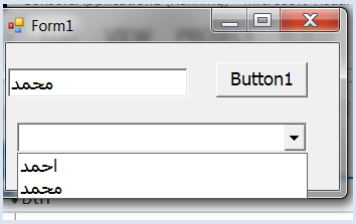


فكرة هذه المثال انك تقوم بكتابة العناصر في الtxt1 وعند الصغط على Btn1 تظهر في الComb

قفم بكتابة الكود في الحدث Click للزر

```
Cmb.Items.Add(txt.Text)
```

وعند تشغيل البرنامج قم بكتابة اي اسم ثم اضغط على الزر Btn ستجد تظهر على الCmb كما هو في الشكل الاتي



كما رايت اننا استخدمنا الخاصية items ليتم اضافة العنصر ولكن يجب ان تعرف ان هذه الخاصية مهمة جدا في تطبيقات النوافذ حيث انك ستجدها في كثير من الادوات التي نسخدمها ليس فقط في الاداة ComboBox ولكن في ادوات اخرى تحمل عدة عناصر مثل listBox , DatagrideView وTreeView وغيرها

ولكن هذه الخاصية من نوع ObjectCollection بحيث انها تعمل على تجميع العناصر و العناصر هي البيانات التي ندخلها ويجب ان تكون من انواع البيانات يعني يجب ان تكون من نوع رقمي او حرفي او Boolen او تاريخي ولكي تتعرف اكثر عن هذه الخاصية فهي تحتوي على عدة دوال مهمة

Add	هذه الدالة تقوم باضافة العنصر داخال بحيث تحتوى على معامل واحد يقوم بادخال اعنصر داخل الخاصية او داخل المجموعة فان صيغة الدالة

	تكون كلاتي Add(item as object) بحيث item معامل من نوع object يدخل البيانات
AddRange	هذه الدالة تقوم بادخال العناصر دفعة واحدة لذا تجد هنا معامل عبارة عن مصفوفة تدخل البيانات كلها دفعة واحدة فتجد ان الصيغة هي Add(item() as object)
Remove	هذه الدالة تقوم بالبحث عن العنصر الذي نريده من المجموعة ثم تحذفة فتخيل ان المجموعة بها اسماء البلاد ونريد ان نحذف كلمة الولايات المتحدة منها فاننا نكتب في المعامل الكلمة وتكون الصيغة كلاتي Remve(item as object)
RemoveAt	هذه الدالة تقوم بحذف العنصر الذي نحدد رقمه من الخلال المعامل
Insert	هذه الدالة تحدد مكان العنصر الذي نريد تعديله ثم نعدله فتجد صيغتها كلاتي insert(index as integer,item as object) حيث ان index هو رقم العنصر و item هي القيمة التي نريد تعديلها للعنصر
IndexOf	هذه الدالة رقمية بحيث تبحث عن العنصر الذي نريد ان نصل اليه فتعطينا رقم العنصر
Count	هذه خاصية وليست دالة تحدد عدد العناصر في الخاصية items

ولنفترض انك تريد تحديد رقم العنصر الذي قمت باختياره من الاداة comboBox فنك لا تستطيع تحديده في وقت التصميم يجب ان تحدده في وقت التشغيل لذا سوف نستعرض بعض العناصر التي لا تظهر على نافذة الخصائص ولكن نجدا اثناء كتابة الكود وتظهر نتائجها فقط في وقت تشغل البرنامج

SelectedIndex	تحدد رقم العنصر الذي اخترناه
SelectedValue	تحدد قيمة العنصر الذي اخترناه

GroupBox

هذه الاداة التي تعتبر من الادوات الحاضنة للادوات الاخرى بحيث تقوم باحتواء ادوات تحكم فهي تشبه بشكل كبير النموذج لذا ستجد خصائص كثيرة بها تشبه الخصائص التي في النموذج ولا يوجد خاصية تجعلها تختلف عن باقي الادوات وتوجد اداه اخرى تشبهها هي Panal

PicturBox

هذه الاداة هي تقوم بعمل رسم للصورة التي تحتويها بحيث ان هذه الاداة مشهورة في التعامل مع الصور بحيث تحدد اللون و العرض كما يمكنك انقوم بالتعديل بها واهم خاصية بها هي الخاصية image التي تستبقل الصور

Timer

هذه الاداة تختلف بشكل كبير عن العناصر الاخرى بحيث انها لا يوجد لها مكان دخال النموذج بحيث انها تعمل وهي غير موجودة لن تجد بها خصائص كثيرة مثل التي في الادوات التي شرحنها ...هذه الاداة تسمى المؤقت بحيث انها تحدد وقت معين فاذا تم هذا الوقت فانها تفعل حدث يسمى Tick فيتم استدعاء الاجراء الذي يجب ان ينفذ

مثال

افتح نموذج جديد واضف اليه صندوق عنوان label واضف اليها timer

اكتب الان الكود الاتي في الحدث Tick

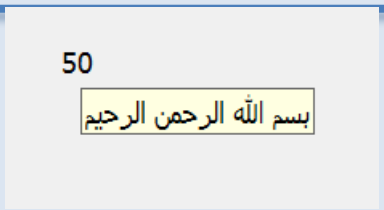
```
Static i As Integer
i += 1
Label1.Text = i.ToString
```

وعند تشغيل البرنامج ستلاحظ ان العداد يقوم بالعد وضيف رقم جديد ولكن يجب ان تتعرف على خصائص هذه الاداة لانها مهمة جدا

Interval	خاصية رقمية تعتبر بمثابة عداد يقوم بعد من 0 الى الرقم المحدد داخل الخاصية وليكن 100 فاذا وصل الى الرقم 100 فانه يفعل الحدث tick ويعند الرقم مرة اخرى الى 0 ليعيد العد مرة اخرى مع العمل ان الثانية الواحدة تساوي الرقم 1000 في هذه الخاصية يعني اذا كانت القيمة تساوي 1000 وقام العداد ووصل العد الى 1000 فانه استغرق ثانية في العد
Enable	تحدد هل العداد مفعل ام لا

ToolTip

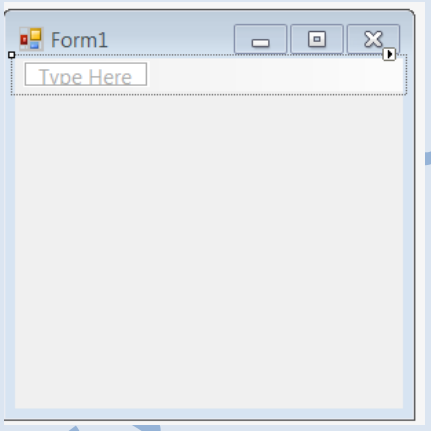
هذه الاداة ايضا تختفي ولا تظهر في النموذج ولكن هذه الاداة مهمتها هي ان تعطي تلميح للمتسخدم عندما يقم الماوس على اي اداة لتستطيع فهمي اكثر قم موضع اداة tooltip على النموذج في نفس المشروع الذي كنا نشرحه في الاداة Timer ثم اكتب في الخاصية ToolTip on ToolTip التي تخص label وهذه الخاصية لا تظهر الا اذا استخدمت هذه الاداة فكتب في "بسم الله الرحمن الرحيم " وعند التشغيل قف على صندوق العنوان ستجدها تظهر كلاتي

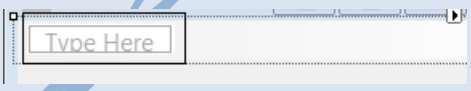


القوائم menus

اذا كنت تستخدم تطبيقات الاوفس ستجد هناك شريط يسمى menus bar وتجد فيه قائمة File و Edite وغير ذلك وسوف نستعرض الان كيف تنشئ قوائم مشابهة للقوائم في تطبيقات الافس

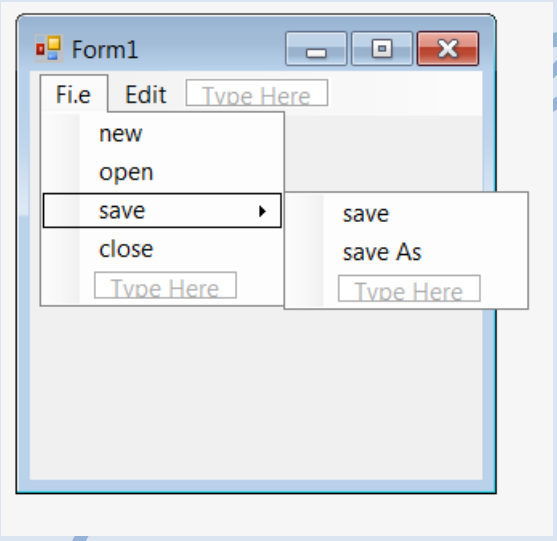
افتح مشروع نوافذ ثم اختر من صندوق الادوات ToolBox الاداة MenuStrip وستجد انه تظهر على النوافذ كلاتي



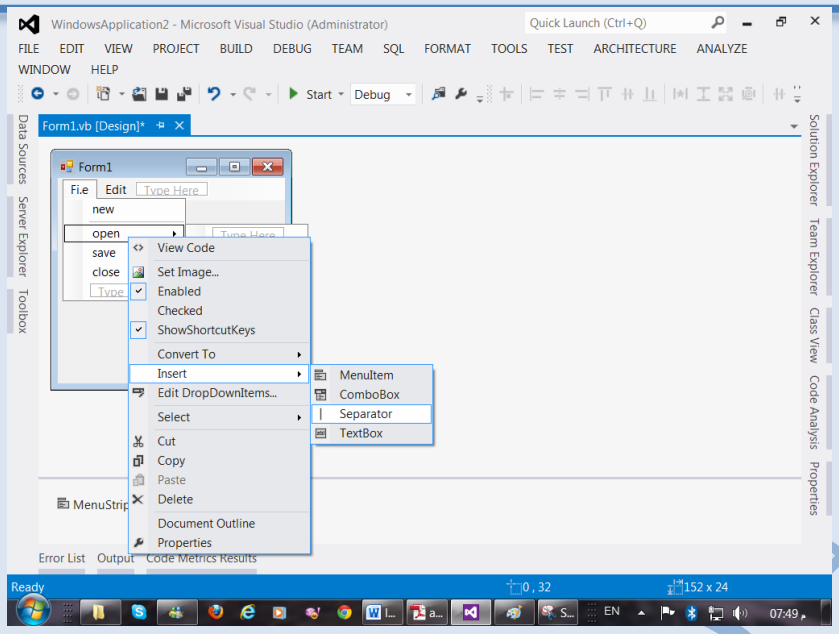
ثم اكتب على المربع النص الموجود في القائمة  بحيث ان هذا مربع سيحمل اسم القائمة دعنا نكتب كلمة File

ثم نكتب عناصر القائمة كما هو موضح في الشكل الاتي

اكتب بعد ذلك باقي عناصر القائمة بالشكل الذي تريده انا ساكمل قائمتي حتى تصبح على الشكل الاتي

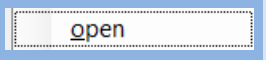


يمكنك عمل فاصل بين الامرين وذلك باختيار Separator من القائمة الفرعية insert كما هو موضح في الشكل الاتي



واذا اردت حذف اي عنصر انقر عليه ثم تغط على زر حذف سيتم حذفها

واهم الخصائص التي تميز القوائم هي

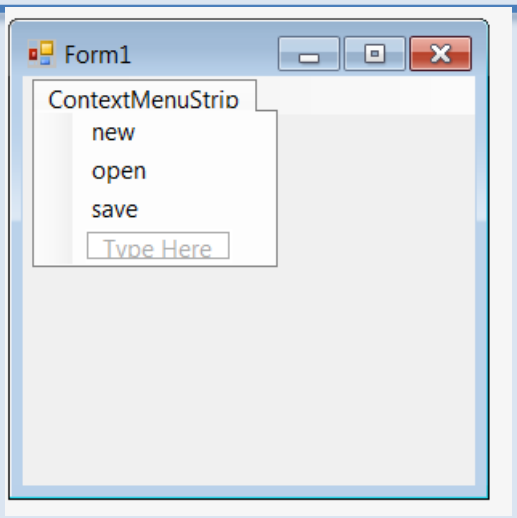
Text	هو يكتب اسم العنصر واذا اردت وضغ اختصار للاسم قم بوضع حرف & قم الكلمة مثل &File فتظهر على النافذ على الشكل الاتي 
Check	هو ان تضع علامة صح عن اختيار العنصر
ShortCutKeys	تضع مفاتيح اختصار للقائمة مثل Ctrl O لاختيار العنصر open
ShowshortCutKeys	يظهر مفاتيخ الاختصار بجانب العنصر في القائمة

واذا اردت ادخال اي امر بداخل اي عنصر انقر عليه مرتين ثم اكتب الكود الذي تريده

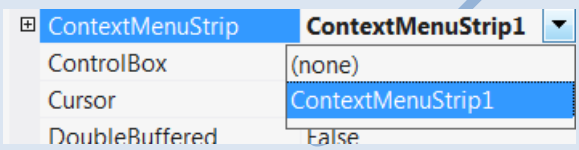
ContextMenuStrip القائمة المختصرة

هذه الاداة تشبه تماما الاداة menueStrip ولكن لا تظهر الا عند النقر بالزر الايمن للماوس على اي اداة .

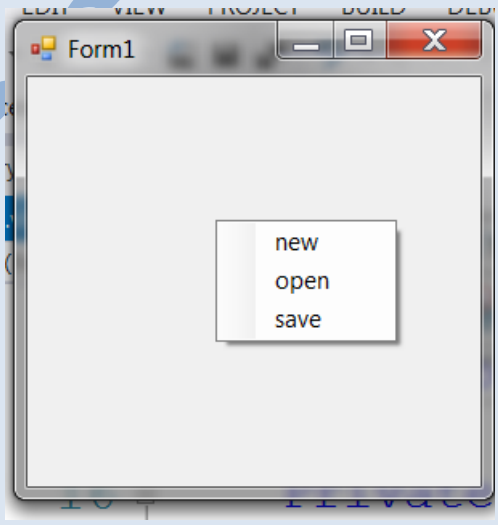
واذا اردت عمل ذلك انقر ContextMenuStrip من صندوق الادوات ثم ادخل العناصر لكي تظهر على الشكل الاتي



واذا اردت ان تظهر هذه الاداة عندما ننقر بالزر الايمن على الماوس قم بفتح نافذة الخصائص للنموذج واختر الخاصية ContextMenuStrip التي تخص النموذج ولا تظهر الا عند استخدام الاداة ContextMenuStrip واختر القائمة كما هو موضح في الشكل الاتي



واذا شغلت البرنامج ونقرت بالزر اليمين تظهر القائمة كما الشكل الاتي

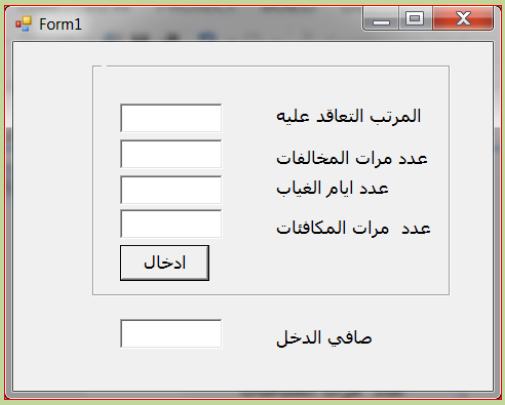


وبذلك نقوم قد استعرضنا هم الادوات التي تستخدم بشكل كبير في تطبيقات النوافذ سنتقل الان الفصل هو من اهم الفصول في هذا الكتاب وهو تصميم ادوات التحكم

الفصل السابع

تصميم ادوات التحكم UserControl

بعد ان تناولنا اهم الادوات التي نستخدمها في تطبيقات النوافذ تعالى معي نتوقف ونسرح بخيالينا تخيل انك الان تعلم مبرمج في احدى الشركات وكنت تقوم ببرمجة برنامج خاص بالمرتبات وقمت بعمل التصنيف الذي شرحنا في الباب الاول وهو SalaryCls في الفصل الخاص بالتصنيفات وتريد ان تدخله في تطبيقات النوافذ قمت بتصميم النافذة كالاتي



وقمت ببرمجتها لاظهار صافي الراتب ثم قمت بعمل هذه النافذة في قسم المبيعات و الحسابات والمشتريات لحساب مرتبات الخاصة بكل ادارة لوحدها **فانك ستقول سوف انسخ الاكواد من كل قسم واضعه في القسم الاخر ونسخ النافذة من واصمم نافذ لكل قسم بحيث تشبه هذه النافذة** فاني ابشرك بضياع وقت طوييييل وخاصة لو انك لا تعمل في برمجة المرتبات فقط فقد فتخيل انك تعمل على ببرمجة المبيعات و المشتريات فتجد انك تجد تشابه كبير في شكل النوافذ و الاكواد فقوم بعمل نسخ ولصق للاكواد ثم تكتشف انك ارتكبت خطا برمجي فقوم باصلاح الكود في كل قسم على حدة بل قد تنسى اين وضعت الكود فتضيع وقت في البحث عن الكود ويضيع منك وقت ثمين يمكنك استغلاله في جعل برنامج قوي ويعمل بكفائة اكثر من انك تقوم بعمل برنامج يكفي بالغرض في وقت قصير وتضع وقتك في الدعم الفني واصلاح البرنامج و تخسر ثقة العميل

ان هذه القصة هي منقولة عن تجربة وخطا وقعت انا فيه منذ ان بدات حياتي البرمجية لذا انصحك ان تركز معي في هذا الفصل ففي هذا الفصل سوف اشرح لك كيف تحل هذه المشكلة وهي تصميم اداة التحكم UserControl

لكي تكون مبرمج محترف وتتمكن العمل ضمن فريق عمل يجب ان يكون الكود الذي تكتبه واضح و مرتب لذا انصحك بعدة نصائح يجب ان تراعى اثناء كتابة الاكواد وخاصة في المشاريع العملاقة و التي قد تنسى ما كتبه من الاكواد او تسلم جزء من عملك لزميلك في العمل فيسهل عليه فهم ما كتبته من اوامر فيخرج منتج متقن وينال اعجاب العملاء لذى عليك مراعات الاتي

- **حدد ما تريده و كتبه مختصر في ورقة**

ان اول ما خلق الله من الخلائق القلم و جعله يكتب مقادير الخلق وهذه رساله لنا من الخالق ان لكي نبدأ اي مشروع يجب كتابته كتابة على ورقة ولو كانت مسودة حتى لا تنسى اي شئ

- **رتب الاكواد**

وهو ان تجعل كل الاكواد المشتركة في تنفيذ امر معين مجمعة في مجموعة وذلك بكتابة كود

#Region"اسم المجموعة"

وهذا سيسهل عليك طرق البحث عن الكود الذي تريد التعديل ولكن ماهو كود region ؟؟ هو كود كما قلنا يجمع عدة اكواد في مجموعة واحدة لكي يسهل عليك الوصول اليها فيكون كلاتي

ملحوظة

هذه الاوامر لا تؤثر على اداء الكونترول في وقت التشغيل ولا حتى عندما تستخدمه في مشروعك ولكن هذه الاوامر ستساعدك على ترتيب اكوداك ليسهل عليك الوصول اليها في اي وقت تريد واذا كنت لا تريد اتسخدامه فاني اذكرك انك ستتعب فيما بعد عندما تتعامل مع مشروع اكبر واكود كثيرة فانك ستحتاج اليها

و الان سوف نستعرض اهم التعليمات التي يجب اتباعها اثناء كتابتك للكود

1) استخدم التعليقات لتساعدك فيما بعد تذكر الكود ولماذا كتبت استخدم المتغير xsum مثلا بدلا من ysum وماهي مهمة الدالة IncSum() مثلا و ماذا وضعتها في هذا المكان ... هذا يسهل عليك تذكر الكود بشكل جيد فلا يستهلك وقت طويل حتى تتذكر الكود الذي كتبته فيسهل علك تطويره او تعديله بالشكل الذي تريد و التعليقات تكتب باستخدام العلامة (') او الكلمة Rem ثم تكتب الكود بعدها مثل هكذا الكود

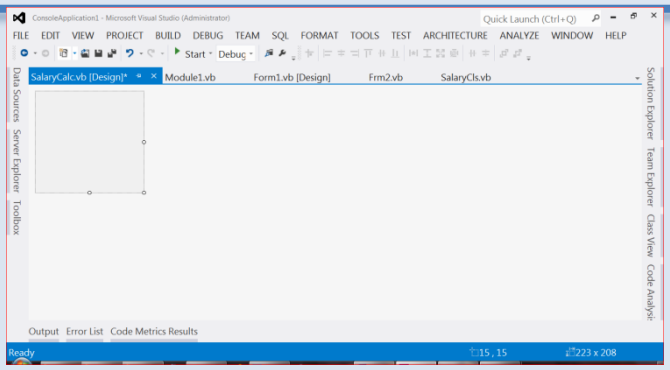
```
Dim Base_x As integer ' هذا الكود من لحساب طول القاعدة
او

Dim Base_y as integer  Rem هذا الكود لحساب عرض القاعدة
```

ويكون لون التعليق اخضر

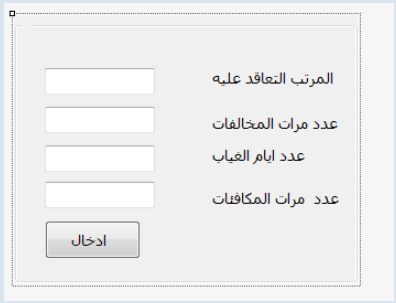
2) استخدم اسماء المتغير واضحة و سهلة حتى يكون ملائم للماهم التي تقوم بها مثل NameTxt و EdetBtn ولا تستخدم اسماء لا معنى لها مثل sdfsdfasd او dsfsdf هذا يعقد عليك الامور
3) استخدم الاحرف الكبيرة و الصغيرة وعلامة _ في تسمية المتغيرات
بحيث يكون الاحرف الكبرة في اول الكلمة مثل NameTxt او يكون _ المسافة الفاصلة بين الكلمتين مثل the_Report_Form

بعد هذه التعليمات نبدا بعمل التصميم الادوات قم باتباع الخطوات الاتية

1. قم باضافة اداة userControl وذلك بفتح قائمة project ← Add user Control وعند ظهور مربع الحوار اكتب اسم الاداة وهي SalaryCalc فتظهر لك الشاشة الاتية



2. قم تصميم بوضع الادوات التي تحسب المرتب على النموذج لكي يظهر على الشكل الاتي



ثم نعدل خصائص الاسماء لصناديق النصوص بحيث تكون كلاتي

صندوق النصوص	الاسم الجديد
TextBox1	ContactSalarytxt
TextBox2	DescountMoretxt
TextBox3	AppcentDaytxt
TextBox4	IncenticeMoretxt

3. نبدا بكتابة الخصائص التي تخص التصاميم فنكتب الخصائص التي تخص الاتي

- خاصية مسئولة عن تعديل الخلفية لصناديق النصوص
- خاصية مسؤولة عن خطوط النصوص
- خاصية مسئولة عن لون النصوص
- خاصية مسئولة لون الخط في صناديق العناوين
- خاصية مسئولة عن خط صناديق العناوين

نفتح نافذة الكود ونكتب الكود الاتي

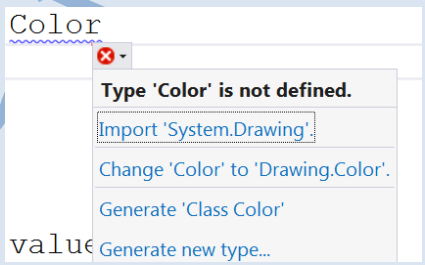
```
#Region "خصائص مسئولة عن تصميم الاداة"
    Private TxtColor As Color
    Public Property TextBox_BackColor As Color
        Get
            Return TxtColor
        End Get
        Set(value As Color)
            TxtColor = value
            ContactSalarytxt.BackColor = value
            DescountMoretxt.BackColor = value
            AppcentDaytxt.BackColor = value
            IncenticeMoretxt.BackColor = value
        End Set
    End Property
```

في الكود السابق تلاحظ اننا فقط كتبنا الخاصية الاولى التي تقوم تغير لون خلفية صناديق النصوص الاربعة فتلاحظ اننا في قسم set المسئول عن اخذ القيم للخاصية قمنا بتعديل خصائص خلفيات صناديق النصوص بحيث انك اذا غيرت لون الخاصية للون الاخضر ستجد ان الصناديق الثلاث تكون خضراء

ملاحظة قد تكتب الكود السابق فتظهر لك علامة خطا وهي الخط الزرق الذي يخرج اسفل الكلمة وتجده يخرج تحت كلمة Corlo كما في الشكل الاتي

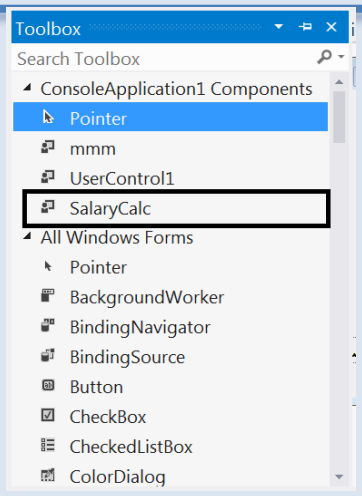
```
Private TxtColor As Color = Color.White
```

اذا وقفت على النقطة الحمراء فتظهر لك الخيارات الاتية

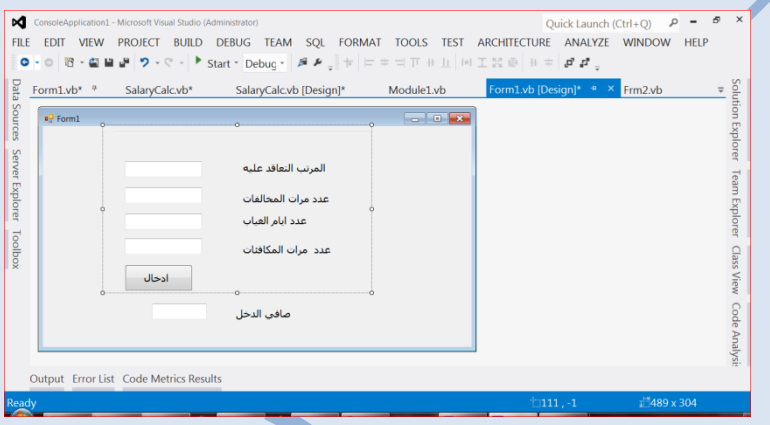


قم بالنقر على الاختيرا الاول وهو يقول انك سوف تستورد فضاء الاسماء الذي به هذا التصنيف و الفضاء هو Drawing النابع من فضاء الاسماء system

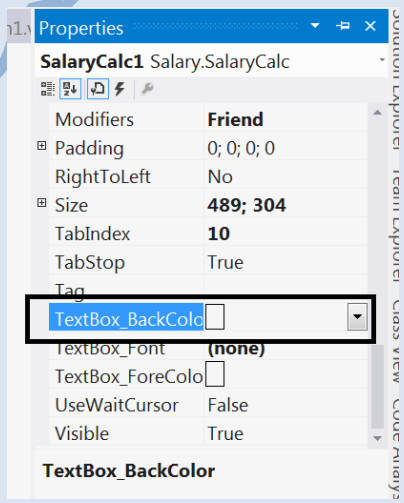
للتاكد من عمل الخاصية قم ببناء المشروع باختيارBuild Solution من القائمة Build ثم اضف نموذج جديد واذهب لصندوق الادوات ولاحظ ظهرة الاداة في الصندوق كما ترى في الشكل الاتي



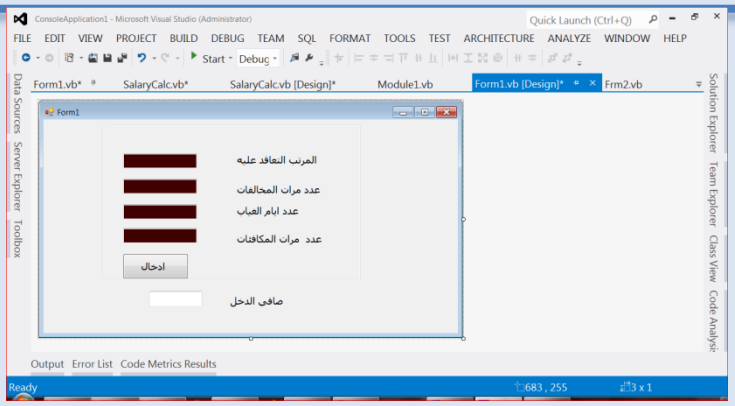
قم بوضعها على اي النافذة او النموذج فتكون كلاتي



ومن نافذ خصائص فتلاحظ ظهور الخاصية التي كتبناها



غير للون الى اي لون تريده فتجده تظهر في صناديق النصوص التي في الاداة



وانصحك ان تضع قيمة افتراضية للخاصية عن طريق اعطاء القيمة للمتغير الخاص TxtColor فضع لها قيمة افتراضية فيكون الكود كلاتي

```
Private TxtColor As Color = Color.White
```

قم بكتابة باقي الخصائص التي تخص صناديق النصوص لتقوم بتعديل كل القيم في صناديق النصوص فيكون الكود كلاتي

```
  Private TxtFont As Font
  Public Property TextBox_Font As Font
      Get
          Return TxtFont
      End Get
      Set(value As Font)
          TxtFont = value
          ContactSalarytxt.Font = value
          DescountMoretxt.Font = value
          AppcentDaytxt.Font = value
          IncenticeMoretxt.Font = value
      End Set
  End Property
```

الكود السابق عن الخاصية مسئولة عن تعديل الخطوط

```
Private TxtFontColor As Color
Public Property TextBox_ForeColor As Color
    Get
        Return TxtFontColor
    End Get
    Set(value As Color)
        TxtFontColor = value
        ContactSalarytxt.ForeColor = value
        DescountMoretxt.ForeColor = value
        AppcentDaytxt.ForeColor = value
        IncenticeMoretxt.ForeColor = value
    End Set
End Property
```

وهذه الخاصية التي تقوم بتغديل لون الخطوط

ننتقل لعمل نفس الخصائص بنفس الطريقة لصناديق العناوين

```
Private LBlFontColor As Color
Public Property label_ForeColor As Color
    Get
        Return LBlFontColor
    End Get
    Set(value As Color)
        LBlFontColor = value
        Label1.ForeColor = value
        Label2.ForeColor = value
        Label3.ForeColor = value
        Label4.ForeColor = value
    End Set
End Property
```

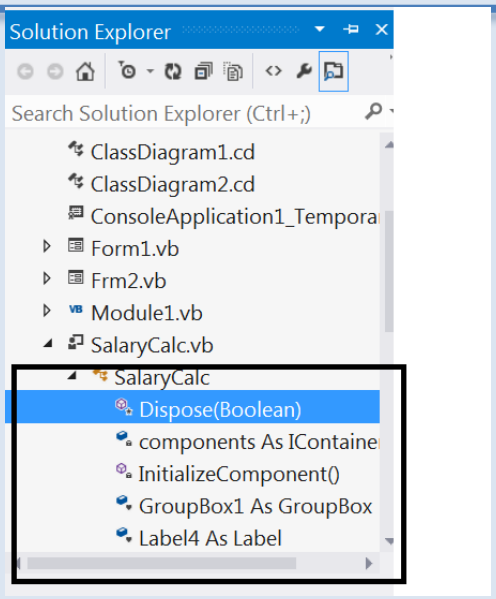
الخاصية السابقة للتغيير لون الخطوط لصناديق العناوين

```
Private LBlFont As Font
Public Property label_Font As Font
    Get
        Return LBlFont
    End Get
    Set(value As Font)
        LBlFont = value
        Label1.Font = value
        Label2.Font = value
        Label3.Font = value
        Label4.Font = value
    End Set
End Property
```

هذه الخاصية التي تتعامل مع شكل الخطوط في صناديق العناوين

4. ننتقل للخطوة الاتية وهي عمل خصائص تخرج النتائج لحساب المرتبات وذلك اما ان نقوم بعمل تصنيف يشبه التصنيف SalaryCls ولكني اريد ان انشاء كائن من التصنيف SalaryCls لكي اعلمك كيف تستخدم تصنيف اخر داخل تصميم الادوات بحيث انك يجب ان تاخذ في الاعتبار انك لا تستطيع عمل inhirt له لان الاداة التي نعمل بها الان ترث من تصنيف اخر وهو userControl **ولا يمكن ان يرث من تصنيفين** .... والسؤال الذي يطرح نفسه كيف عرفت انه يرث من هذا التصنيف ؟؟ الاجابة اتبع الخطوات التية

1- افتح النافذة solution Explorer وقم بفتح الملف SalaryCala على الشكل الاتي وذلك بالضغط على السهم الذي في جانبه



اتضغط على الملف despose وهو في الاصل ليس ملف هو عبارة عن اجراء انهاء مكتوب في التصنيف المسؤول عن تصميم الاداة SalaryCalc اثناء وقت التصميم الذي شرحناة وقلت لك اجل فهمها الى ان ياتي الوقت المناسب فهذا هو الوقت افتح التصنيف بالضغط على هذا الملف وانظر الى ما يرث التصنيف كما هو موضح في الشكل السابق

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated
Partial Class SalaryCalc
    Inherits System.Windows.Forms.UserControl

    'UserControl overrides dispose to clean up the component li
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
```

التصنيف الذي يرث منه

ثم انظر الى الاجراء InitializeComponent ستجد انه يقوم بوضع الادوات التي نستخدمها في التصنيف كما هو موضح في الشكل الاتي

```
Private Sub InitializeComponent()
    Me.GroupBox1 = New System.Windows.Forms.GroupBox()
    Me.Label4 = New System.Windows.Forms.Label()
    Me.IncenticeMoretxt = New System.Windows.Forms.TextBox()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.ContactSalarytxt = New System.Windows.Forms.TextBox()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.DescountMoretxt = New System.Windows.Forms.TextBox()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.AppcentDaytxt = New System.Windows.Forms.TextBox()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.GroupBox1.SuspendLayout()
    Me.SuspendLayout()
    '
    'GroupBox1
    '
    Me.GroupBox1.Controls.Add(Me.Label4)
    Me.GroupBox1.Controls.Add(Me.IncenticeMoretxt)
```

لا يهم الان ما هو الكود المكتوب ولكن لو تمعنت فيه قليلا ستتعرف عليه كيف يكتب وللعلم فقط ان الاجراء يقوم باستدعاء نفسه اثناء الكتابة وذلك من خلال المواصفات المكتوبة فوقه كما هو موضح في الشكل الاتي

```
'UserControl overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
```

هذه المواصفات تقوم بترجمة الكود لتساعدك في التعامل مع الكود في وقت التصميم وسوف اشرحها بالتفاصيل في الفصل القادم باذن الله

نعود الان الى كيفية عمل خصائص لحساب المرتبات وقلنا اننا سنستدعي التصنيف الذي صممناه في الفصول السابقة لحساب المرتبات واذا اردت نسخ الكود ووضعه في الاداة فلا مشكلة في ذلك ولكني اريد ان علمك كيف تستخدم التصنيف بدون ان ترث منه او تنسخه

قم بعمل متغير اسمه mySalary من نوع SalaryCls بانشاء الكائن باستخدام الدالة new فيكون الكود كلاتي

```
Private mysalary As New SalaryCls
```

ثم قم بعمل الخصائص الاتية التي ستقبل البيانات

```
Private ContactSalr As Single
Public Property ContactSalary As Single
    Get
        Return ContactSalr
    End Get
    Set(value As Single)
        ContactSalr = value
        mysalary.ContactSalary = value
    End Set
End Property
```

الخاصية السابقة تستقبل المرتب المتعاقد عليه وتعطيه للمتغير mySalary الذي ينشى التصنيف SalaryCls

```
Private DesCmore As Integer
Public Property DescountMore As Integer
    Get
        Return DesCmore
    End Get
    Set(value As Integer)
        DesCmore = value
        mysalary.DescountMores = value
    End Set
End Property
```

وهذه لحساب عدد مرات المخالفة

```
Private appcent As Integer
Public Property AppcentDay As Integer
    Get
        Return appcent
    End Get
    Set(value As Integer)
        appcent = value
        mysalary.AppsentDay = value
    End Set
End Property
```

وهذه لحساب عدد ايام الغياب

```
Private incCmore As Integer
Public Property IncentiveMore As Integer
    Get
        Return incCmore
    End Get
    Set(value As Integer)
        incCmore = value
        mysalary.incentiveMores = value
    End Set
End Property
```

وهذه الخاصية لحساب عدد مرات المكافئات

ننتقل لكتابة الخصائص المصممة للقرائة فقط

```
Public ReadOnly Property descount As Single
    Get
        Return mysalary.Descounts
    End Get
End Property
```

هذه الخاصية تظهر قيمة اجمالي الخصومات

```
Public ReadOnly Property incentive As Single
    Get
        Return mysalary.incentives
    End Get
End Property
```

هذه لحساب جمالي المكافئات

```
Public ReadOnly Property RealSalary As Single
    Get
        Return mysalary.MySalary
    End Get
End Property


Public ReadOnly Property netincom As Single
    Get
        Return mysalary.Netincome
    End Get
End Property
```

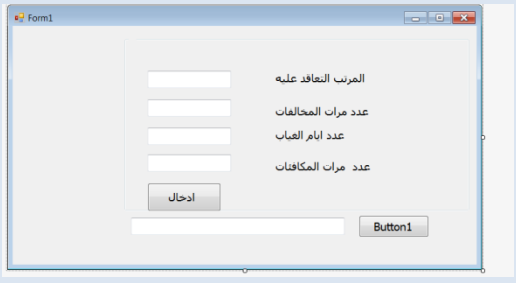
الخاصيتين هما لحساب المرتب بعد خصم ايام الغياب في الخاصية الاولى و صافي الدخل في الثانية

قم الان بعمل Build Solution ولكن كن على علم انك لم تنتهي من عمل الاداة لانك لم تربط الخصائص الخاصة بحساب المرتبات بصناديق النصوص بحيث كل صندوق يعطي القيمة التي بداخله للخاصية المناسبة له وذلك بكتابة الكود الاتي في الحدثClick للزر الذي وضعنا

```
Me.ContactSalr = CSng(ContactSalarytxt.Text)
Me.DescountMore = CInt(DescountMoretxt.Text)
Me.AppcentDay = CInt(AppcentDaytxt.Text)
Me.IncentiveMore = CInt(IncenticeMoretxt.Text)
```

الان نكون قد انتهينا من عمل الاداة قم ببناء المشروع

وضع الاداة على النافذة ثم اضف صندوق نصوص ليكون هو المستقبل لحساب صافي الدخل ويكون اسمه NetIncomTxt ولكن ستجد انك لا تستطيع اعطاء قيمة له الا باستخدام زر اخر ليقوم باعطاء قيمة لصندوق النصوص ويكون اسم الزر فيكون شكل النموذج كلاتي



يكون اسم الزر Btn

قم بكتابة الكود الاتي في الزر Btn

```
Private Sub Button1_Click(sender As Object, e As EventArgs)
    NetIncomTxt.Text = SalaryCalc1.netincom
End Sub
```

قم بتشغيل البرنامج بالضغط على الزر F5 ثم ادخل القيم التي تريدها في صناديق العناوين ليكون الشكل الاتي

طبعا انا الذي كتبت الارقام اكتب انت الارقام التي تريده ثم انقر على الزر Button1 ستجد انه لم يحسب بل اخرج النتيجة 0 لكن قم بالضغط على زر الادخال الذي في الاداة ثم انقر على Button1 ستجده يقوم بحساب المرتب .... طبعا هذه الفكرة سيئة جدا لانها لا تليق ان تخرج برنامج تجبر المستخدم بالضغط على زرين لحساب نتيجة معينة في حين تستطيع انت ان تجعله يضغط على زر واحد بمعنى ادق يجب ان تصنع برنامج يريح المستخدم لا ان يتعبه فلو انك اخرجت هذا البرنامج بهذه الطريقة للمستخدم سوف يرتكب خطا في استخدام البرنامج وخاصة انه سوف يكون تحت ضغط عمل فينقر على زر استخرج الناتج button1 دون ان ينقر على زر الادخال الذي في الاداة فهذا يؤدي الى اظهرا نتائج خطا وسوف تتهم انت الاول بانك لم تصمم برنامج يسهل على المستخدم .....خلاصة القول يجب ان نعيد برمجة الاداة بحيث انها تستخرج الناتج بالنقر على زر واحد فقط وهو زر الاداخال الذي في الاداة ..... فكر معي اريدك ان تفكر كيف نحل هذه المشكلة لوحدك **قبل ان تقرا السطر الاتي**

الحل الوحيد لهذه المشكلة هي عمل حدث يسمى Button_Click او اي اسم تسميه لكن سنسميه هنا بهذا الاسم بحيث انه يشنط عند النقر على الزر الادخال ثم نقوم بكتابة الاجراء الخاص باستخراج الناتج الذي كتبناه على الزر Button1 في اجراء ثم يستدعى عند تنشيط الحدث

قم بانشاء الحدث بحيث يستدعي اجراء فيكون كلاتي

```
Public Event Button_Click()
```

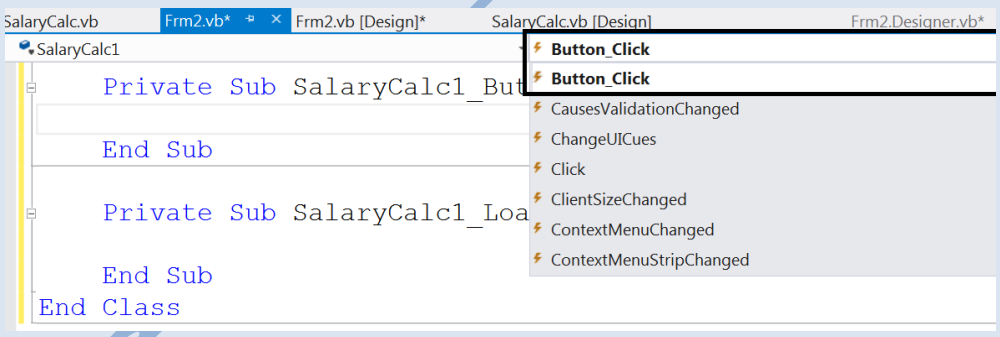
ثم نشط الحدث عند النقر على الزر كما هو في الكود الاتي

```
Private Sub Button1_Click(sender As Object, e As EventArgs)

    Me.ContactSalary = CSng(ContactSalarytxt.Text)
    Me.DescountMore = CInt(DescountMoretxt.Text)
    Me.AppcentDay = CInt(AppcentDaytxt.Text)
    Me.IncentiveMore = CInt(IncenticeMoretxt.Text)
    RaiseEvent Button_Click()

End Sub
```

ثم نقوم ببناء المشروع من القائمة Build ثم افتح التصنيف المسئول عن تصميم النموذج وهو الملف Designer.vb وستلاحظ انه استدعى الحدث بالستخدم الكلمة WithEvent فوفر عليك خطوة الاستدعاء كما هو موضح

```
    End Sub
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents SalaryCalc1 As Salary.SalaryCalc
    Friend WithEvents NetIncomTxt As System.Windows.Forms.TextBox
End Class
```

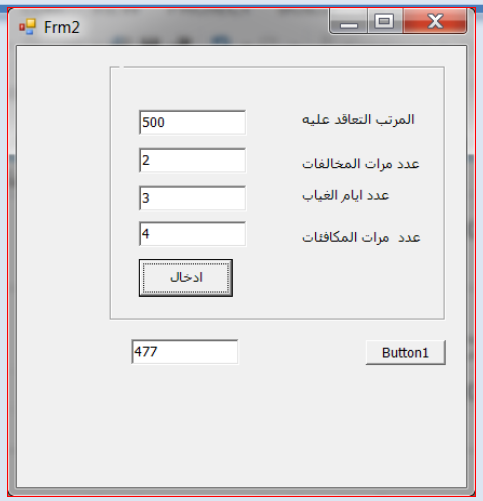
ثم اذهب لشاشة الكود التي تستخدمها في برمجة النموذج وهو الملف resx واذهب الى قائمة الاحداث الخاصة بالادة salaryCalc وستجد الحدث Button_Click كما هو موضح في الشكل الاتي



ثم اتكتب الكود الذي كتبناه في Button1 وهو

```
Private Sub SalaryCalc1_Button_Click() Handles SalaryCalc1
    NetIncomTxt.Text = SalaryCalc1.netincom
End Sub
```

افتح البرنامج و اكتب المرتبات كما كتبنا ثم انقر على زر الادخال ستجد انها تظهر كلاتي



وبذلك تمت عملة برمجة اداة التحكم هذه الطريق طريقة مثالية لحل مشالكنا قبل ان انتقل الى الفصل القادم انصحك ان لا تمسح هذا المثال لاننا سنحتاج الي في باقي فصولنا وانبهك انك تستطيع ان تستغني عن الزر Button1 لانه ليس له اهمية فقد حل الحدث Button_Click محله

انتقل الى الفصل الاتي للتعرف على طرق تحسين اداة الاداة SalaryClac

الفصل الثامن

تحسين اداء اداة التحكم

بعد ان تعلمنا كيف ننشئ الادوات التحكم وتعلمنا كيف تعمل على النموذج فاك ستجد انك تريد تحسين

الاداة فانك تجد خصائص لا تريدها تظهر على نافذة الخصائص وانت لا تريده تظهر على النافذة مثلا

DescoungMor او ContactSalary فانه ليس من اللائق ان تظهر هذه الخصائص على نافذة الخصائص كما انك تريد ان يكون الحدث Button_Click الذي انشاته هو الحدث الافتراضي وغيرها من التحسينات التي يجب ان تضعها على الاداة حتى تصبح اداة قوية .... في هذا الفصل سنتعلم كيف

نزيد من كفائة الاداة باضافة التعليقات و المواصفات التي تقوي من اداءها

التعليقات

سنعود مرة اخرى للتخيل و التصور ... تخيل انك قمت ببرمجة التصنيف المسئول عن تصميم

المربتات SalaryCls ومر عليك بعدما قمت بعمله سنة كاملة وقد انشغلت في عمل برامج وتصينفات اخرى ونيست هذا التصنيف ثم عدت لاستخادمه مرة اخرى فمن الطبيعى تكون قد نسيت الكثير من

الخصائص وخاصة انك قمت بعمل الخصائص ولم تضف اليها التعليقات فمن الطبيعي سوف

تاخذ وقت طويل لكي تتذكر هذه الخصائص

لذا فانه كان من الافضل لك ان تضع التعليقات على كل كود تكتبه حتى تتذكر كل كود اثاء

استدعائه ، لا اعني التعليقات التي نكتبها ورائ كل كود وهي

الكلمة المفتاحية Rem او العلامة ' لا ليست هذه فقط وان كانت مهمة انا اعني التعليقات التي نضعه اعلى كل عنصر سواء كان خاصية او دالة بحيث اذا انشائنا الكائن وقمنا باستدعاء هذه الخاصية التعليقات تظهر في اللTooltip التي تظهر لتوجيهك اثناء كتابة الكود

دعنا ندخل في الموضوع افتح التصنيف SalaryCls او ثم قف اعلى الخاصية MySalary

ثم اكتب هذه الكلمة ''' ستلاحظ ضهرة اداة بهذا الشكل

```
''' <summary>
'''
''' </summary>
''' <value></value>
''' <returns></returns>
''' <remarks></remarks>
Public ReadOnly Property MySalary As Single
    Get
        Return MainSalary(ContactSalary, AppsentDay)

    End Get
End Property
```

هذه العليقات تسمى XMLComment فهي تتعامل مع ملفات Xml لتخزين التعليقات

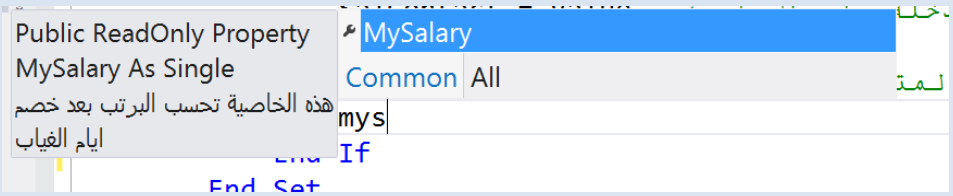
قم بكتابة الكلمات الاتية في الجزء الفارق فهو مخصص لكي تكتب تعريف لهذه الخاصية فاكتب الكلمات الاتي :

هذه الخاصية تحسب المرتب بعد خصم ايام الغياب

فيظهر الكود كلاتي

```
''' <summary>
''' هذه الخاصية تحسب البرتب بعد خصم ايام الغياب
''' </summary>
''' <value></value>
''' <returns></returns>
''' <remarks></remarks>
Public ReadOnly Property MySalary As Single
    Get
        Return MainSalary(ContactSalary, AppsentDay)

    End Get
End Property
```

قم باستدعاء هذه الخاصية في أي مكان داخل التصنيف او خارجه ستجد ظهر التعليق الذي كتبناه على الToolTip كما هو موضح في الشكل الاتي .



ما رايك طبعا ستعجبك كثيرا هذه الفكرة قم بكتابتها في كل الخصائص حتى لا تنسى أي خاصية

جرب كتابته الثلاثة علامات (''')فوق أي دالة او اجراء به معاملات تلاحظ ظهور الاداة التي تخرج التعليقات على الشكل الاتي

```
''' <summary>
'''
''' </summary>
''' <param name="Salary"></param>
''' <param name="AppsDay"></param>
''' <returns></returns>
''' <remarks></remarks>
Private Function MainSalary(ByVal Salary As Single, ByV
    Dim salPerDay As Single 'قوم بحساب قيمة عمل اليوم
    salPerDay = Salary / 30 ' الايام الشهر وهي ٣٠ يوم
    Dim DescountS As Single ' متغير لحساب الخصومات
    DescountS = salPerDay * AppsDay ' عدد ايام الغياب
    Return Salary - DescountS ' طرح المخصوم عن الاساسي
End Function
```

تلاحظ ظهور هذه الTages

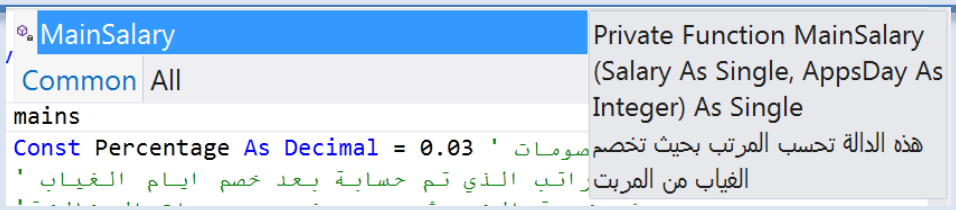
```
''' <summary>
'''
''' </summary>
''' <param name="Salary"></param>
''' <param name="AppsDay"></param>
''' <returns></returns>
''' <remarks></remarks>
```

هذه الTages تقوم بوضع العليقات التي تخص المعاملات قم بكتابة التعليقات كما في الكود الاتي

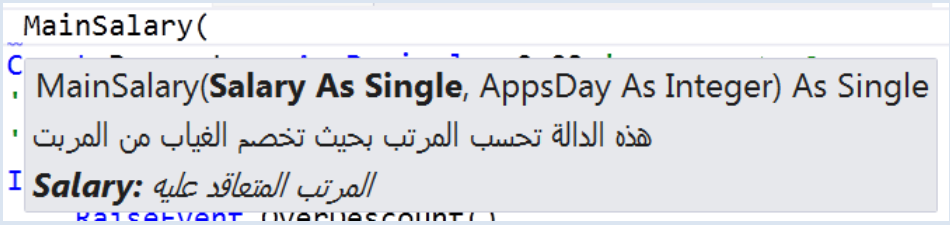
```
''' <summary>
''' هذه الدالة تحسب المرتب بحيث تخصم الغياب من المربت
''' </summary>
''' <param name="Salary">المرتب المتعاقد عليه</param>
''' <param name="AppsDay">عدد ايام الغياب</param>
''' <returns></returns>
''' <remarks></remarks>
```

تعليق على المعاملات

قم باستدعاء الدالة من أي مكان طبعا من داخل التصنيف لانها دالة خاصة



وعند فتح المعاملات تلاحظ ظهور ال Tooltip التي توجه المبرمج للتعامل مع المعاملات كما هو موضح في الشكل الاتي



تلاحظ ظهور التعليق المسئول عن المعامل الاول واذا نتقلت للمعامل الثاني ستلاحظ ظهور تعليق الخاص على المعامل الثاني

يجب ان تعرف ان هذه التعليقات مكتوبة في ملف داخل المجلد Debug الذي يحتوى الملفات التي تشغل المشروع وقت التشغيل اذهب الى هذا المجلد ستجد مجلد يحمل اسم المشروع الذي تعمل فيه وليكن Salary وتجد ان الامتداد Xml افتح الملف ستجد يظهر لك على الشكل الاتي

طبعا أي متصفح انترنت يفتح هذه الملفات ولاحظ التعليقات التي كتبناها

بذلك نكون قد انتهينا من شرح التعليقات قم باستخدامها مع باقي الدوال و الخصائص

المواصفات

هذه المواصفات عبارة عن تصنيفات نقوم بكتابتها فتتعامل مع الادوات في وقت التصميم بحيث تتحكم في الخصائص و الدوال في وقت التصميم بمعنى ادق هي تصنيفات تتعامل مع المترجم وهي

هذه المواصفات مطورة من فضاء الاسماء Syslem. ComponentModel

لذا يجب عليك استيدراته بكلمة imports وهذه المواصفات بها عدة انواع منها يسخدم قبل عناصر التصنيف ومنا يستخدم قبل التصنيف نفسه

سنتناول بعض هذه المواصفات حتى تساعدك في تطوير الادوات

BrowsableAttribute	مسئول عن اظهار الخاصية في نافذة التصفح ام لا
CategoryAttribute	مسئولة عن تجميع الخصائص في مجموعة
DescriptionAttribute	مسئولة عن كتابة وصف و تعليق عن الخاصية
DefaultValueAttribute	وضع قيم افتراضية

كما يوجد مواصفات توضع قبل التصنيف نفسه مثل DefoultEvent , DefoultPoperty

وسوف نستعرض هذه المواصفات بالتفصيل

BrowsableAttribute

هذه المواصفات تسمح الخاصية هل تكون على نافذة الخصائص ام لا

ولكي تستطيع كتابة أي مواصفات اتبع القاعدة الاتية

```
<AttrebutName(Value), Another> _

memper
```

كما تلاحظ اننا نكتب بين القوسين <> التي هي اسم المواصفة التي نستخدمها ولتكن Browsable ثم نضع القيمة بين القوسين لانها منعامل واذا كتبنا ، القاصلة فانه ينتقل لمواصفة اخرى وهكذا وبعد اغلاق القوسين <> يجب ترك مسافة وكتابة _ وهذا يعني انه كتابة الكود على سطر واحد ولكنك تكتبه الكود الذي يكتب على سطر واحد على سطرين فتقوم بوضع العلامة _ وهذه العلامة لا تستخدم فقط في المواصفات بل في أي كود تكتبه فاي كود تكتبه وتريد ان يكون السطر الواحد مقسم سطرين او عدة استطر باستخدام ال _

اما كلمة member فهي تعني الكود التي تكتب تحته مثل التصنيف او خاصية او دالة او اجراء مثلما

نبدا الان بكتابة الكود الاتي فوق خاصية TextBox_BackColor بحيث انها سوف تظهر في نافذة الخصائص كما هو موضح في الكود الاتي

```
<Browsable(True)> _
Public Property TextBox_BackColor As Color
    Get
        Return TxtColor
    End Get
    Set(value As Color)
        TxtColor = value
        ContactSalarytxt.BackColor = value
        DescountMoretxt.BackColor = value
        AppcentDaytxt.BackColor = value
        IncenticeMoretxt.BackColor = value
    End Set
End Property
```

وعند بناء المشروع ستلاحظ ظهرة الخاصية جرب ان تكتبها في خاصية اخرى ولتكن incenticeMore ويكون الكود كلاتي

```
<Browsable(False)> _
Public Property IncentiveMore As Integer
    Get
        Return incCmore
    End Get
    Set(value As Integer)
        incCmore = value
        mysalary.incentiveMores = value
    End Set
End Property
```

بعد بنا المشروع ستجد ان الخاصية اختفت من نافذة الخصائص اننا استخدمنا القيمة False

CategoryAttribute

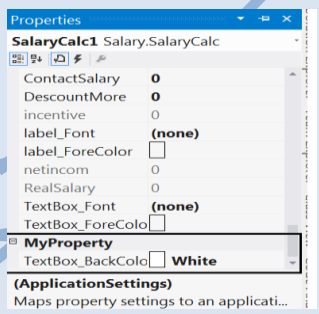
اذا فتحت نافذة الخصائص تجد بها زرين كما هو موضح في الشكل الاتي

اذا نقرت على الترتيب على اساس المجموعات ستجد الخصائص مرتب على اساس المظهر او القيم او التعامل مع البيانات وغيرها فنحن نريد ان نكون مجموعة للخصائص التي كتبناها وتكون المجموعة MyProperty

اكتب الكود الاتي فوق TextBox_BackColor بجانب الخاصية الاولى فيكون الكود كلاتي

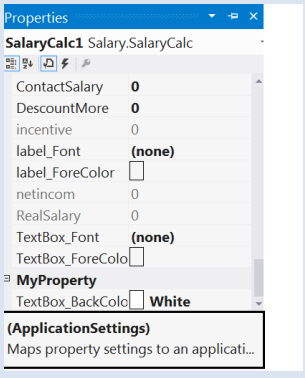
```
<Browsable(True), Category("MyProperty")> _
Public Property TextBox_BackColor As Color
    Get
        Return TxtColor
    End Get
    Set(value As Color)
        TxtColor = value
        ContactSalarytxt.BackColor = value
        DescountMoretxt.BackColor = value
        AppcentDaytxt.BackColor = value
        IncenticeMoretxt.BackColor = value
    End Set
End Property
```

وعند بناء المشروع افتح نافذة الخصائص فتجد الخاصية تظهر في المجموعة في الشكل الاتي



DescriptionAttribute

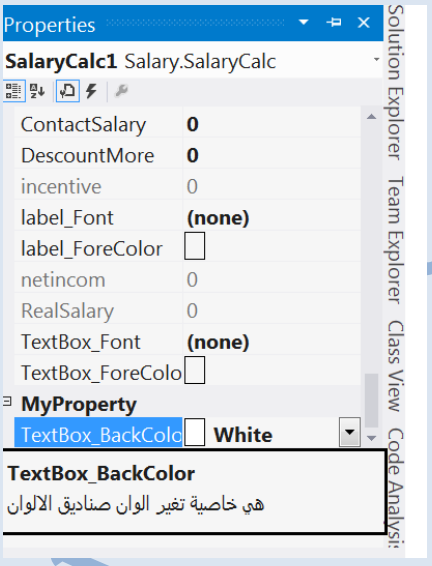
في هذه المواصفة اريدك ان تلقي نظرة على اخر جزء في نافذة الخصائص فانك تجد انك اذا وقفت على اي خاصية فان التعليق يطهر في اخر النافذة كما هو موضح في الشكل الاتي



سوف نكتب الكود الاتي مثل الاكواد السابقة وسوف نكتبها ايضا في الخاصية textbox_BackColor كما هو موضح في الكود الاتي

```
<Browsable(True), Category("MyProperty"), Description("هي خاصية تغير الوان صناديق الالوان")> _
Public Property TextBox_BackColor As Color
    Get
        Return TxtColor
    End Get
```

طبعا انا اظهرت فقط اعلى الخاصية قم بانشاء المشروع وانظر اسفل النافذة فسوف تلاحظ التعليق الذي كتبناه كما هو موضح الشكل الاتي



DefaultValueAttribute

هنا نقوم بعمل قيمة افتراضية للخاصية فقم بكتابة الكود الاتي

```
  <Browsable(True), Category("MyProperty"), Description("ي خاصية تغير الوان صناديق الالوان
DefaultValue(GetType(Color), "red")> _
  Public Property TextBox_BackColor As Color
      Get
          Return TxtColor
      End Get
      Set(value As Color)
          TxtColor = value
          ContactSalarytxt.BackColor = value
          DescountMoretxt.BackColor = value
          AppcentDaytxt.BackColor = value
          IncenticeMoretxt.BackColor = value
      End Set
```

ستلاحظ اننا استخدمنا الدالة GetType هذه الدالة تقوم باعطاء نوع للبيانات ويقوم بعمل تحويل بحيث انك تستخدم معاملين هم getType(*Type*,*Onject*)

Type هنا نكتب النوع الذي نريده

Object البيان ونقوم بتحويل اي لا اي نوع خارج اي بيانات نستخدم المعامل النصي بمعنى ننا نريد اختيار اللون الاحمر واللون اصلا ليس من انواع البيانات لذا يجب اننا نستخدم النصوص لكي نحولها الى اللون

DefoultEvent و DefoultPoperty

هذه المواصفات تكتب اعلى التصنيف لكي تحدد حدث افتراضي يطرعند النقر مرتين على الاداة و الخاصية الافتراضية عن فتح نافذة الخصائص لاول مرة اكتب الكود الاتي اعلى التصنيف .

```
<DefaultEvent("Button_Click"), DefaultProperty("TextBox_BackColor")> _
Public Class SalaryCalc
#Region "خصائص مسئولة عن تصميم الاداة"
    Private TxtColor As Color = Color.Red
```

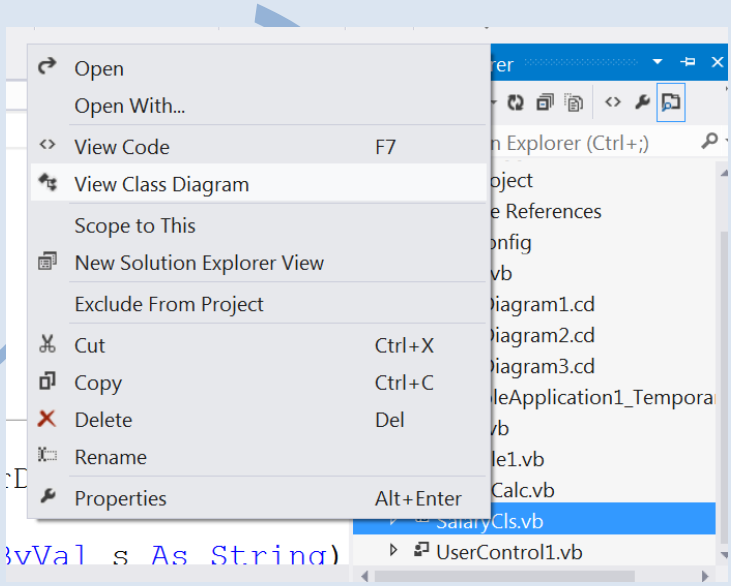
وعند بناء المشروع قم النقر مرتين على الاداة ستلاحظ ظهور الحدث الذي انشاناه و افتح نافذة الخصائص ستظهر الخاصية textbox_BackColor

وبذلك نكون قد انتهينا من شرح اهم المواصفات طبعا هذه المواصفات يجب ان نفرد لها كتاب خاص ولكن اعطيتك ما يفيدك في تطوير ادواتك واذا اردت معرفة المزيد انصحك بزيارة الموقع الخاص بشركة ماكروسوفت http://msdn.microsoft.com الذي يحتوى على كل محتويات الNetFramWark وبها شرح جيد لكل التصنيفات و العناصر

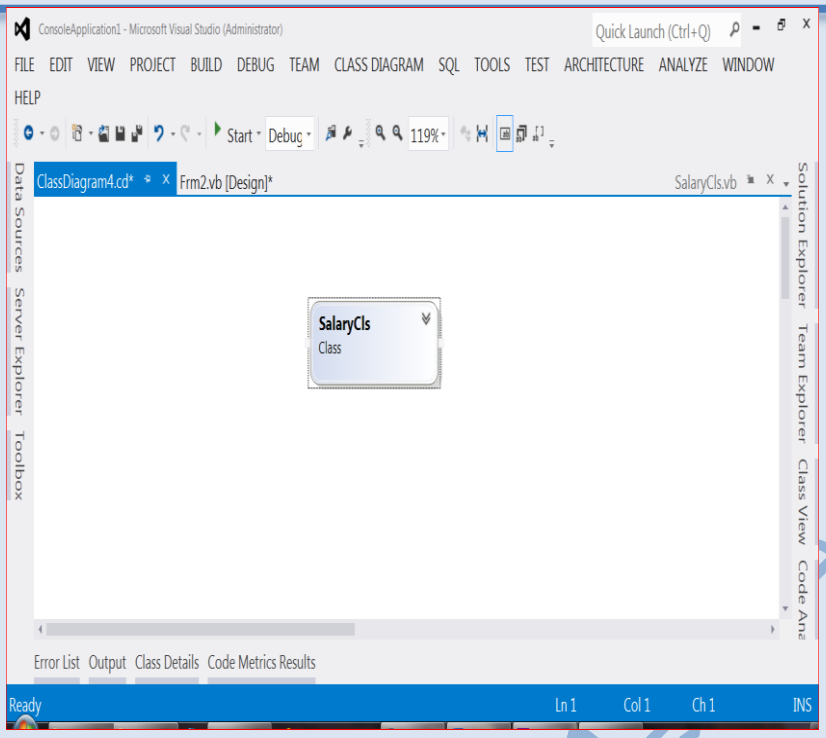
وضع التصنيف في Daigram

بدل ما تقوم بكتابة الاكواد التي تنشئ التصنيف و العناصر ما رايك في ان تقوم بعمل تصميم رسومي للتصنيفات

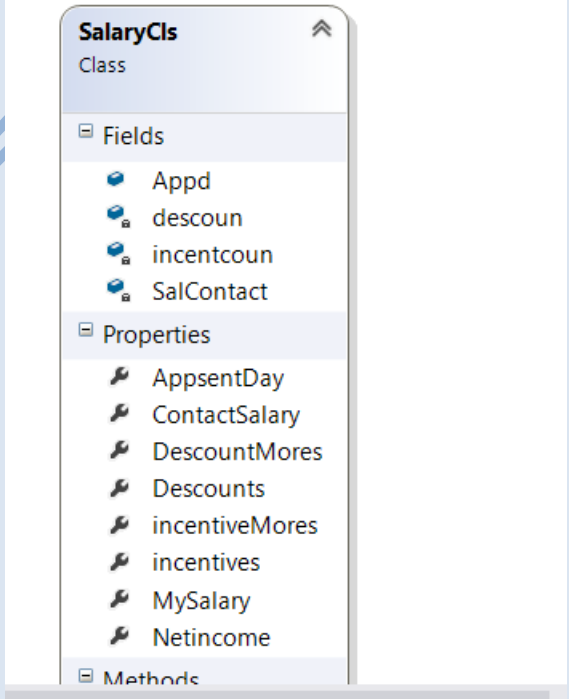
قف على التصنيف SalaryCls في نافذة المتصفح وانقر عليه باليمني واختر من القائمة View Class Daigram كما هو موضح



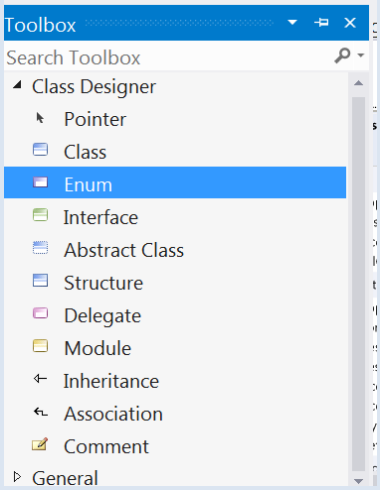
فتظهر لك الشاشة الاتية



ستلاحظ ان التصيف تحول الى شكل بياني انقر على السهم ق سيظهر لك (كلاتي



كما تلاحظ تظهر لك عناصر التصنيف مقسمة على اساس نوع العنصر فالمتغيرات مع بعضها و الخصائص مع بعضها ولكن الدوال ولاجرائات في مجموعة تسمى methods يعني طرق ما رايك ان تقوم بعمل اضافة للخاصية جديدة للتصنيف افتح صندوق الادوات و ستلاحظ ان الادوات هي عبارة عن عناصر التصنيف كما هو موضح في الشكل الاتي

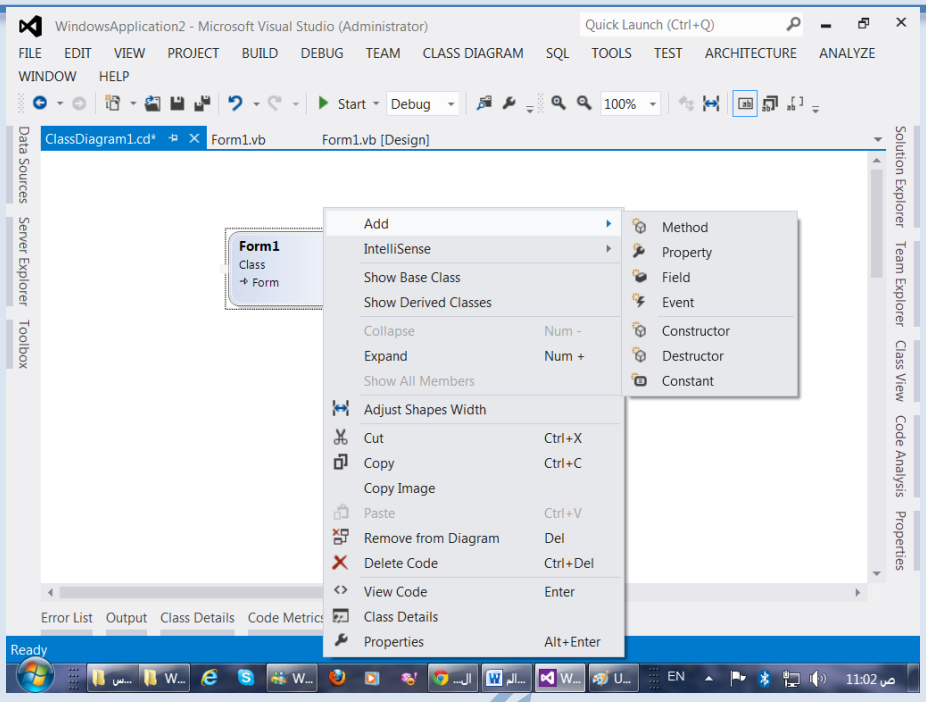


كما تلاحظ وجود ادوات مثل اداة لعمل الواجهات واداة لعمل التعدد Enum واداه اخرى لعمل التوريث من التصنيف الاساسي

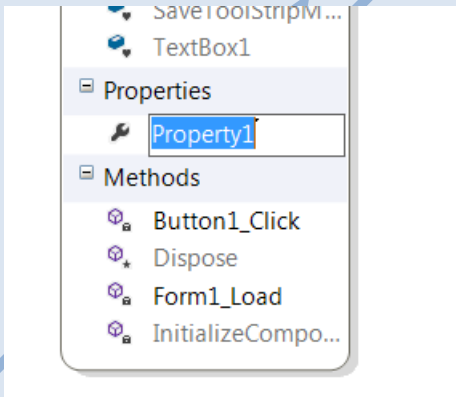
اذا نقرت على اي عنصر او اي تصنيف تستطيع ان تجد ان كل هذا خصائص وهي الخصائص توفر عليك كتابة سطور من الاكواد افتح مثلا الخصائص التي تخص عنصر الدوال ستجد ان هناك خصائص توفر عليك مجهود كبير في كتابة الاكواد مثل خاصية Summary فهي تنشئ التعليات Xmt وخاصية inhirtanceModifaire التي تحدد هل يمكن عمل overload او ovwrridy وغيرها من الخصائص

اريد ان انبهك ان التصميم الرسومي للتصنيف Class Daigram ما هو الا لوضع العناصر فقط انما البرمجة الداخلية واستخدام جملة التدفق فهي ليست في هذا التصميم فيجب كتابتها بيدك

ولكي تقوم باضافة عناصر على التصنيف انقر على التصنيف بالزر الايمن للماوس ثم اختر Add من القائمة الفرعي فتظهر لك العناصر التي تضيفها كما في الشكل الاتي



اختر اي عنصر فيظهر لك على الشكل الاتي ويكون جاهز لكتابة الاسم عليه فمثلا سوف نختار خاصية فتظهر لنا على الشكل الاتي

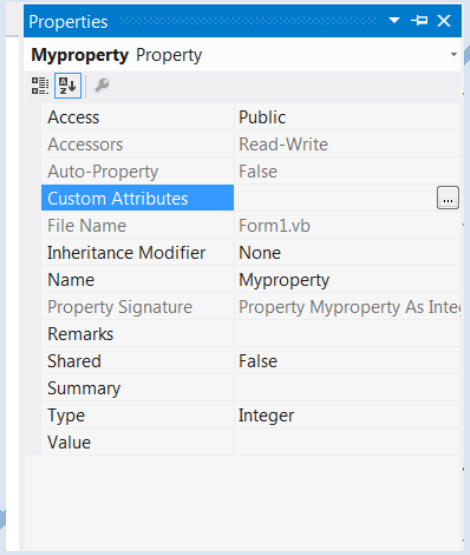


فنقوم بكتابة الاسم عليها وليكن Myproparty

واذا نقرت مرتين علها تظهر لك الخاصية في شاشة الكود

```
Public Property Myproperty As Integer
    Get

    End Get
    Set(value As Integer)

    End Set
End Property
```

نعو] للشاشة تصميم التصنيف لو نقرنا على الخاصية الزر الايمن للماوس واخترانا propreties فتظهر لنا شاشة الخصائص



لو تلاحظ ان الخصائص تصف العنصر الذي نقوم بانشاءه داخل التصنيف فسوف نستعرض بعض الخصائص التي تخص العنصر الذي من نوع خاصية property

Access	تعني محددات الوصول كpublic , privat , Friend
Name	اسم الخاصية
Type	نوع الخاصية
Shared	هل الخاصية مشتركة ام حالة
Summary	اضافة تعليقات Xml في الوسم Summary
Value	اضافة تعليقات Xml في الوسم Value
Inheritance	تعطي اختيارات هل العنصر او الخاصية قابل للتعددoverload او

Modifier	overridable او MustOverride او NotOverridable

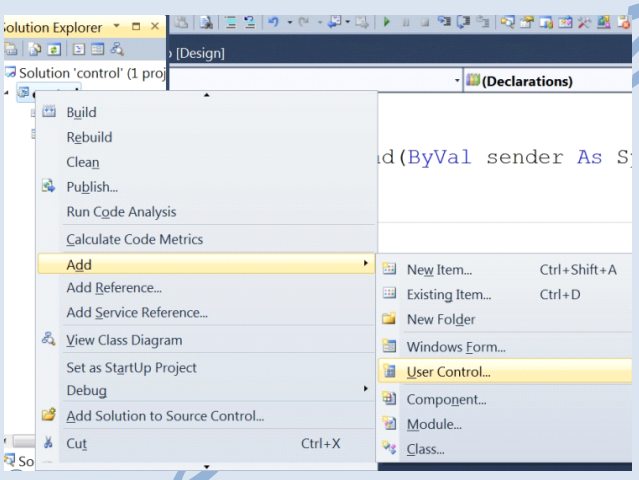
البدايات المضيئة

الفصل التاسع

Component

في الفصول السابقة شرحنا كيف نصمم ادوات تحكم لتسهيل عليك برمجة تطبيقات النوافذ بكافئة .... ما رايك ان تقوم بعمل ادوات منفردة ؟؟؟ سوف نتعلم في هذا الفصل كيف نقوم بعمل زر يتغير لون خلفيته اذا مر عليه الماوس فيكون اذا هو زرك التي تستخدمه في تطبيقاتك وتقوم بتطويره بحيث يكون له شكل مختلف مما يعطي لتطبيقاتك شكلها المميز عن باقي التطبيقات حتى في الشكل الخارجي تابع معي المثال الاتي

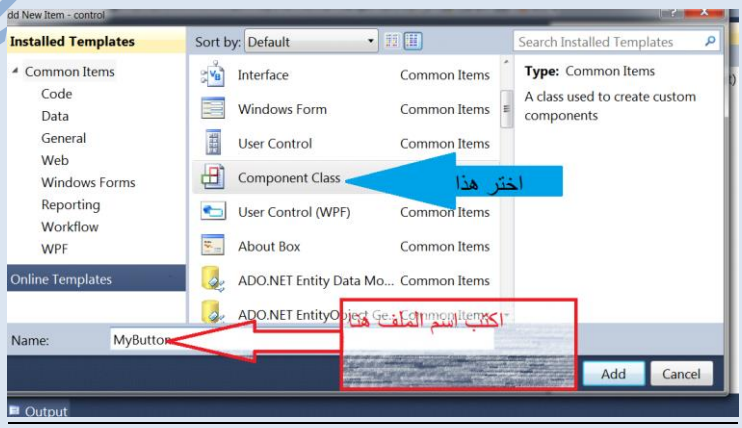
افتح مشروع جديد ثم من نافذة Solution Explorer اضغت بالزر الايمن للماوس على اسم المشورع وليكن Contolثم اختر Add ثم اختر user control كما في الشكل التالي



ثم اكتب اسم الكونترول وليكن MyButton

ثم اضغط زرAdd

كما في الشكل الاتي



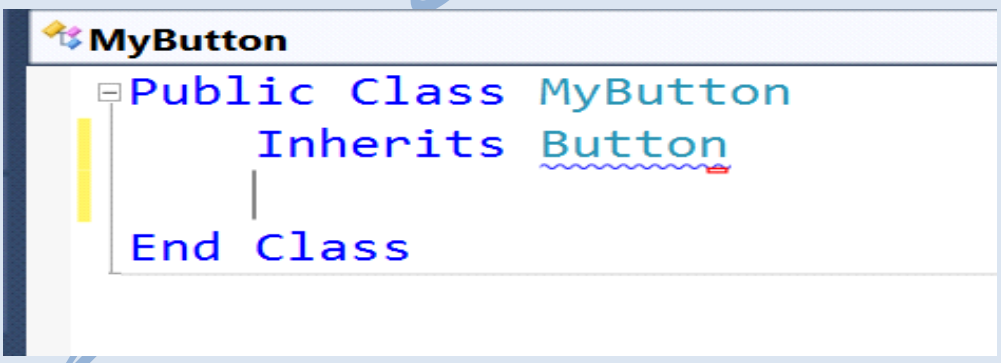
فيظهر لك الشكل كلاتي

وهذا يعني انك لن تستخدم الواجهة الرسومية بل ستعتمد على كتابة الكود في عمل الكونترول

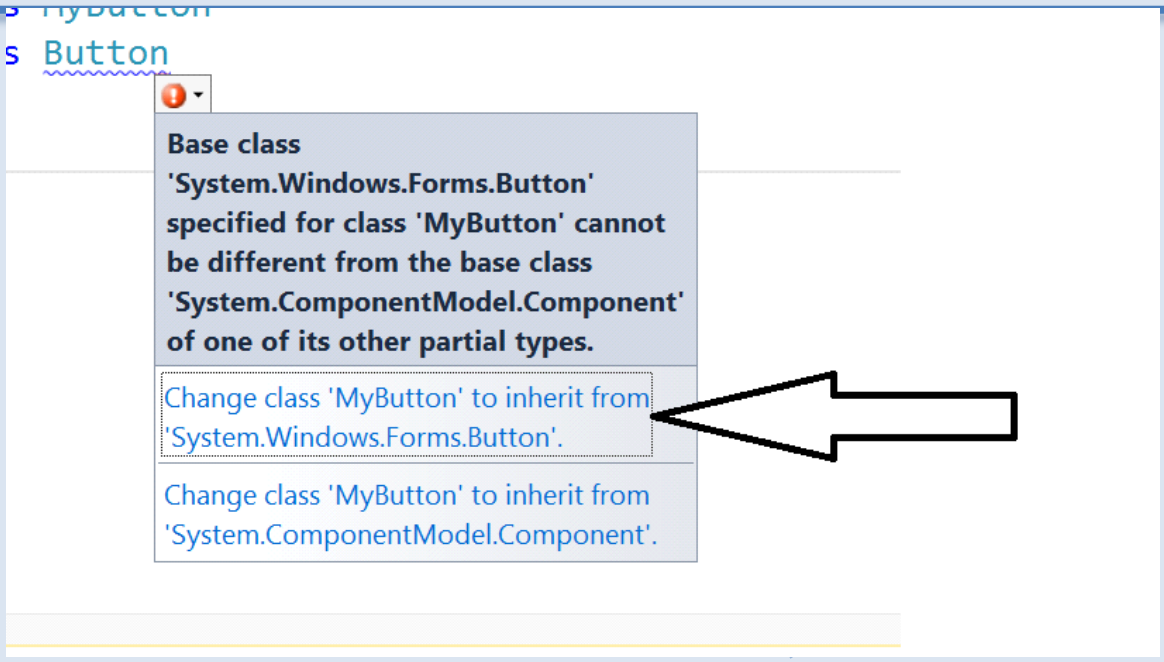
ادخل على شاشة الاكواد بالنقر مرتين على المربع الكونترول ثم اكتب الكود الاتي

```
Inherits Button
```

تحت اسم الكلاس لكن شكل الكود كلاتي



لاحظ وجود خط ازرق تحت كلمة Button وهذا يعني وجود مشكلة في فهم الكود عند المترجم ولكي نعرف المشكلة نقف بالماوس على النقطة الحمراء فتظهر لنا النافذة الاتية



هذه النافذة تعطيك حلول لحل المشكلة فالمشكلة هي ان المعالج يسالك ال تريد ان يرث الكونترول الذي تريدة من كلاس الخاص بال buttonاوكلاس خاص بال user controlفاختر الاختيار الاول

وبهذا نكون مستعدين لكتابة الكود اللازم للكونترول

- **كتابة الكود**

و الان حان وقت كتابة الكونترول وسوف نضع الان تصور عن هذا الكونترول

- هو زر يتغير لون الخلفية عند المرور عليه من الماوس
- و تغير لون الخلفية للون اخر عندما يتم الضغط عليه
- ويتغير لون النص المكتوب عليه عند مرور الماوس عليه

ولكي يتم ذلك سوف نقوم بكتابة ثلاثة خصائص property تخص الخلفية و خاصيتان للون النص ثم تقوم بعمل اجراء مسؤول عن تغيير لون الخلفية واجراء اخر مسؤول عن تغيير لون الخط وسوف نستعمل الاحداث الاتية

MouseLeave , MouseMove , Click

و الان نبدا بكتابة الكود : سيكون مقسم الى ثلاثة اقسام قسم خاص بالخصائص التي تتعامل مع الخلفية و قسم خاص للتعامل مع لون الخص و قسم خاص بالاجرائات ولكي يتم تقسم صفحة الكود نستخدم امر

```
#Region"اسم المجموعة"
```

فيتم كتابة الكود بالشكل التالي ويكون دخل الكلاس الخاص و الكونترول

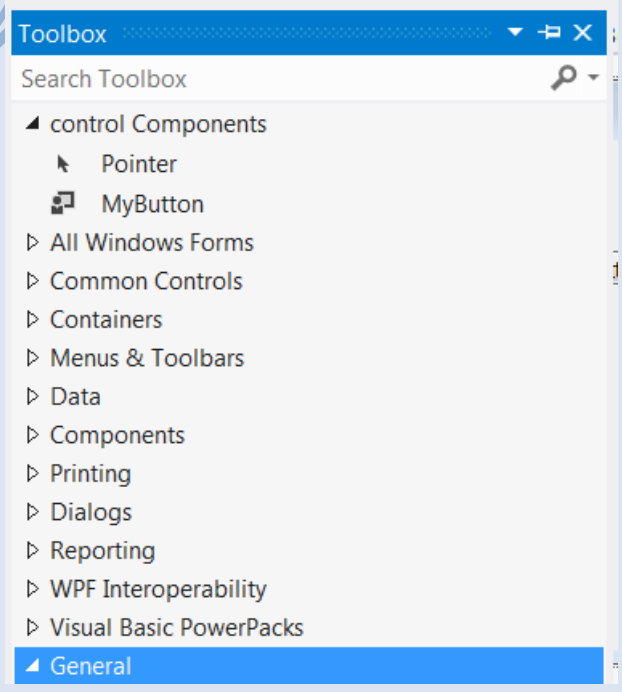
```
#Region "الخصائص التي تتعامل مع الخلفية"

#End Region
#Region "الخصائص التي تتعامل مع الخط"
```

```
#End Region
#Region "الاجراءات"

#End Region
```

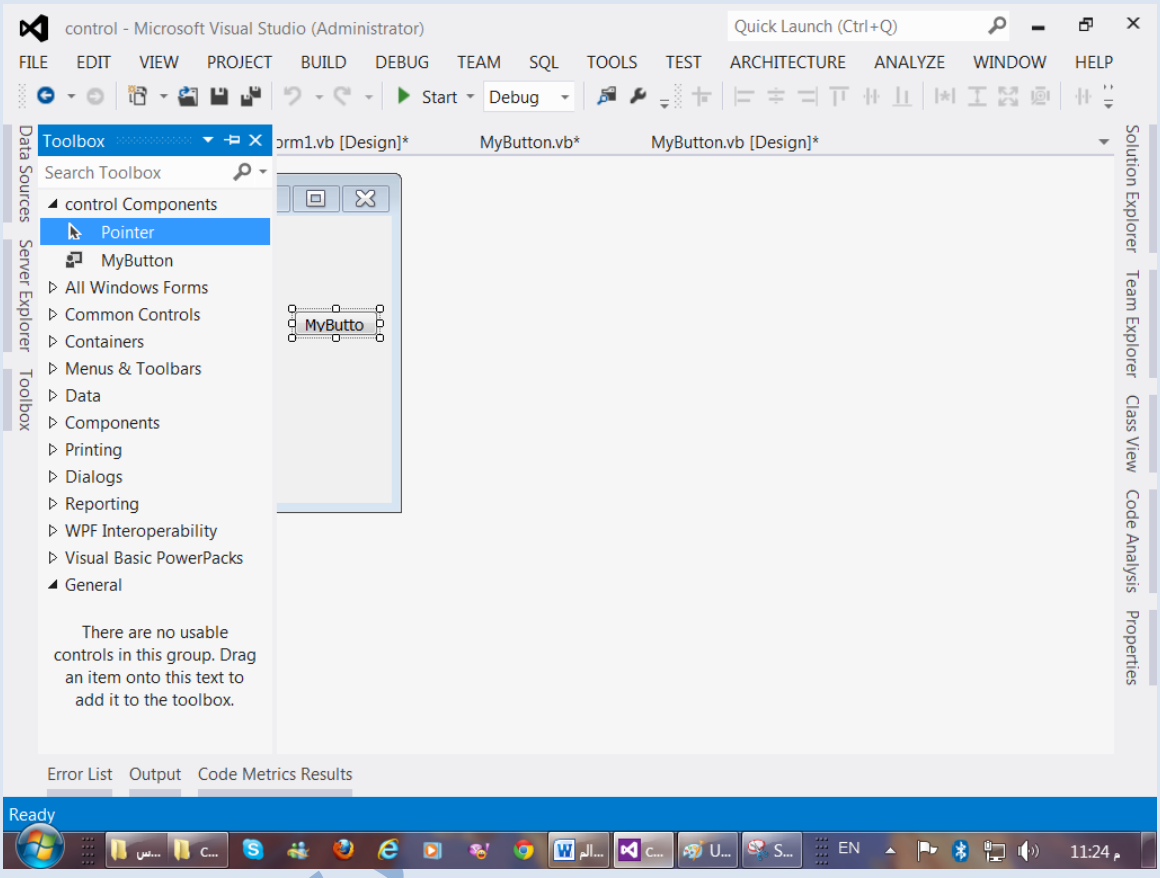
نكتب الان الخاصائص التي تتعامل مع الخلفيات ونبدا الخاصية التي تغيرلون الخلفية عند مرور الماوس عيها ويكون الكود كلاتي

```
Private MousemoveC As Color = Color.Red
    Public Property MouseMoveColor As Color
        Get
            Return MousemoveC
        End Get
        Set(ByVal value As Color)
            MousemoveC = value

        End Set
    End Property
```

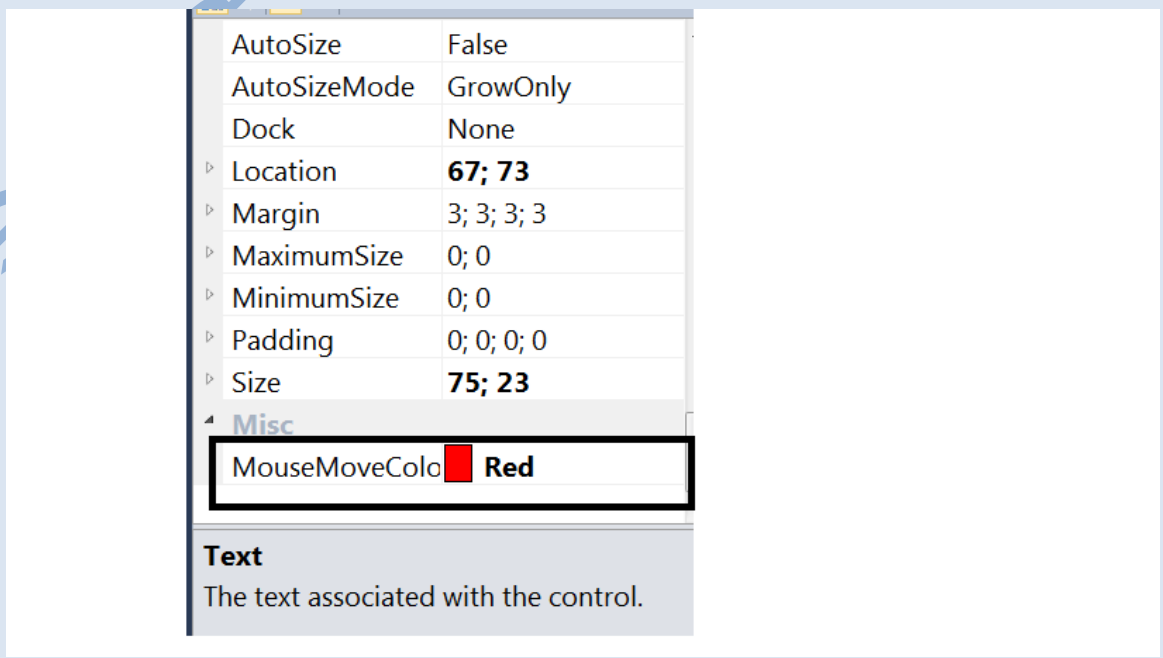
في هذا الكود قمنا بالاعلان عن متغير خاص يحمل لون الخلفية الافتراضي و ثم قمنا بالاعلان عن خاصية عامة لانها ستظهر لنا عند استخادمها في مشروعاتنا ولكي نتاكد من ذلك نقوم ببناء المشروع بالضغط على Build ثم نتجه الى اضافة فورم الى المشروع ثم نفتح صندوق الادوات فنجد الكونترول الذي قمنا بتصميمه موجود كما هو موضح في الصورة الاتية



فنقوم بسحبه على الفورم فيكون في الشكل التي



فعندما نفتح نافذة الخصائص نجد الن الخاصية التي كتبناها موجودة و بنفس اللون الافتراضي الذي كتبناه



ولكن لن يتغير لون الزر عند مرور الماوس حتى لو كان في وقت التشغيل لاننا لم نعطي له الامر بذلك دعنا الان نعود للكونترول ونكتب باقي الكود في الخصائص التي تتغيل لون الخلفية عند مغادرة الماوس و عند النقر وكون اسمهما MouseleveColorو MouseClickColor

ويكون شكل الكود كلاتي

```
#Region "الخصائص التي تتعامل مع الخلفية"
    Private MousemoveC As Color = Color.Red
    Public Property MouseMoveColor As Color
        Get
            Return MousemoveC
        End Get
        Set(ByVal value As Color)
            MousemoveC = value

        End Set
    End Property
    Private MouseClickC As Color = Color.Yellow
    Public Property MouseClickColor As Color
        Get
            Return MouseClickC
        End Get
        Set(ByVal value As Color)
            MouseClickC = value
        End Set
    End Property
    Private MouseleveC As Color = Color.Gray
    Public Property MouseleaveColor As Color
        Get
            Return MouseleveC
        End Get
        Set(ByVal value As Color)
            MouseleveC = value
        End Set
    End Property

#End Region
```

وسوف نطبق هذا ايضا في الخصائص التي تتعامل مع لون الخط

```
#Region "الخصائص التي تتعامل مع لون الخط "
```

```
    Private MousemovefC As Color = Color.Red
    Public Property MouseMoveFontColor As Color
        Get
            Return MousemovefC
        End Get
        Set(ByVal value As Color)
            MousemovefC = value

        End Set
    End Property
    Private MouseleaveFC As Color = Color.Yellow
    Public Property MouseleaveFontColor As Color
        Get
            Return MouseleaveFC
        End Get
        Set(ByVal value As Color)
            MouseleaveFC = value
        End Set
    End Property

#End Region
```

و الان نتقل الى الاجراءات فنقوم بعمل اجرائين sub واحد يقوم بتغيير الخلفية و اخر يقوم بتغيير لون الخط

اجراء تغيير لون الخلفية يكون هكذا

```
Private Sub BackColorChange(ByVal meColor As Color)
        Me.BackColor = meColor
    End Sub
```

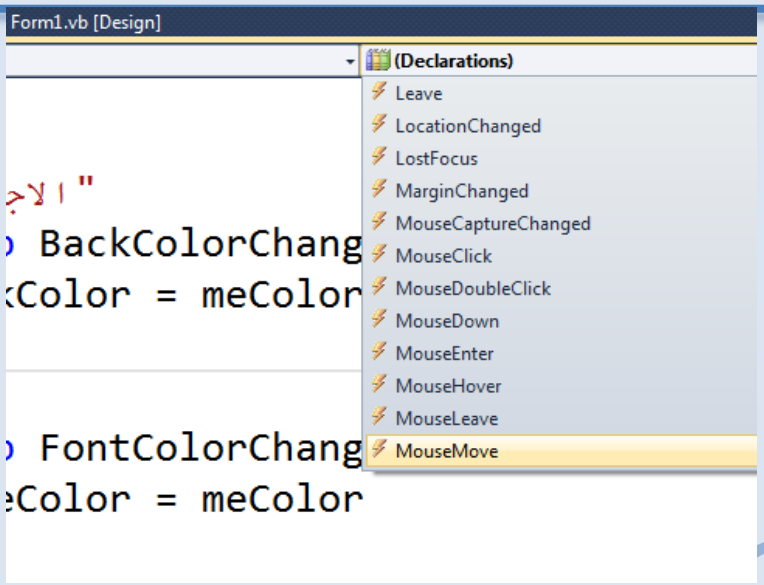
نلاحظ اننا اعلنا عن اجراء خاص حتى لا يستخدمة مبرمج اخر فهو خاص فقط بهذا الكونترول ثم اعطيناه معامل اسمه meColor يقوم باستقبال قيمة اللون ثم يعطيها للخاصية BackColor واللتي كبتناه في السطر الاول واستخدما كلمة Me لاننا نتعامل مع نفس التصنيف الذي نكتب فيه الكود

ثم نقوم بكتابة اجراء اخر الخاص ب foreColr فيكون الكود كلاتي

```
Private Sub FontColorChange(ByVal meColor As Color)

        Me.ForeColor = meColor

    End Sub
```

قمنا الان بعمل الاجراءات ولكن يجب ان نستدعيها حيث انها ستغير لون الخلفية التي تخص الزر و لون الخط عندما ينشط الحدث الخاص بmouseMove و الحدث الخاص بالmouseLeave

وسوف نختار الاحداث من قائمة الاحداث في شاشة الكود التي تخص الزر Button كما هو موضح في الشكل الاتي



عندما نختار الحدث mousemove نكتب الكود الاتي

```
Private Sub MyButton_MouseMove(sender As Object, e As MouseEventArgs)
    Me.BackColorChange(Me.MouseMoveColor)
    Me.FontColorChange(Me.MouseMoveFontColor)
End Sub
```

في الاجراء المرتبط بالحدث mouseMove استدعينا اجراء BackColorChange ثم قمنا باضافة الخاصية mouseMoveColor في المعامل حتى تتغير لون الخلفية عند مرور الماوس وكذلك في اجراء FontColorChangeاضفنا خاصية MouseMoveFontColor

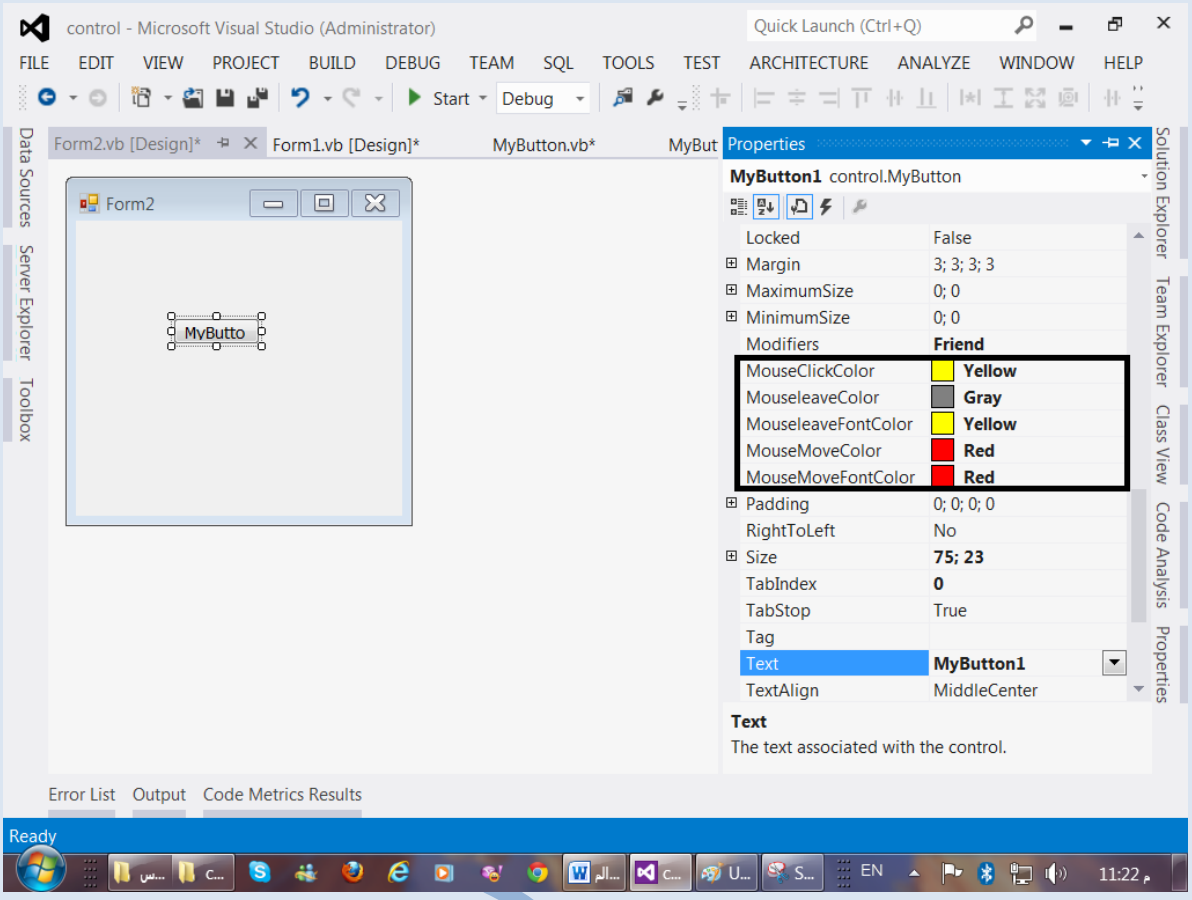
وعندما نختار حدث mouseLeave وذلك لارجاع اللون كما كان عندما يغادر المواس مكان الزر فنكتب الكود الاتي ويكون استعداء نفس الاجرائين لكن نضيف الخاصئص التي تخص مغادرة الماوس كما في الشكل الاتي

```
Private Sub MyButton_MouseLeave(sender As Object, e As EventArgs)
    Me.BackColorChange(Me.MouseleaveColor)
    Me.FontColorChange(Me.MouseleaveFontColor)
End Sub
```

وعندما نختار الحدث Click نكتب الكود الاتي

```
Private Sub MyButton_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
    BackColorChange(Me.MouseClickColor)
    FontColorChange(Me.MouseMoveFontColor)

End Sub
```

وبذلك نكون قد انتهينا من كتابة الكود نقوم ببناء المشروع بالضغط على Build ثم ننتقل الفورم الذي وضعنا فيه الزر فنجد الخصائص مفعلة وموجودة في قائمة الخصائص فنقوم بالتعديل فيها كما نشاء



ونقوم بتشغيل البرنامج بالظغط على F5 فنجد الزر يتغير لون خلفيته و لون الخط عند مرور الماوس

وبذلك نكون انتهينا من عمل الزر ارجو ان تكون قد استفدت من هذة المادة وارجو من اعداة قرائتها مرة اخر وتقوم بتجربتها في ادوات اخر مثل TextBox بحيث يتغير لونه عند الكتابة ليه وعند مغادرته

كما اترك لك ان تضيف انت عليها المواصفات و التعليقات بالشكل و الطريقة التي تريدها

الفصل العاشر

اكتشاف الاخطاء وتصحيحها

نحن نعلم ان اي عمل بشري دائما يشوبه النقص و الخطأ فالكمال لله وحده لذا فانك مهما قمت ببرمجة برامج كثير كبيرة كانت او صغيره فمن المؤكد انك سوف تخطئ مهما كنت مبرمجا محترف لذا فاني في هذا الفصل اشرح لك كيف تقلل نسبة الخطا في تطبيقاتك بان تنظم اكوادك فكل الشرح السابق الذي شرحته من تصميم تصنيفات و عمل ادوات تحكم ما هو الا محاولة لتنظيم الكود واكتشاف الاخطاء والوصول لنقاط الضعف وتصحيحها لذا انصحك قبل ان تقوم بعمل اي برنامج ان تخطط له جيدا وتحدد الاوامر المتشابهة ووضعها في تصنيفات وغيرها من كل ما شرحته في الكتاب وبعد ان تقوم بذلك تتابع معي في هذا الفصل الطرق التي تستخدمها في اكتشاف الاخطاء لانك مهما كنت حريصا لمنع وقوع اي خطا فانك ستكتشف انك وقعت في اخطاء بدون اي قصد لانك لم تعمل حسابها وانت تصمم تطبيقاتك لذا تابع معي هذا الفصل لمعرفة كيف تكتشف الاخطاء

اي خطا في البرمجة لن يخرج من ثلاثة انواع الاتية

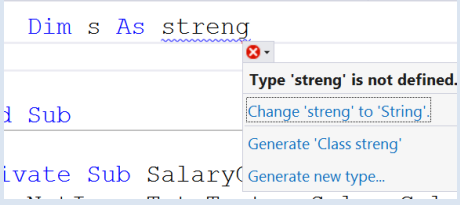
1- خطأ وقت التصميم
2- خطا في وقت التشغيل
3- الشوائب

خطأ وقت التصميم

هذا الخطا من اسهل الاخطاء لان هذا الخطا يحدث عندما تقوم بكتابة الكود ويكون هذا الخطا خطا هجائي بمعنى انك تكتب الكود بشكل خطئ مبعنى انك تبدل ما تكتب كلمة String تكتب streng فتجد ان المترجم قد اكتشف خطاك وانت تكتب البرنامج فيضع خط ازرق تحت الكلمة كما هو موضح في الشكل الاتي

```
Dim s As streng
```

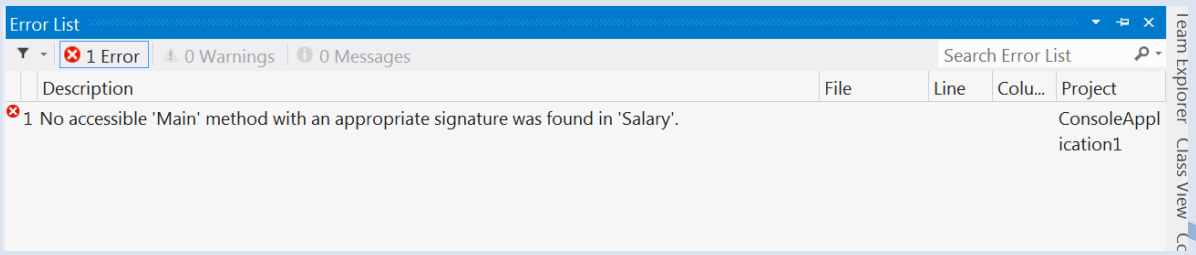
عند الوقوف عند النقطة الحمرا فانها تظهربعض المقترحات كما هو موضح في الشكل الاتي



وتجد حل اخر لهذه المشكلة هي قائمة العناصر الالي Auto list member وهي القائمة التي تخرج اثناء كتابة الكود وهي تساعد كل المبرمجين سواء كانوا محترفين او مبتدئين فهي عند كتابة الكود فانا تعرض لك العناصر مرتبة ترتيب

ابجدي كما اذا كتبت اي كائن ثم ضغطت على Dot (.) فانها تعرض لك العناصر التي بداخل الكائن مرتبة ترتيب ابجدي

وبعد كل هذا يمكن ان تكتب خطا وانت لا تدري وعند تشغيل البرنامج تظهر لك الرسالة الاتية



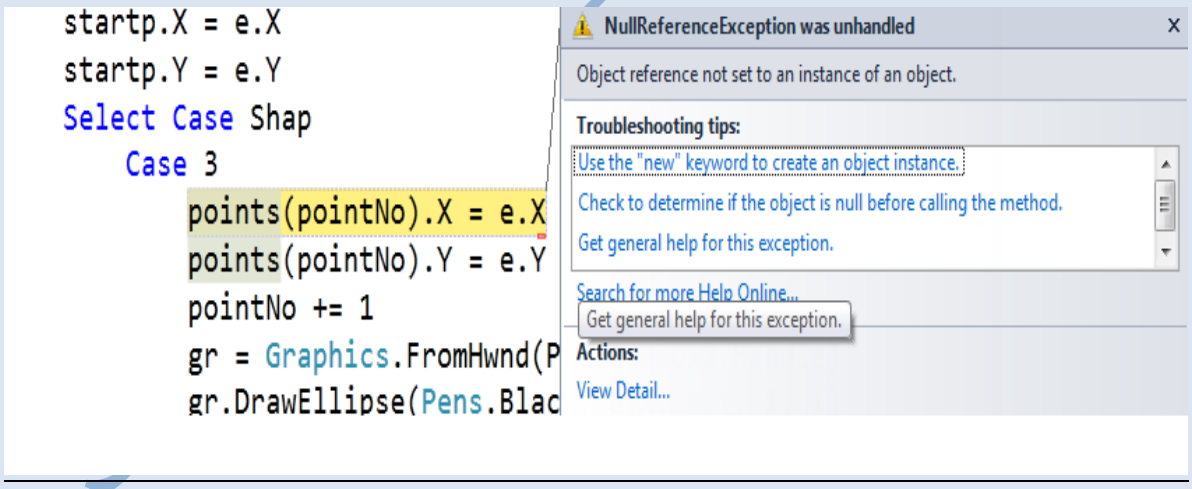
وهذه نافذة تكشف الخطا تعرض نوع الخطا ورقم السطر و عدد الاخطاء كما هو موضح في الصورة السابقة

عند النقر عليها تذهب الى الخطا

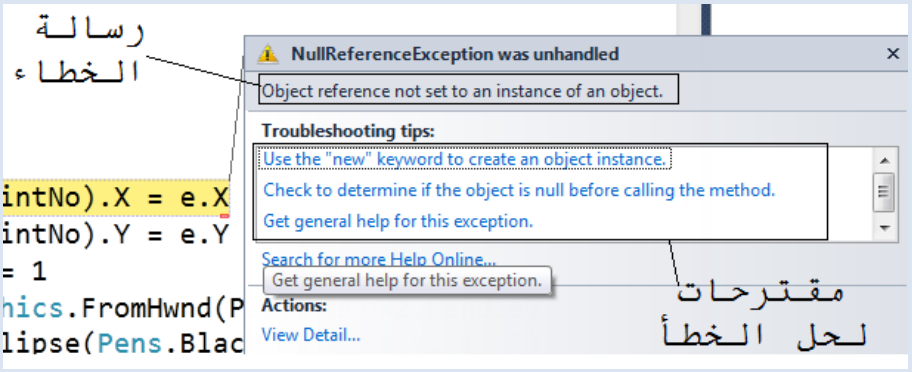
بهذه الطريق تقوم بالكشاف الاخطاء الهجائية وقت التشغيل ننتقل الان الى النوع الثاني من الخطاء

خطأ وقت التشغيل (الاستثناءات)

هذا الخطا يحدث عند تشغيل البرنامج فيجده يعمل بشكل جيد ثم فجأة يخرج وتظهر لك الرسالة الاتية

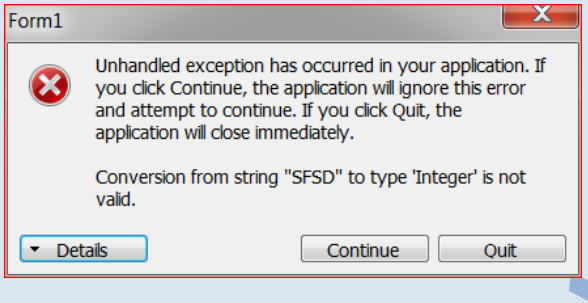


هذا صورة تعبر عن البرنامج يخرج عن العمل ثم يعرض الرسالة التي عرضناها في الشكل السابق لو نظرنا الى هذه الرسالة سنجد انها تكون من الاقسام الاتية

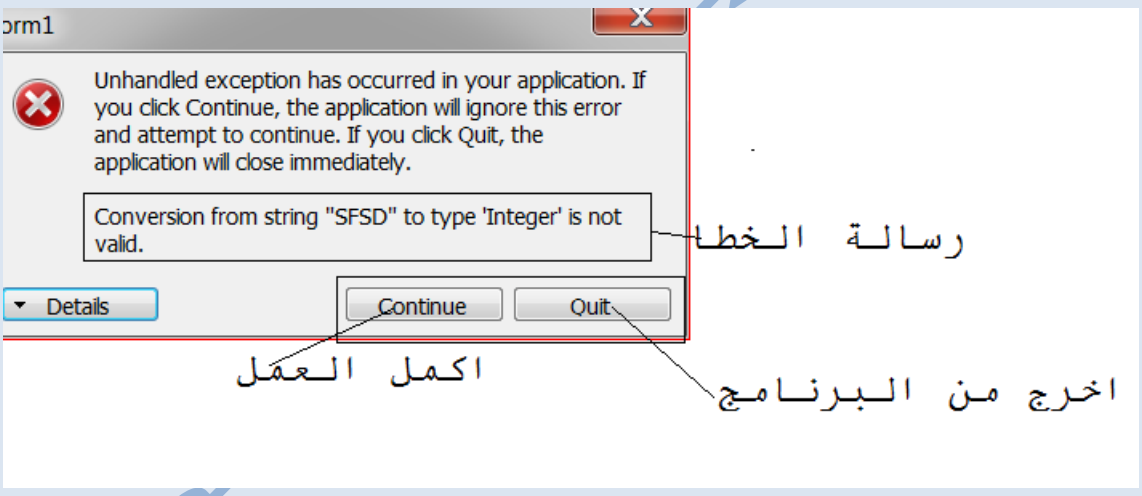


رسالة الخطأ هي الرسالة التي تعرض سبب حدوث الخطا ، بينما مقترحات لحل هذا الخطا او اذا لم تجد الحل داخل المساعدة في الفيجوال ستوديو يعرض عليك مقترح ان يخرج للموقع الاساسي لشركة مايكروسوفت msdn.microsoft.com

لو قمنا باخرج البرنامج للنور استخدمه العميل ثم حدث خطا استثناءي فانه يظهر رسالة على الشكل الاتي



ستجد انا مكونة من الاتي



المهم كيف نتفادى حدوث هذه الاخطا حتى لا تخرج هذه الرسائل المزعجة ؟؟؟

الحل في استخدام جملة try .... Catch ....Endtry

Try... Catch ....Endtry

هذه الجملة موجودة في كل لغات البرمجة من فيجوال بيسك دوت نت و C# كما توجد في لغة JAVA هذه الجملة تعمل على قنص الخطا هذه الجملة تكتب على الشكل الاتي

```
Try
الاوامر
Catch
```

الاوامر عند حدوث الخطاء

Finally

اوامر تحدث سواء كان حدث خطا ام لا

End Try

هذه الجملة تعبر مثل جملة الشرط if حيث انها تضع الاكود بداخلها بحيث انك تكتب هذه الجملة داخل العناصر مثل الاجراء و الخصائص و الدوال فعند كتابة الكود بين كلمة try و Catch تكتب الكود الذي تريده وبين Catch و Finally هي الاومر التي تحدث عند حدوث الاستثناء و الاوامر التي تكتب بعد Finally هي الاوامر التي تكتب سواء حدث خطا ام لا

ملحوظة هامة

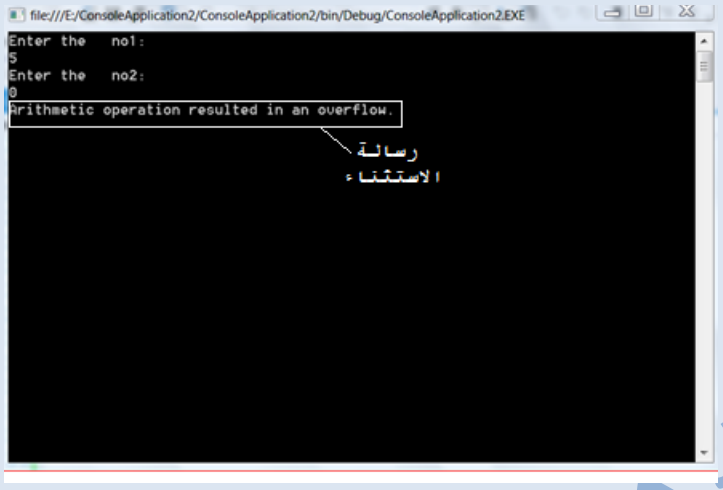
1- يمكن الاستغناء عن جملة finally فيمكن ان يكون الكود فقط try .....Catch....Endtry
2- يمكن تمرير معامل *ex As Exception* Catch_ هذا المعامل من نوع *Exception* بحيث ان *Exception* عبارة عن تصنيف يندرج من فضاء اسماء System مهمته اكتشاف الاخطاء وقنصها وهذا التصنيف يحتوي على عدة دوال اهمها الدالة massage التي تظهر نوع الخطأ وهي دالة من نوع String

مثال

افتح مشروع كونسل واكتب الكود الاتي في الاجراء main

```
Try
    Dim i, i1, i2 As Integer
    Console.WriteLine("Enter the   no1:")
    i1 = CInt(Console.ReadLine)

    Console.WriteLine("Enter the   no2:")
    i2 = CInt(Console.ReadLine)
    i = i1 / i2
    Console.WriteLine(i)

Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.Read()

End Try
```

عند تشغيل البرنامج قم باعطاء رقم 0 للمتغير الثاني ثم اضغط ادخال فسوف تجد ان الكنسل يظهر لك رسالة كما هو موضح في الشكل الاتي



بعد ان تعلمنا كيف نكتشف الاخطاء ما رأيك بعمل اخطاء ؟؟؟

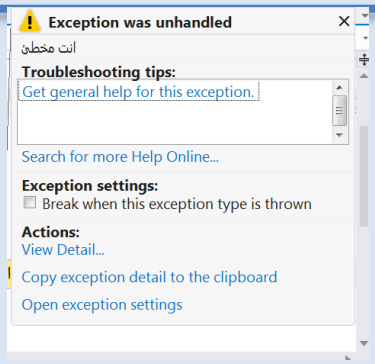
رمي الاستثناء

رمي الاستثناء ما هو الا حل لحماية الدواتك و برمجك من ان يرتكب اي احد خطا في برمجتها ... طبعا ستفتح فمك بانك لا تفهم ما اقصده ..... تخيل انك سلمت الاداة التي انشأنها لزميلك في العمل ليقوم بربطها بقاعدة البيانات الخاصة بالموظفين ثم تكتشف انه يخطى في استخدامها وتخرج قيمة المرتبات بقيمة سالب مثلا فانت لا تريده يخرج الارقام سلبية فتعطيه رسالة تحذره بعدم اخراج الارقام سلبية ....ستقول يمكن حلها باستخدام الاحداث ... طبعا الاحداث جزء من الحل ولكن يجب اننبه المبرمج في وقت التصميم حتى لا يقع في خطا فيقوم البرنامج بتنبيهه اثناء اختباره للبرنامج ولكن اذا اعتمد على الاحداث فالاحداث تحدث في وقت التشغيل فهي ستنبه المستخدم ولكننا نستخدم الاستثنائات لننبه زملائنا وهم يصمموا البرنامج و يختبروه ولكي تقوم بعمل رمي للاستثناء فانكتستخدم كلمة thraw وهي كلمة مفتاحية تستقبل اي خطا قم بعمل المثال الاتي

افتح مشروع جديد واضف زر للنموذج واكتب الكود الاتي في حدث Click للنافذة

```
Dim i As Integer = 5
Dim y As Integer = 10
Dim x As Integer
x = i / y
Throw New Exception("انت مخطئ")
```

واذا شغلت البرنامج ونقرت على الزر تظهر كل الرسالة الاستنائية الاتية



تلاحظ ظهور الرسالة الاتية لكن انصحك ان لا تكثر من استخدام الاستثناءات بشكل كبير لانها تستهلك قدر كبير من الذاكرة استخدمها وقت الحاجة الملحة فقط

حسنا.... ما رأيك ان نصمم تصنيف استثنائي مثل Exception وهذا التصنيف سيكون مشتق من التصنيف Application Exception قم الان بعمل التصنيف الاتي

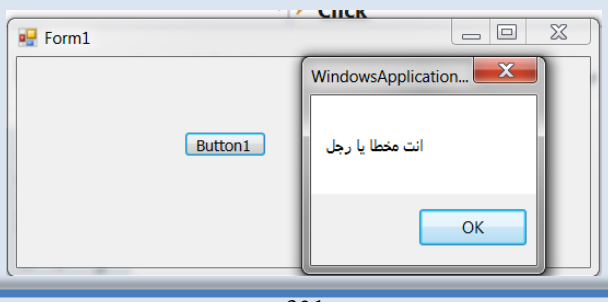
```
Public Class MYException
    Inherits System.ApplicationException
    Public Overrides ReadOnly Property Message As String
        Get
            Return "انت مخطأ يا رجل"
        End Get
    End Property
```

ثم اكتب اكتب الكود الاتي في الحدث Click للزر الذي وضعناه في المثال السابق

```
Try
            Dim i As Integer = 5
            Dim y As Integer = 10
            Dim x As Integer
            x = i / y
            Throw New MYException هنا استدعينا التصنيف الذي انشاناه
        Catch ex As MYException  >>>>هنا استدعينا تصنيفنا
            MsgBox(ex.Message)
        End Try
```

في الكود السابق لن تجد اي غريب غير اننا كتبنا كلمة thraw ثم انشانا التصنيف الذي انشأناه

فطبعااذا حدث استثناء وخطا فينتقل الى الاوامر بين Catch واخر الجملة فنكتب الدالة msgBox و التي تقوم بانشاء رسالة تحذير سوف نشرحها في الفصول القادمة هذه الرسالة تعرض رسالة الخطاء التي كتبناها كما هو موضح في الشكل الاتي عند تشغيل البرنامج و الضغط على الزر



جملة On Error

هذه الجملة لن تجدها في اي لغة الا في لغة الفيجوال بيسك فهي قادمة من الفيجوال بيسك 6 السابق وهذه الجملة تكتشف الاخطاء الاستثنائية ايضا وتكتب في اعلى الاجراء ولها نوعان

On error resum next

وهي تعني في حالة الخطا اذهب للسطر الذي يليه ولا تنفذ هذا السطر

On error Go to

كلمة Go To هي جملة التي تقوم بعمل قفز من سطر الى سطر اخر ولكي تستطيع فهمي اكثر اذهب الى اي مشروع كتبه ثم اكتب الكلمة في اي سطر في لاجراء الاتية

Start :

ثم اذهب في اي مكان اخر في الاجراء واكتب جملة Go to ثم الكلمة Statr فتصبح Go to Start

ثم في حالة التنفيذ ستلاحظ اذ اوصل المترجم الى السطر الذي فيه جملة Go To يذهب الى السطر الذي في كلمة Start

لا يشترط كلمة Start يمكن كتابة اي جملة اخرى بشرط اتباع شروط التسمية التي شرحنها في المتغيرات ثم كتباة النقطين (:)

الاخطاء المتطقية

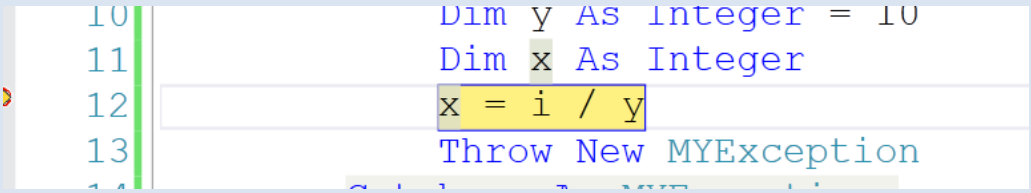
هذه الاخطاء هي الاخطاء التي تحدث ولا يكتشفها الكمبيوتر وتكتشفها البشر بحيث انك قمت بكتابة القواعد بشكل سليم و لا يحدث استثناء فتجد البرنامج يخرج نتائج خاطئة لان بصراحة الكمبيوتر معروف بانه الالة السريعة لكنها غبية لانها تنفذ كل الاوامر التي تاتيها منك فلو انك ادخلت عليها اوامر خاطئة ستنفذ الاوامر الخاطئة . ولكي نستطيع تتبع هذه الاخاطاء ونكتشفها بسهولة فقط اتاحت لك الدوت نت عدة طرق لاكتشاف هذه الاخطاء

Break Point

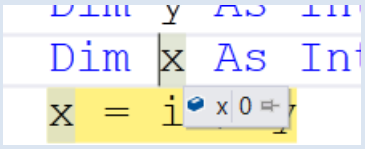
نقطة التوقف مهمتها ان توقف البرنامج على كود معين تخيل لو انك تريد ان تختبر متغير معين لمعرفة قيمته وهذا المتغير لا يعطيك النتيجة بشكل مباشر بلي بعطيها متغير اخار ثم يتم معالجة هذه القيمة وانت تريد ان تعرف القيمة قبل معالجتها فتقوم بعمل نقطة توقف على هذه النقطة ... تابع معي الشكل الاتي

```
11          Dim x As Integer
12          x = i / y
13          Throw New MYException
```

ستجد انه يقف على نقطة معين واذا شغلت البرنامج ووصل المترجم الى هذه النقطة فيتوقف تماما كما هو موضح في الشكل الاتي



ولكي تعرف قيمة المتغير قف عليه فتظهر لك النتيجة كلاتي



وانصحك عزيز القارئ ان اذا اردت معرفة قمية متغير معين ضع نقطة التوقف في السطر الذي يليه حتى تتم اعطاء القيمة للمتغير

استخدام لوحة المفاتيح

تستطيع معرفة الاخطاء واكتشافها و اختبار قيمة المتغيرات و الخصائص عن طريق الضغط على الازرارالاتية في لوحة المفاتيح

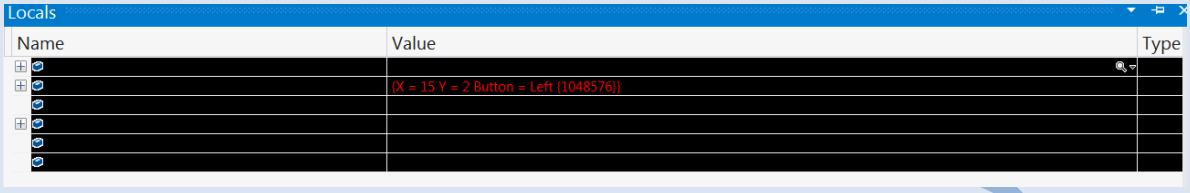
F11	هنا يقوم بعملية توقف على كل سطر بمعنى انك اوقفت البرنامج على نقطة معينة وتريد معرفة النقطة التي تليه فاذا ضغطت على f11 فانه ينتقل الى السطر الذي يليه ثم الذي ليه عند الضغط حتى تتاكد من خلوها من الاخطاء
F10	هنا يقوم بعملية توقف على كل سطر بمعنى انك اوقفت البرنامج على نقطة معينة وتريد معرفة النقطة التي تليه فاذا ضغطت على F10 فانه ينتقل الى السطر الذي يليه ثم الذي ليه عند الضغط حتى تتاكد من خلوها من الاخطاء ولكن اذا كان هناك استدعاء لاجراء او دالة او خاصية فانه ينتقل اليها ما عدا الخواص التي تستدعى من فضاء اسماء اخر فانه لا يسمح له بالدخول عليها وقفا لمبدا الكبسلة
F5	هو زر تشغيل البرنامج لو اردت تشغيل البرنامج بعد التوقف

مراقبة المتغيرات

في السطور السابقة قلنا انك تريد ان تختبر المتغيرات او العناصر فانك تستخدم نقطة التوقف او ازرار لوحة المفاتيح ولكنك تستطيع متابعة المتغيرات عن طريق نوافذ مراقبة وهي watch , locals , immediateوهذه النوافذ يمكن الوصول اليها اذا لم نجدها في بيئة البرمجة عن طريق القائمة Debug ثم الاختيار Window فتخرج منه قائمة فرعية فيها كل نوافذ المراقبة

Locals

مهمة هذه النافذة هي تحديد قيمة المتغيرات المحلية المعلن عنها داخل الاجراء او الدالة او الخاصية واظهار قيمتها كلمتا ضغطنا على الزر F11 من لوحة المفاتيح للتنقل سطر سطر ومع كل سطر تلاحظ تغير القيم للمتغيرات اثناء التنقل من خلال النافذة كما هو موضح في الشكل الاتي



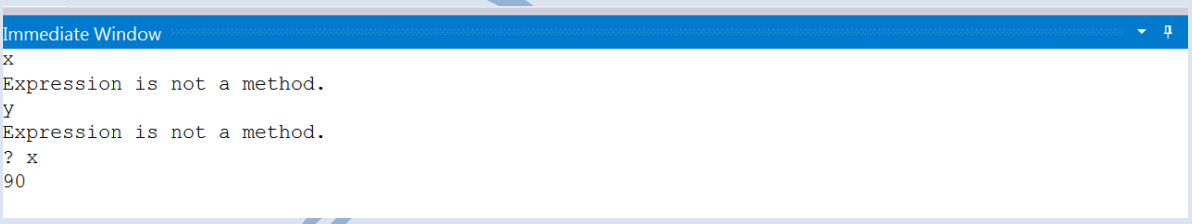
كما تستطيع ان تختبر اي متغير عن طريق تغيير قيمتها وتجبرها عن طريق النقر على مفتح F11 لمعرفة النتيجة بعد ذلك

Watch

هذه النافذة مثل سابقتها في مراقبة المتغيرات لكنها تكتشف المتغيرات العامة وغير المحلية وطبعا باستخدام المفتاح f11 او نقطة التوقف فهي مثل سابقتها

Immediate

لو انك تريد ان تختبر متغير معين فانك تكتب قيمة المتغير على النافذة والضغط على مفتاح enter فانك تكتشف قيمته وعند النقر على مفتحا F11 فانك تكتشف انه يتغر قيمته كما هو موضح في الشكل الاتي



حيث انه يجب عليك كتابة علامة استفهام قبل كتابة المتغير هكذا ? VaribleName

مع العلم ان قمية المتغير تتغير عند النقر على المفتاح F11

نوافذ اخرى

نستعرض سريعا بعض النوافذ الاخرى التي نستخدمها في اكتشاف الاخطاء

Autos	مهمتها كشف المتغيرات وتوظيح قيمتها وتعمل عند وقت التوقف يعني عن نقطة التوقف او الضغط على F11
BreackPoint	تعرض لك قائمة بكل نقاط التوقف
Call Stack	هي تعمل عند وقت التوقف وهي تتبع الاكواد المصدرية بمعنى انك

	تفحص اجراء معين به خطا فتقوم باسخدام هذه النافذة معرفة متى تم استدعاء هذا الاجراء وكيف تم استدعاءه وذلك النقر مرتين على النافذة او النقر على الزر الايمن وختيار الاختيار Go to Source code
Memory	تستعرض ما في الذاكرة

هذه اهم النوافذ التي نستخدمها في اكتشاف اخطانا

الى هنا نكون قد نتهينا من شرح مبادئ البرمجة و صناعة الادوات مهمتك هي قراءة الكتبا مرة اخرى وتطبيق ما شرحنها جيدا ثم ننتقل الى الفصل الاتي الذي يعتبر ملحق بالكتاب بحيث يحتوي على كثير من الدوال و التصنيفات التي ستساعدك في فهم كثير من الاكواد التي تصادفك اثناء رحلتك العملية .

الفصل الحادي عشر

اهم التصنيفات و الدوال

انا ما اخدتهاش في الكلاس هذه الكلمة التي ذكرتها في المقدمة الهمتني ان اقدم لاعزائي القراء بعض النقاط الهامة جدا التي تجعل منهم مبرمجين محترفين ،هذه النقاط سوف تساعدك باذن الله في حياتك العملية بحيث ان فيها مفاتيح كثيرة لمن يريدون تطوير انفسهم فيقوموا بالتصفح عبر الانترنت ويفتحون المواقع العربية او الاجنبية للبحث عن معلومات اضافية لتطوير مهاراتهم فقد يقع في ايديهم كثير من الاكود تجعلهم لا يستطيعوا فهمها ، لذلك فقد قررت ان اقدم لهم اهم المفاتيح التي تساعدهم في فهم كثير من الاكواد التي تستعرض في هذه المواقع العالمية وسوف نستعرض في هذا الفصل باذن الله اهم التصنيفات الموجودة في الNetFramWark و التي دائما ما نستخدمها في البرمجة لتسهل على كثير من المبرمجين المبتدئين لكنهم جاديون يريدون تطوير انفسهم ويريدون التفوق في هذه المهنة فيجدوا انفسهم حائرين امام كثير من هذه التصنيفات ، ولذلك اطلب منك ان تركز معي في هذا الفصل حتى تستوعب هذه التصنيفات

SYSTEM.OBJECT

التصنيف هو object وهو تصنيف مسؤول عن التعامل مع البيانات الاساسية وللعلم ان كل التصنيفات التي في اطار العمل FramWark مشتقة من هذا التصنيف فهو يعتبر التصنيف الاب في البرمجة تحت بيئة الدوت نت

دوال التصنيف object

سوف نستعرض اهم الدوال الاساسية في التصنيف object منها وللعلم ان هذا التصنيف مجرد يعيني MustInhirts فلا يمكن انشاءه باستخدام الدالة new

Equal	هذه الدالة من نوع Boolen تستخدم عمل في المقارنة بين الكائنين بحيث انه يقارن بين الكائنين هل هما الكائنين من نفس النوع فيعطي True ولا يعطي False فلو كان لدينا كائنين A,b, ونريد ان نعقد المقارنة فاننا نكتب الكود الاتي If b.Equals(A) then هذه جملة شرط تختبر اذا كان الكائن b هو نفسه الكائن A
RefranceEqual	هي مثل الدالة السابقة لكنا غير قابلة لاعادة القيادة NotOverridable و دالة مشركة Shared فتجد في هذه الدالة معاملين وليس معامل واحد حتى تعقد المقارنة بينهما فيكون الكود كلاتي If object.RefrenceEquals(b,A) then

ToString	كنا استخدمنها في تحويل اي قيمة الى قيمة نصية

المتغيرات ذات مرجعية و متغيرات ذات قيمة

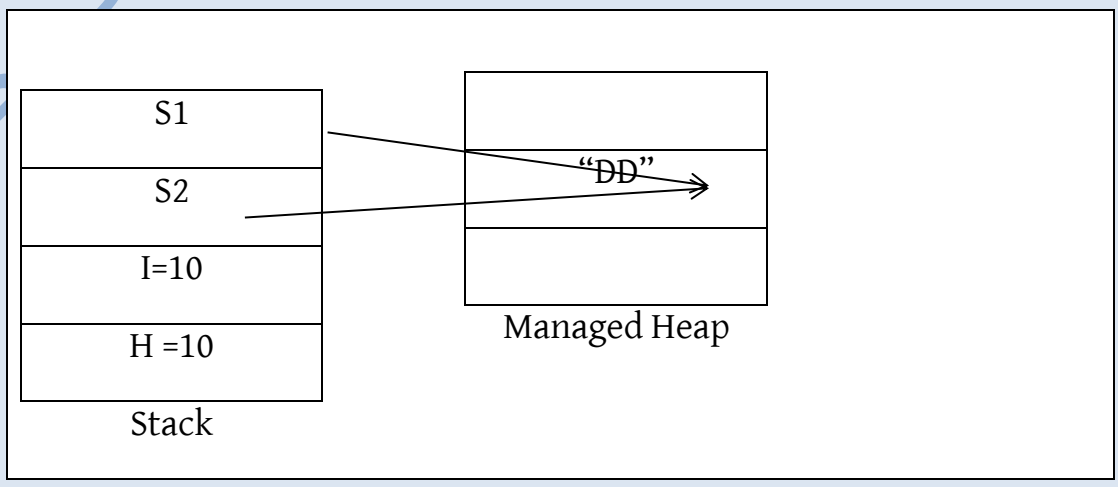
في هذا الجزء سنشرح شيء مهم جدا جدا جدا ما هي المتغيرات المعروفة بـValue Type متغيرات بقيمة و ما هي المتغيرات Refrance Type ..... خلاصة القول كل ما ساذكره الان هي النوع ذات مرجغية وغير ذلك هو هو متغير مرجعة فالمتغيرات الرقمية بكل انواعها التي ذكرناها في الفصل الاول و نوع Boolen وكل الهياكل Structure و كل Enum و Charهذه المتغيرات تحمل قيمة وغير ذلك هي متغيرات مرجعية يعني Refrance type

السؤال لماذا اشرح ذلك ؟؟ اريد ان ارجعك للفصل الاول بحيث اننا كنا نشرح الذاكرة على انه مسودة على شكل مربعات واذا اردنا اسناد قيمة فاننا نحجز مربعات بعدد البايت هذه الفكرة فعلا صحيحة ولكن اريد ان اشرح ذلك بالتفصيل قليلا

اريد ان اوضح لك ان لدينا قسمين في الذاكرة قسم يسمى Stack و قسم managed Heap القسم الاول وهو Stack يقوم بتخزين المتغيرات ذات قيمة و اسماء المتغيرات ذات مرجعية اما القسم Managed heap فهو يخزن القيم التي تشير اليها المتغيرات المرجعية !!!! اذا كنت لا تستطيع فهم الفقرة السابقة تابع معي الكود الاتي

```
Dim s1 As String = "DD"
Dim s2 As String
s2 = s1
Dim i As Integer = 5
Dim h As Integer
h = i
```

لو راينا ماذا يحدث في الذاكرة تابع معي الشكل الاتي



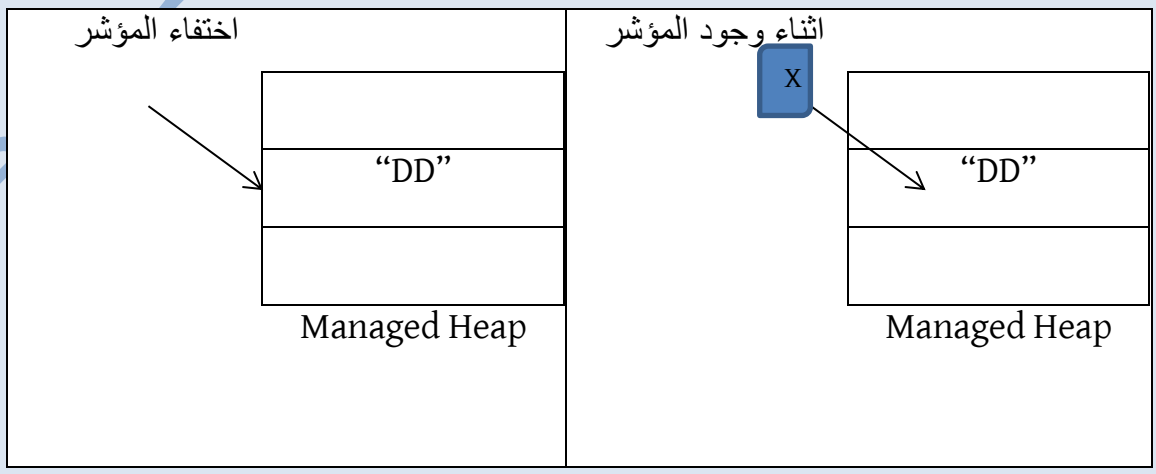
تلاحظ من الشكل السابق ان المتغيرين S1,S2 هما متغيرين من نوع String يعني متغيرين مرجعيين فيتم حجز لهم مكان في القسم Stack كاسم لهم بينما القيمة يكون لها مكان في القسم Managed Heap بينما المتغيرين i , h فهما من نوع integer لذا هما متغيرين قيمة Value Type فيكون لهم مكان في الذاكرة Stack

ولكن اوضع قليلا سوف نعقد مقارنة بين المتغيرات ذات قيمة ومتغيرات مرجيعة

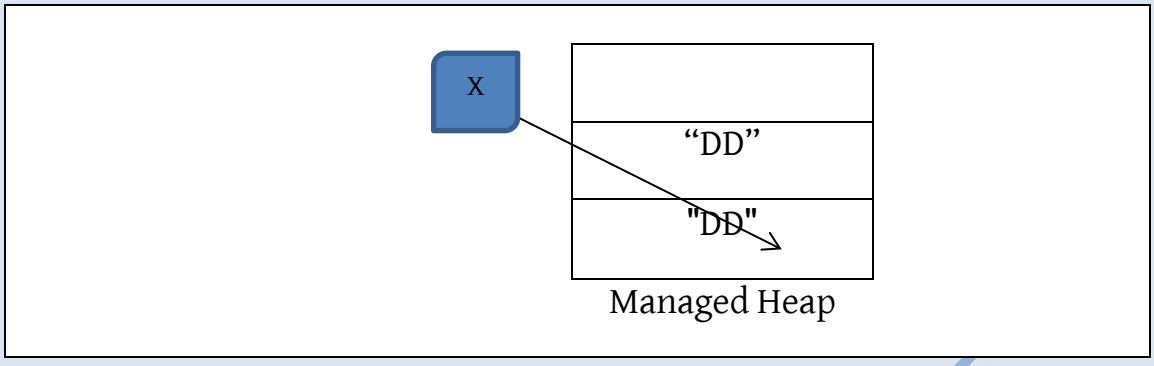
Vaue type	Ref type
المغيرات ذات قيمة يخزن المؤشر (اسم المتغير)و القيمة في القسم Stack	المتغيرات المرجعية يخزن المؤشر(اسم المتغير) في القسم Stack و القيمة في القسم managed heap
المتيرات ذات قيمة مشتقة من تصنيف System.ValueType	المتغيرات ذات مرجعية مشتقة من التتصنيف الاساسي وهو System.object
المتغيرات ذات قيمة يتم الوصول اليه بسرعة	اقل سرعة للوصول اليها
لا نحتاج للتصنيف Garbage Collection لافراعها من الذاكرة	نحتاج للتصنيف Garbage Collection لافراعها من الذاكرة

GARBAGE COLLECTION وطرق ادارة الذاكرة

السؤال الذي يطرح نفسه ما هي Garbage Collection هو تصنيف مسؤول عن افراغ القيم الموجودة في الذاكرة ولكنه متخصص لافراغ القيم الموجودة في القسم manege Heap في الذاكرة فقط فمهمة هذا التصنيف هو افراغ الذاكرة عندما يختفي الموشر الذي يخصها من القسم Stack فاذا اختفى المؤشر تظل القيمة موجودة في القسم manage heap بل لو تم انشاء كائن اخرى فانه تزيد في الذاكرة كما هو موضح في الشكل الاتي



وعندما استدعيت الكائن من نفس النوع ستظهر الذاكرة على الشكل الاتي



كما ترى يتم انشاء كائن في مكان اخر في الذكرة فاذا استمر ذلك تسوف تجد ان الذاكرة قد امتلات ويؤدي ذلك الى توقف البرنامج لان الذاكرة لا تسمح وهذا قد شكل عبئ كبير بالنسبة للمبرمجين لذا تعين على مطوري مكتبة netFramWark انشائ التصنيف Garbage Collection الذي ينظف الذاكرة اول باول

قبل ان ننتهي من ذلك سوف اعطيك طريقة لادارة الذاكرة قسم Stack عندما تريد ان تعرف ما هي اكبر قيمة مسموح بها في الذاكر قم بكتابة الدالة minValue لاختيار اقل قيمة و maxValue ولكن يجب ان تعرف ان هذه الدوال مشتركة يعني يجب استدعاء النوع ثم كتابة الدالة مثل هذا الكود

Double.MaxValue

الدوال تتعامل مع المتغيرات النصية

البيانات النصية تعتبر من اهم الادوات في البرمجة فالمشكل النصية تعتبر من اهم المشاكل التي تهم كثير من العملاء بحيث اننا نتسخدمها في التحقيق من صحة البيانات الواردة و التي يدخلها المستخدم وتخزن في قاعدة بيانات وقد يقع المستخدم في اخطاء كثيرة في ادخال البيانات مما يترتب عليه مشاكل كثيرة اثناء استدعاء البيانات فاذا كانت لديك فركة عن قواعد البيانات و تعلمت جمل الاستعلام SQlستجد ان من اهم المشاكل التي نعاني منها مع مدخلي البيانات لان مدخل البيانات وهو تحت ضغط العمل قد يدخل بيانات خاطئة في قاعدة البيانات فمثلا ينسى ان يدخل بيانات الرقم القومي كاملة او يخطئ في كتابة التاريخ وغيرها من الاخطاء الشئعة عند مدخلي البيانات وعند اسدعاء هذه تظهر مشاكل كثير حيث انك تستخرج بيانات خاطئة نتيجة خطا في الادخال بدون عملية تنقيح وتصحيح البيانات الواردة لقاعدة البيانات لذا فان التعامل مع البيانات النصية مهمة جدا في تصحيح البيانات او مراجعتها قبل ادخالها

بعد هذه المقدمة الطويلة تعالى ناخذ جولة صغيرة في التصنيف String ، هو تصنيف مشتق من التصنيف الاب Object وهذا التصنيف لا يورث NotInhertable ولكن عند الانشاء فان له ثلاثة دوال انشاء وهي

Strang(Char())	هذه مجموعة مصفوفة من نوع Char يعني حرف واحد فمصوفة بها مجموعة عناصر كل عنصر بحمل حرف واحد ليتم تجميعها
String(Char(),	هي نفس المعامل ولكن في المعامل الثاني startفبه معامل رقمي يحدد

start as int , length as int	نقطة البداية في النص و المعامل الثالثlength يحدد اقصى عدد مسموح به في عدد احرف النص
String(char , integer	المعامل الاول يحدد حرف معين و المعامل الثاني يحدد عدد مرات تكراره

ننتقل الان الى الدوال و الطرق فلدينا نوعين من الدوال

دوال الحالة

هي الدوال التي تجده من اي متغير نصي

Indexof	هي دالة مسئولة عن بحث عن نص صغير داخل المتغير النصي الذي يحمل نص كبير يمعنى لو ان لنا متغير X يه القيمة "مصر ام الدنيا" ونريد ان نبحث عن "ام الدنيا" فنكتب الكود الاتي Dim y as integer Y = x.indexof("ام الدنيا") فيتم تخزين مكان اول حرف في النص في المتغير y لون كان مكان اور حرف وهو (ا) من كملة "ام" 5 فان قيمة المتغير y ستكون 5 مع العلم انك وانت تحسب مكان اي حرف داخل النص تاخذ في اعتبارك انك تعد ايضا المسافات فالمسافة خرف له رقمه في النص
Insert	هذه الدالة مهمتها هي ادخال نص داخل النص الكبير الذي يحمله المتغير و الدالة بها معاملين كما هو موضح في الدالة Insert(string,start as integer) معامل نصي String بحيث نرفق فيه النص الذي نريد ادخاله و معامل start رقمي بحيث نحدد من خلاله مكان النص Insert(string,start as integer)
Remove	هذه الدالة تمسح جزء من النص ويوجد منها دالتين الاولى Remove(start as int) Remove(start as int , long as integer)

	Start يحدد بداية الحذف Long يحدد طول النص الذي سيحذف
Replace	هذه الدالة تقوم باستبدل نص مكان نص اخر وبها اختيارين من الدوال Replace(string, string) المعامل الاول النص الذي نبحث عنه ليتم استبداله و المعامل الثاني النص الذي نريد ان نضعه Replace(char, char) بينما دالة الثانية نفس المعاملات لكن نوعها Char

الدوال التي من Visual Basic 6

هنا من كرم مايكروسوفت علينا بانها قامت بارفاق الدوال التي كنا نستخدمها في فيجوال بيسك 6 الى الدوت نت واليك هذه الدوال ولكن هذه الدوال نقوم باسدعائها بشكل مباشر بدون الاستعانة بمتغير يعني مثلا

```
Dim s as integer
S = instr(.....)
```

Insrt	هذه الدالة مثل indexof بحيث انها دالة رقمية تحدد مكان النص وبها نوعان هما Inst(str1,str2,[compar] Inst(start as int ,str1,str2,[compar] سنجد ان امعاملات متشابهة بحيث انك في المعاملٍ str1 يحدد النص الذي يتم فيه البحث و Str2 النص الذي نبحث عنه في Str1 اما المعامل Start فهو يحدد بداية البحث اما [compar],هو معامل اختياري يحدد نوع النص الذي نبحث عنه فيه ومعامل اختياري يعني يمكن ان تتركه بدون التعديل في قيمته
Trim	هذا دالة نصية تقوم بعمل افراغ للمسافات في النص اول النص وفي اخره ويتم استدعائها كالتي Dim s as string = “ hi World ” Dim x as string

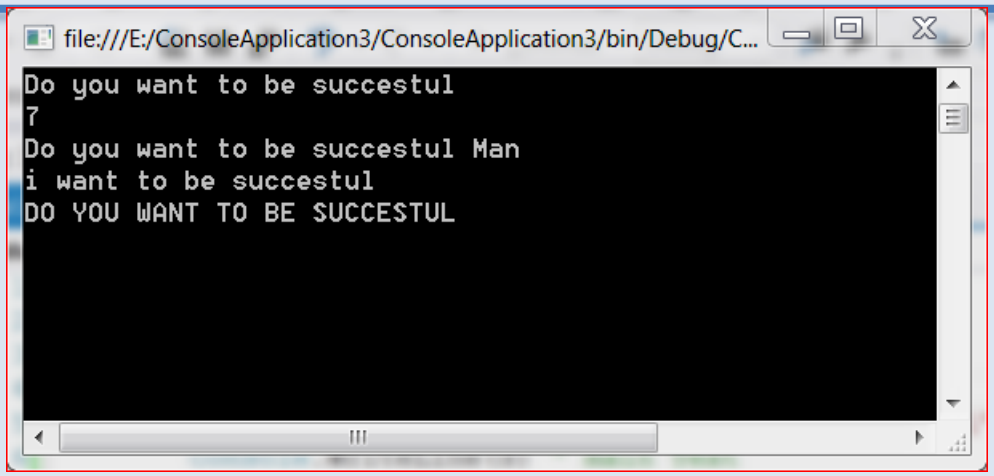
	X = trim(s) تلاحظ وجود مسافة في او النص واخره وعند استخدام الدالة يكون الناتج "hi World"
Mid	هذه الدالة تنسخ جزء من النص وهذه الدالة بها نوعان هما Mid(string, start as int) Mid(str,start as int ,leng as int) حيث ان Str يحدد النص الذي نريد انسخ منه Strat يمثل بداة النسخ كما يجب ان نوضح ان النوع الاول يقوم بتحديد نقطة البداية ثم ينسخ الى اخر النص اما النوع الثاني فيه المعامل الرقمي length بحيث يحدد حد معين
Ucase	هذه الدالة النصية تحول الاحرف الصغيرة للكبيرة
lcase	هذه الدالة النصية تحول الاحرف الكبيرة للصغيرة
len	هذه الدالة تحدد طول النص

مثال

افتح كنسل جديد وكتب فيه الكود الاتي

```
Dim s As String = "Do you want to be succestul"
Console.WriteLine(s) ' main Text
Console.WriteLine(s.IndexOf("want")) 'بحث علن نص
Console.WriteLine(s.Insert(s.Length, " Man")) 'ادخال نص
Console.WriteLine(s.Replace("Do you", "i")) 'استبدال
Console.WriteLine(UCase(s)) 'تحويل حروف كبير لحروف صغيرة
Console.ReadLine()
```

وعند التشغيل تظهر النافذة كلاتي



حاول ان تجرب باقي الدوال على هذا المثال

دوال التعامل مع المتغيرات التاريخية

لكي تعرف كيف تتعامل مع التاريخ افتح كنسل جديد وفي الدالة Main اكتب الكود الاتي

```
Dim d as date = #02/11/2011#

Console.Writeline(d.Tostring)
```

سيكون الناتج كلاتي

02/11/2011 12:00:00 Pm

حيث ان التاريخ يكتب باتباع الصيغة الاتية MM/ dd/yyyy hh:mm:ss tt

هذه الصيغة تعني الاتي

MM هذه الحروف كبيرة تعطيك رقم الشهر بالنسبة للسنة واذا اردت اسم الشهر فبدل الصيغة بحيث تكون MMMM في يصبح الشهر بدال ما يكون 01 سيكون يانايير

dd هذه الصيغة تعني اليوم بحيث انه يعطي اليوم بالنسبة للشهر ولكن اذا كنت تريد اسم اليوم في الاسبوع فتكون الصيغة كلاتي dddd فتعطيك اسم اليوم في الاسبوع فبدل ما يكون 25 سيكون يوم الثلاثاء

yyyy هذه الصيغة تعطيك السنة بالرقم كامل يعني 1985 ولكن اذا اردت الرقم 85 فعدل الصغية بحيث تكون yy

hh هذه الصيغة تعطيك الساعة بالاثنى عشر من 1 الى 12 واذا اردت حساب الساعة بحيث تكون رقم الساعة في اليوم ب24 يعني الساعة من 0 الى 23 فتكون الصيغة HH حروف كبيرة

ss تعني الثواني

mm الدقائق ويجب ان يكون حرف الm حرف صغير

tt تحدد الساعة الاثنى عشر صباحة ام مساءا Am او Pm

ولكن لا يشترط ان تكتب الصيغة كاملة فيمكن ان تجتزء جزء منها وتعدل فيه كما تريد مبعنى

لو اننا كتبنا الصيغة الاتية M/d/yy فيكون التاريخ 2/11/11حيث اننا في شهر فيبراير في اليوم الحادي عشر في السنة 2011

فلو عدلنا الصيغة الى dd/MM/yyyy فتظهر في التاريخ الاتي 11/02/2011

مع العلم ن الصيغة تختلف من بلد لاخر فنحن في مصر يكون اليوم قبل الشهر اما في الولايات المتحدة يكون الشهر قبل اليوم

كذلك الامر في الساعات فلو كانت الصيغة HH:mm فيكون الوقت 22:44 الساعة العاشرة مساء و 44 دقيقة

فلو كانت الصثيغة هكذاtt hh:mm:ss فتكون الوقت 11:44:33PM الساعة الحدية عشرة مساءا و 44 دقيدة و 33 ثاية

عرفنا من ما سبق الصيغة التي تكتب التاريخ ولكن كيف نجزئ التاريخ بحيث اننا لا نريد ان يتم عرض التاريخ بالصيغة كاملة ..... هنا سنستخدم الدالة Format وهي دالة نصية تحول التاريخ الى نص يكتب الصيغة التي نريد ان يظهر من خلالها التاريخ

الدالة Format

دالة نصية تستخدم في تعديل شكل التاريخ ليتم تشكيلة بالشكل و الطريقة التي نريدها ولكن يجب اتباع الصيغة التي شرحناها سابقا ولكي تستطيع فهم هذا الكلام قم بتعديل الكود السابق ليكون كلاتي

```
Dim d As Date = #2/11/2011#
Console.WriteLine(Format(d, "dd/MM/yy"))
 Console.ReadLine()
```

في الكود السابق كتبنا التاريخ في السطر الاول اما في السطر الثاني استخدمنا الدالة Format و التي يكون صيغتها القياسية كلاتي Format(object , String)

حيث ان المعامل الاول objectيدخل من خلاله قيمة التي نريد ان نعدل في شكلها سواء اكانت رقمية او تاريخية وفي هذا المثال نستخدم قيمة تاريخة

اما في المعامل الثاني String نستخدم الصيغة التي تقوم بتشكيل القيمة التي في المعامل الاول وعند تشغيل البرنامج سيكون الناتج **02/11/11**

ننتقل الان الى اهم الخصائص التي توجد في التصنيف Date

Date	التاريخ فقط	Date
TimeSpan	الوقت فقط	TimeOfDay
Integer	شهر	Month

Day	يوم في الشهر	Integer
Houre	ساعة (0-23)	Integer
Menite	دقيقة	Integer
Second	ثانية	Integer
Millisecond	مللي ثاينة	Integer
Dayofweek	يوم في الاسبوع 0-6	Integer
dayofyear	يوم في السنة 1-366	Integer
Now	التاريخ الان ولكن هذه الخاصية مشتركة Shared	Date
toDay	لاعطاء التاريخ اليوم وهي خاصية مشتركة	Date

مع العلم ان جميعها خصائص للقراءة فقط readonly property

دوال التعامل مع التاريخ

في السطور القادمة سوف اشرح دالتين مهمتين للتعامل مع التاريخ ودائما ما نحتاج اليها وهما

DateAdd و DateDeff

DateAdd

هي دالة من نوع Date تاريخ مهمتها هي اضافة تاريخ للتاريخ الذي معنا يعني مثلا نفترض اننا نريد ان نعرف التاريخ الذي سيتم فيه سداد باقي المبيعات و المدة الباقية هي 55 يوم فنستخدم هذه الدالة التي تكون صيغتها كلاتي

DateAdd(*DateInterval*,*Double*,*Date*)

DateInterval هو معامل Enum بحيث يحدد من خلاله نختار ما النوع الذي نريد ان نغيره هل نضيف القيمة الى اليوم او شهر او يوم

Double تحدد القيمة التي نريد اضافتها

Date التاريخ الذي نريد تغييره

نفترض اننا نريد معرفة متى سيسدد العميل الدفعة القادمة من القسط مع العمل ان القسط سيكون بعد 55 يوم و اليوم هو 2012/12/24 هنا نقوم سنكتب الكود الاتي

```
d = DateAdd(DateInterval.Day, 55, d)
```

حيث ان d هي متغير من نوع Date يحمل قيمة التاريخ الذي نريد تغييره

DateDiff

هذه الدالة دالة رقمية تحسب الفرق الزمني بين تاريخين مثال حساب عمر شخص ما فنستخدم الدالة التي صيغتها القياسية هي DateDiff(DateIntervale, date1 , date2)

لحساب عمر شخص ما سوف نعلن عن متغير لتخزينه تاريخ الميلاد و متغير به تاريخ اليوم ثم تغير رقمي نخزن به العمر الذي سيستقبل البيانات من الدالة DateDeff كما هو موضح في الكود الاتي

```
Age= DateDiff(DateInterval.Year, d, d1)
```

حيث ان Age متغير رقمي يحمل التاريخ اما d متغير من نوع تاريخ يخزن تاريخ اليوم اما d1 يحدد تاريخ الميلاد

تحويل التاريخ الميلادي للتاريخ الهجري

احببت ان اضيف هذه الفقرة لاحبائي القراء في بلاد الحرمين حيث البلاد تعتمد في التقويم على التقويم الهجري فالتاريخ الهجري يتم اعتماده في الدوت نت فيمكنك ان تتغير التاريخ الميلادي للتاريخ الهجري الطريقة الاتية افراها ثم ادعو لي وانت في العمرة باذن الله

اولا يجب عليك استيراد فضاء الاسماء Globalization من الفضاء الاساسي system

ثم نكتب الكود الاتي

```
Dim Hijra As DateTimeFormatInfo
Dim cal = New CultureInfo("ar-sa")
Hijra = cal.DateTimeFormat
Console.WriteLine(DateTime.Now.ToString("dd/mm/yyy", Hijra))
Console.ReadLine()
```

في الكود السابق قمنا بعمل الاتي

في السطر الاول اعلنا عن متغير من نوع DateTimeFormatinfo

في السطر الثاني اعلنا عن كائن من نوع CultureInfo وعند انشاء الكائن يجب ان نضع رمز ar-sa

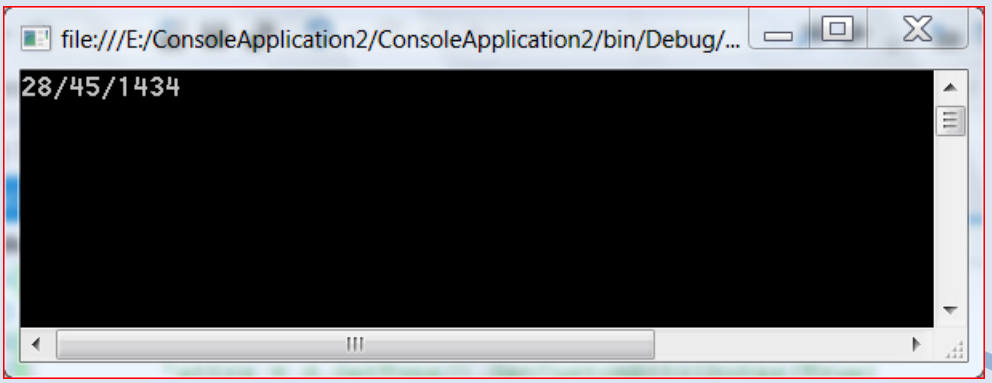
وهو الرمز يوضح اننا سوف نستخدم التقويم السعودي

في السطر الثالث اعطينا قيمة dateTimeFormat التي الكائن Cal الى المتغير hijra

في السطر الرابع يقوم الكنسل بعرض التاريخ ولكن تلاحظ اننا استخدنا الدالة Tostring التي في التصنيف DateTime حيث ان هذه الدالة بها خيارات overload واحد من هذه الخيارات به معاملين هما Format as string , وهذا معامل نقوم بوضع تنسيق التاريخ مثل الذي كنا نضعه في الدالة Format

المعامل الثاني نضع فيه المتغير hijra حيث انه يستقبل قيمة المتغير ويجعل التاريخ هجري

وعند تشغيل البرنامج يظهر كل التاريخ الهجري



اما لو تريد ان تدخل الرمز المصري عند انشاء الكائن Culture سيكون "Ar - eg"

و الجدول الاتي يوضح رموز الدول العربية_

السعودية	Ar-Sa
مصر	Ar-eg
البحرين	Ar-bh
الامارات	Ar-ae
العراق	Ar-iq
لبنان	Ar-lb
الاردن	Ar-jo
السودان	Ar-Sd
تونس	Ar-tn
ليبيا	Ar-ly
الجزائر	Ar-dz
المغرب	Ar-ma
الكويت	Ar-kw
قطر	Ar-qa
سوريا	Ar-sy
اليمن	Ar-ye

عمان	Ar-om

واليك بعض رموز الدول الاجنبية

الولايات المتحدة	en-us
بريطانيا	en-gb
اليابان	ja
استراليا	en-au
الصين	zh-cn الصين نفسها zh-hk هونج كونج zh-tw تايوان
ايطاليا	It
فرنسا	fr
اسبانيا	En –au

تنسيق الارقام

كما قلنا ان الدالة Format تنسق التاريخ فهي تنسق ايضا الارقام عن طريق الرموز الاتي

#	يمثل رقم او مسافة خالية
0	يمثل رقم او صفر
.	علامة غشرية
,	فاصلة في الالف
E	يعرض الرقم في صورة اسية
%	نسبة المائة

ساتركك تجرب بنفسك المثال فهو سهل جدا ولكن يجب ان تعرف الفرق بين العلامة # و 0 فالعلامة # تمثل رقم فمثلا لو قمت بادخال الرقم الاتي 12354.255 ثم جعلت التنسيق كلاتي ###.## فيكون الناتج كلاتي 12354.25 بينما لو

كان التنسيق 0##.## سيكون الناتج 12354.250 فلو جعلنا التنسيق كلاتي 00.00 00 سيكون النتاتج 123 54.25
بمعنى انه يحسب عدد الرموز التي بجانب الفاصلة سواء كان يمين ام يسار فلو كان الرموز القريبة للفاصلة 3 فانه يحد ثلاثة ارقام بعد الفاصلة ثم يعطي المسافة ثم يعطي باقي الارقام

دوال اختبارية

هي دوال لاختبار البيان الذي يتم ادخال قيمته

IsNumeric	الارقام هل القيمة الداخلة تمثل رقم ام لا فاذا كان رقم كانت القيمة Trueوالا تكون False
IsDate	هل القيمة التي في المعامل من نوع تاريخ
IsReferance	هل القيمة من تصنيف مشتق من التصنيف الاب Object
IsArray	هل القيمة الواردة عبارة عن مصفوفة ام لا

اريد ان انبه ان كل الدوال السابقة بها معامل واحد من نوعobject

الانعكاسات REFLICTION

تعلمنا في الفصول السابقة ان نقوم استدعاء المتغيرات او الخصائص نقوم بتخصيص قيمة لها او قراءة منها قيمة وكل ذلك في وقت التصميم ..... ما رايك في ان تستعدي متغير او خاصية في وقت التشغيل Run Time وتخصيص قيم له بمعنى انك تكتب على صندوق النص اسم متغير وتنقر على زر فيتم استدعاءه ثم نخصص له قيمة .... هذا ما يسمى بالانعكاس

والانعاكس او Refliction هو فضاء اسماء من الفضاء الاسماء الاساسي System وهو متخصص في هذه المهمة لذا يجب عليك ان تستدعيه بكلمة Imports

```
Imports system.Refliction
```

كما يوجد تصنيف اخر يسمى Type لديه نفس المهمة وهو ما سوف نتناوله الان

التصنيف Type

هذا التصنيف من الفضاء الاسماء الاساسي System ومع ذلك فهو مثل فئات الانعكاس فهو يتعامل مع البيانات الساسية من بيانات رقمية و بيانات من نوع Boolen كما يتاعمل من بيانات المشتقة من التصنيف Objict و التي تصمم التصنيفات التي نعرفهاClass ..... End Class .

هذه التصنيف لا يمكن ان ننشئ منه كائن باستخدام الدالة new ولكن نستدعيه عن طريق استخدام الدوال اخرى اهمها GetType و التي من تصنيف اخر اسمه Assebly ولكن وانت تتعامل مع اي كائن ستجد هذه الدالة من ضمن الدوال التي تستخدمها .... لكي تستطيع استيعاب ما قلته سابقا تابع مع المثال الاتي

اكتب الكود الاتي في الدالة main

```
Dim t As Type
t = GetType(Integer)
Console.WriteLine(t.FullName)
Console.ReadLine()
```

فيكون الناتج

System.int32

تعالى معي ناخذ جولة عن دوال و خصائص التصنيف Type وسوف نبدا الخصائص وليكن في علمك انها خصائص للقراءة فقط

Name	يحدد اسم التصنيف الذي نختبه
FullName	يحدد اسم التصنيف ولكن يحدد مسارة من بداية فضاء الاسماء الاساسي فلو كان التصنيف مثال Graphics فيكون الناتج System.Drowing.Graphics
MemberType	هي تعود بنوع العنصر من حيث انه متغير او خاصية او طريقة(اجراء ام دالة)

MemberInfo

هذا التصنيف من فضاء الاسماء Refliction وهو مشتق من التصنيف Type ومهمته هي ان يتعامل مع العناصر التصنيف من دوال و اجرائات و خصائص و احداث و متغيرات كما يشتق من هذا التصنيف تصنيفات اخرى تتعامل مع كل عنصر على حدة

FieldInfo	للتعامل مع المتغيرات
PropertyInfo	للتعامل مع الخصائص
MethodInfo	للتعامل مع الدوال

EventInfo	للتعامل مع الاحدث

ولكي يتوضح الرؤيا اكثر تعالى معي تطبق هذا المثال اكتب الكود الاتي في الاجراء main

```
Dim t As Type
t = GetType(Integer)
Dim m As MemberInfo
For Each m In t.GetMembers
Console.WriteLine(m.Name)
Next
 Console.ReadLine()
```

في الكود السابق اعلنا عن متغير من نوع Type و في السطر الثاني اخذنا نوع Gettype(int) واعطيناه للمتغير T

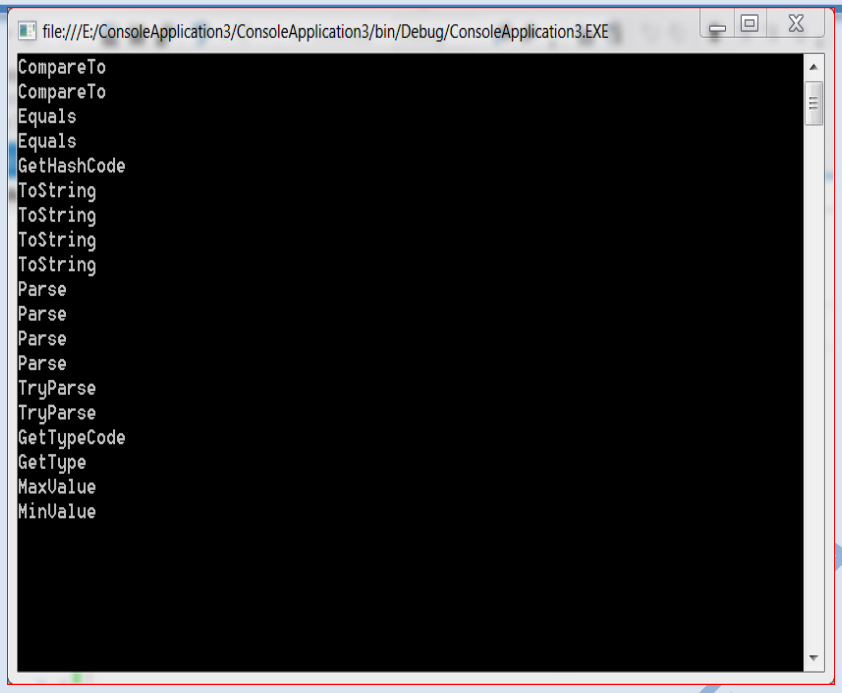
اما المتغير m هو memberinfo وهي سوف تكتشف كل الاعضاء التي في اي تصنيف

اما الجملة For each هي جملة تكرارية مهمتها ان تتعامل مع اي عنصر داخل مجموعة

بحيث انها تبحث عن عناصر من داخل التصنيف integer ثم تعطيه للمتغير m ومن المتغير m سوف نتعامل معه داخل الحلقة التكرارية حيث اننا في الحلقة طلبنا من الكنسل ان يكتب اسم كل عنصر من العناصر التي يستقبه المتغير m

وسوف اشرح هذه الجملة بشكل تفصيلي في اخر هذا الفصل

فلو شغلنا البرنامج سيكون البرنامج على الشكل الاتي



تلاحظ انه الجملة تستقبل عنصر علئص فتعطيه للمتغير m فيكتب اسمه وعند الوصول للكلمة next يعود للجملة مرة اخرى ولكن ينتقل للعنصر الذي يليه

اظن انك بدات فهم ما يعنيه الانعكاس سوف نتناول الان ما هي التصنيفات التي تتعامل مع كل عنصر على حدة

FieldInfo

هي فئة مشتقة من الفئة memberinfo ولا يمكن اشائها بدالة الانشاء new ولكن يتم استدعائها من الدالة GetField() من التصنيف Type

واهم الدوال هي

SetValue	التي تخصص قيمة للمتغير
GetValue	تقرا قيمة من التغير
واليك اهم الخصائص	
Is public	هذه الخصائص تختبر محدد النوصول للمتغيرهل هو عام او خاص او صديق وغيره
isPrivate	
isFriend	

مثال هاااااام جدا

في هذا المثال سوف نطبق عليه كيف نستدعي اي متغير ونعطيه قيمه او نقرا منه قيمة افتح ايم مشروع كنسل واكتب الكود التي وهو عبارة عن تصنيف سوف نجري كل اختباراتنا عليه

```
Class ReflectionExample
    Public  Field As String
    Private m_Property As Integer
    Public Property  EProperty() As Integer
        Get
            Return m_Property
        End Get
        Set(ByVal Value As Integer)
            m_Property = Value
        End Set
    End Property
    Public Sub Method(ByVal x As Integer, ByVal y As Integer)
        Console.WriteLine(-x)
        Console.WriteLine(-y)
    End Sub
    Public Sub Method2(Optional ByVal y As Integer = -1)
        Console.WriteLine(y)
    End Sub
End Class
```

في الكود القادم سوف نستدعي المتغير Field ونضيف اليه قيمة ثم تظهرها على الكنسل باستخدام الدالة GetField

```
Dim obj As New ReflectionExample()
        Dim F As FieldInfo = obj.GetType.GetField("Field")
        F.SetValue(obj, "how are you")
        Console.WriteLine(F.GetValue(obj))
        Console.ReadLine()
```

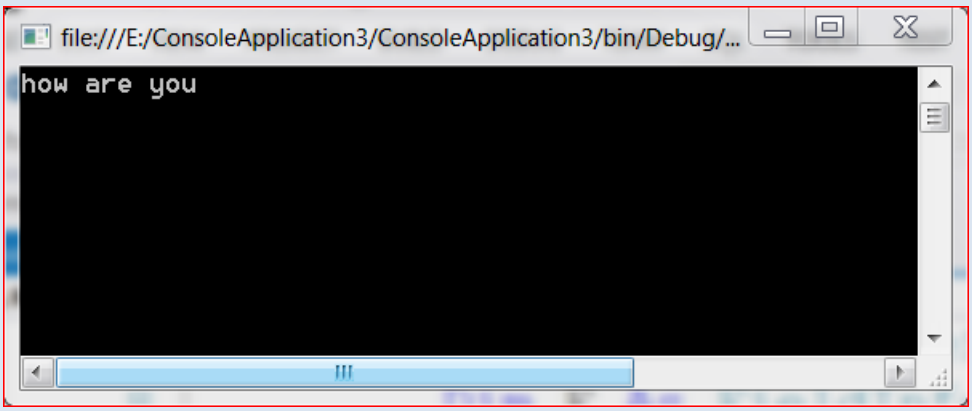
في الكود السابق انشانا كائن من التصنيف الذي كتبناه في الاول ثم في السطر الثاني اعلنا عن متغير من نوع Fieldinfo ونحن نعلم ان هذا التصنيف يستدعى من الدالة GetField من التصنيف Type تم نقوم باستدعاء المتغير

اريدك ان تعرف شيء مهم جدا هو اننا عندما نستدعي متغير معين باسمه يجب ان نراعي الاحرف الكبيرة و الاحرف الصغيرة حتى لا يعترض الكمبيوتر اثناء وقت التشغيل

اما في السطر الثالث فقد استخدمنا الدالة SetValue لاسناد قيمة للمتغير

في hالسطر الرابع استدعينا القيمة من المتغير من الدالة GetValue ثم اخرجنها للمستخدم باستخدام الكنسل

وعند التشغيل سنجد القيمة تظهر تظهر على الشاشة



ما رايك سهلة جدا و ممتعة جدا ما اريك ان نجربها في الخاصية ولكن بجب ان نتعرف على التصنيف الذي يتعامل مع الخاصية وهو propertyinfo

Propertyinfo

_هي فئة مشتقة من الفئة memberinfo ولا يمكن انشائها بدالة الانشاء new ولكن يتم استدعائها من الدالة Getproperty() من التصنيف Type

وسوف نستعرض اهم الخصائص و الطرق

GetValue	لاخذ قيمة من الخاصية وهي دالة
SetValue	لاسناد قيمة للخاصية وهي دالة
CanRead	هل الخاصية قابلة للقراءة
CanWrite	هل الخاصية قابلة للكتبة
GetGetmethod	دالة لارجاع الدالة get
GetSetmetho	دالة لارجاع الدالة Set

واترك لك عزيز القارئ ان تجرب بنفسك المثال السابق ولكن طبقه على الخصائص هذه المرة لن تجد اختلاف كثير

methodinfo

_هي فئة مشتقة من الفئة memberinfo ولا يمكن اشائها بدالة الانشاء new ولكن يتم استدعائها من الدالة Getmethod() من التصنيف Type وهذا التصنيف يتعامل مع الدوال و الاجراءات

وسوف نستعرض اهم الخصائص و الطرق

returnType	نوع الارجاع في حالة اذا كانت الطريقة هي دالة
isAbstract	هل الطريقة mustoverridh

واترك لك عزيز القارئ ان تجرب بنفسك المثال السابق ولكن طبقه على الطرق هذه المرة لن تجد اختلاف كثير مع العمل ان معظم الخصائص و الطرق في هذا التصنيف مثل الخصائص التي في التصنيف Property

Parametersinfo

في التصنيفين السابقين خاصية اسمها Getparmeters هي عبارة عن مصفوفة من نوع **Parametersinfo** بحيث انها تسترجع كل المعاملات على شكل مصفوفة

كما يوجد في التصنيف خصائص مهمة وهي

Name	اسم المعامل
Prametartype	نوع المعامل
Isoptional	هل المعامل اختياري
DefaultValue	القيمة الافتراضية للمعامل

المعامل E

المعامل e هو المعامل الذي يظهر في الاجراء الذي يفعل عندما نختار اي حدث داخل تطبيقات النوافذ

فعند اختيار الحدث load فاننا سنجد ان الاجراء يظهر كما في الشكل الاتي

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As New SalaryCls


End Sub
```

هذا هو المعامل الذي نريده

كما ترى في الشكل السابق ان المعامل e هو معامل يكتب عند اختيار الحدث الذي نريده فيقوم المترجم بكتابة الاجراء الذي نريده فيظيف اليه معاملية معامل هو Sender وهو من نوع object ومعامل اخر هو المعامل e هذا المعامل يختلف نوعه من حدث لاخر فننا اذا اختارنا الحديث mousemove يصبح نوع هذا المعامل هو **mouseEventArgs** ... هذا التصنيف مهمته التعامل مع حالات نقر الماوس بحيث يحدد من اي زر تم النقر كم نقر ما هو موقع الماوس وغيرها من الخصائص التي سنوضحها

Click	هذه الخاصية رقمية مهمتها تحدد اذ نقر المستخدم على الماوس مرة او مرتين فلو كان النقر مرة يعطي رقم 1 ولو كان مرتين تعطي رقم2
Delta	هذه الخاصية رقمية تحدد وضع الماوس فلو كان وضع الزر في الضغط تعطي رقم 0 ولو رفع المستخدم يده عن الماوس يعطيرقم 1
Loction	تحدد موقع الماوس على النموذج بحيث تحتوي على خاصيتين هما X, y
Button	تحدد على اي زر نقر المستخدم هل على الزر الايمن او الايسر

X	هي الخاصي تحدد الاحداثي السيني مؤشر الاوس
y	هي الخاصي تحدد الاحداثي الصادي مؤشر الاوس وسوف تتعرف عليها بالتفصيل في كتاب تعلم Gdi للمؤلف وهو الخزء الثاني من السلسلة

ويجب ان تعرف ان كل هذه الخاصائص للقراءة فقط

وليس هذا التصنيف فقط الذي يضاف على المعامل e فيوجد تصنيفات اخرى اختر الحدث **keyDown**

ستجد ان المعامل e اصبح نوعه keyEventArgs والذي يتعامل عندما يقوم المستخدم الضغط على لوحة المفاتيح واليك اهم الخصائص

Alt	تحدد هل قام المستخدم بالتغط على الزر Alt
Shift	تحدد هل قام المستخدم بالتغط على الزر Sheft
Control	تحدد هل قام المستخدم بالتغط على الزر Ctrl
keyCode	هنا نحدد على اي زر قام المستخدم بالنقر عليه
Handle	هنا يحدد هل قام المستخدم بتفعيل هذا الحدث
keyValue	هذه الخاصية تحدد الرقم الاساسي للحرف بحيث انها تستخدم الترميز الوحد اسمه uncode وهو يحدد رقم كل مفتاح في لوحة المفاتيح فمثلا الحرف A يعطيه الرقم 65 ويمكنك ايضا استخدام دالة اخرى غير هذه الخاصية هي الدالة Asc(string) بحيث انها تعطيك رقم كل مفتاح تدخله في المعامل النصي داخل الاداة ويمكن معرفة على ماذا يرمز اي رقم باستخدام الدالة Chr(int) بحيث انك تدخل اي رقم فيعطيك قيمته الحرفية فمثلا الرقم 70 يعطيك الحرف F
Modefiers	هذه الخاصية تحدد المفتاح الذي ضغط عليه المستخدم مع النقر على اي من الارزار Shift او Alt او Ctrl

وكما يوجد ايضا تصنيفات اخرى للمعامل e بحيث انه لو اخترنا الحدث paint فان المعامل e يصبح نوعه Graphics وهذا المعامل سوف اشرحه بالتفصيل في الكتاب GDi في هذه السلسلة

جملة الFOR EACH

تخيل لو اننا نصمم نموذج به 10 صناديق نصوص TextBox ونريد ان نمسح كل القيم المكتوبة على كل صندوق فلو كتبنا كل صندوق على حدة سوف يكون هذا مجهدا جدا يعني لو كتبنا

Textbox1.text = “”

Textbox2.text = “”

Textbox3.text = “”

سيكون ذلك مجهدا ولكن الجملة For Each توفر لنا مجهود كبير في كتابة الكود المكرر فتقوم هي بالنيابة عنك ولكي تكتب هذه الجلمة عليك باتباع القاعدة الاتية

```
For each member in Group

//الاوامر

next
```

Member يقصد به العناصر التي نريد ان نتعامل معها داخل الحلقة

Group المجموعة التي تحتوي على تلك العناصر

مثال

سنكتب الان الجملة وتكون مهمتها هي افراغ العشر صناديق النصوص في النموذج فاكتب هذا الكود في الحدث Load

```
For Each i As Control In Me.Controls
            If TypeOf i Is TextBox Then
                i.Text = ""
            End If
        Next
```

في الجملة السابقة قمنا بالاعلان عن متغير i من نوع Control من داخل النموذج Form1 ولكننا نكتب الكود من داخل هذا النموذج او من نفس التصنيف لذا فاننا نكتب كلمة me ثم نكتب الخاصية Controls وهي خاصية من نوع ControlCollection و مهمتها هي ان نقوم بتجميع كل ادوات التحكم التي يتم وضعها في النموذج فيتم تسجيل بياناتها داخل هذه الخاصية Controls .

فنقوم باستخدام الجملة For Eact باستخدام المتغير I لياخذ العناصر التي في الخاصية Contols فيجري عليها عمليات ولكننا حددنا فقط الادوات فقلنا ان الادوات يجب ان يكون من نوع Textbox كما كتبنا في السطر الثاني فجملة الشرط if تختبر هي المتغير i من نوع textbox ويتم تحديد ذكل باساخدام الجملة Type of is التي تختبر نوع المتغير فاذا كان مثل الذي يلي كلمة is فانه تعطي قيمة true فيتحقق الشرط في الجلمة if فنتفذ الامر الذي يلي كلمة then وهو ان الخاصية textالتي تخص المتغير I الذي هو من نوع control تكون قيمتها “” يعني فارق فاذا ثم نصل الى كلمة next فيعود للجملة من جديد ليتم تنفيذ الاوامر على عنصر جديد من عناصر الخاصية Controls فاذا فرغت كل العناصر نخرج من جملة التكرار

اكواد اتركك تتعرف عليها بنفسك

وكما قلت لك في المقدمة اريدك ان تتعلم كيف تعلم نفسك بنفسك بان تقرا الاكود وتتعرف على معاملاتها وذلك بقراءة الTooltip التي تظهر عليها او ان تبحث عنها في الموقع msdn فاريدك ان تتدرب على ذلك فسوف اعطيك دوال سهلة وكنا قد تعرفنا عليه ولكن لم نتناولها بالتفصيل اتركك ان تتعرف عليها بنفسك

Msgbox

Inpotbox

الدالتان السابقتان شرحتهما في الفصل الاول ولكن لم اشرحهما بالتفصيل لذا اريدك ان تتعرف عليهما بنفسك

كما اريدك ان تقرا الدوال التي شرحنها مرة اخرى ولكن هذه المرة من الفيجوال ستوديو

تمارين متروكة للقارئ

كثير ما اقرا في المنتديات العربية ان كثير من اخوتي القراء يعاني انه قد تعلم وقرأ كثيرا ولكنه لم يجد اي احد يعطيه امثلة يتدرب عليها ..ففي هذا الكتاب سوف اعطيك باقة من التمارين التي مرت علي في حياتي العملية واتركك تتدرب عليها بنفسك حتى لا يكون لديك حجة بعد اليوم

الباقة الاولى (الدوال)

التدريب الاول التقريب لاقرب شلن(5 قروش)

طلب منك مديرك ان تصمم برنامج يقوم بعمل حساب قيمة المبيعات وبما انك تعمل في شركة حكومية فان المصالح و الشركات العامة تتعامل بالقرش و المليم لذا فان المدير يريد عمل برنامج يقوب بحساب المبيعات بحيث ان يكون الناتج مقرب لاقرب شلن بمعنى لو ان لدينا هذه القيمة 15.52 فيصبح الناتج 15.55 ولو كانت القيمة تساوي 15.56 فيصبح الناتج 15.60

فالمطلوب وانت تقوم بعمل برنامجك تاخذ في الاعتبار بكتابة دالة تحسب لاقرب شلن

التدريب الثاني كتابة التاريخ

دائما ما يتم كتابة التاريخ فنقوم ونحنن نكتب تاريخ معين وليكن 2 من يانايير لسنة 2013 فاننا نكتبها هكذا 2013/1/2 فما رايك نقوم فقط بكتابة ارقام بدون العلامة / وتقوم الدالة بكتابة التاريخ فمثال نكتب 23122013 فيكون التارخي 2013/13/23 ...فكر في هذه الدالة مع الاعتبار بالاتي

- -الاخذ بالاعتبار بان المستخدم قد يكتب كلاتي 12 فيجب ان يكون التاريخ 2013/2/1
- الاحذ بالعتبار بايام الشهرو بحيث يكون الاشهر يانيار و مارس و مايو و يوليو و اغسطس و اكتوبر و ديسمبر 31 يوم و ابريل و يوينو و سبتمبر و نوفمبر 30 يوم وناخذ في الاعتبار شهر فيبارير ان يكون به 28 في السنين العادية ويكون 29 يوم في السنة الكبيسة التي يكون رقمها يقبل القسمة على 4
- ان يكون ترتيب الايام يوم / شهر /سنة
- اذا كتب المستخدم الرقم خط هكذا 02202013 فانه يكون التاريخ هكذا 2013/2/20

التدريب الثالث تعديل الاسماء

هذا الدالة تستخدم في تصحيح الاسم اثناء كتابة المستخدم للدالة وذلك بسبب اننا دئما ما يحدث خطا في البحث عن اسم معين وذلك بسبب اختلاف طريقة الادخال بمعنى ان يخطئ المستخدما مثلا في كتابة اسم احمد فمنهم من يكتبه أحمد بوجود همزة على الالف ومنهم يكتبها بدون همزة او كلمة عبد الرحمن فمنهم من يكتبه عبدالرحمن مع عدم وجود مسافة بين كلمة عبد وكلمة الرحمن وهكذا لذا من الافضل كتابة دالة تقوم توضع قاعدة في الادخال و البحث بحيث يكون حرف (ا) بدون همزة وحرف (ى) بفتكون ي بنقطتين ويكون كلمة عبدالرحمن بدون مسافة كما كلمة ابوعلي يكون بدون مسافة وكذا فقم بعمل هذه الدالة باستخدام دوال البحث

الباقة الثانية التصنيفات و الادوات

التدريب الرابع عمل تصنيف لحساب الدفع الاجل

افترض انه طلب منك عمل برنامج لحساب المبيعات التي تباع بالاجل بحيث انه لو قام العميل بشراء اي سلعة وقام بدفع الحساب نقدي فان خاصية الايرادات ستكون قيمتها بنفس القيمة المدفوعة ويكون الخاصية Cash التي تختبر اذا كان الحساب مدفوع كاش او اجل تكون True

ولكن اذا فام العملي بشراء السلع بدفع اجل عليك ان تتبع الاعتبارات الاتية

- ان تاجيل دفع السعر ينتج عنه زيادة في نسبة قيمة الاستحقاق بمعنى لو كنت تشتري 30 قطعة ب20 جينه بدون ان تدفعه نقدي وسوف ندفعه اجل اكثر من شهر فانه تزيد قيمته بنسبة مئوية تحدد على اساس الاتي
  1- تحدد النسبة المئوية على اساس الشرائح بحيث لو قام العميل بشراء من 50قطعة الى 100 قطعة تكون نسبة الزيادة في سعر الاجل 7% واذا كنت كمية الشراء من 101 - 150 تكون النسبة 5% واذا كانت الشريحة في الشراء من 150 لاعلى فان النسبة ستكون 3%
  2- لا يسمح بشراء بكمية اقل من 50 قطعة بدون دفع نقدا
  3- لا يسمح ان يؤخر الاتفاق على دفع الاجل عن مدة تزيد عن 6 اشهر
- ان حساب الايراد في حالة الدفع الاجل يتم عن طريق المعادلة الاتية
  قيمة المبيعات * المدة بالسنة * نسبة الزيادة كما هي موضحة في الشرائح السابقة
- ان حساب قيمة المبيعات تحسب كلاتي : سعر الوحدة * نسبة الخصم * الكمية
- ويحدد نسبة الخصم كلاتي اذا قام العميل بشراء من 25 قطعة الى 50 فان نسبة الخصم تكون بنسبة 5% واذا قام العميل بشراء من 51-100 فيتكون نسبة الخفض تزيد 10% وذا كان 100 الى 150 فتكون النسبة 20% وذا كانت اكثر من 150 فان نسبة الخفض تساوي 25%
- عليك تاخذ في الاعتبر ان نسبة الزياد والتي تسمى الفائدة التي تحسب على الايراد الاجل تختلف عن سعر الخفض يستخدم في حساب قيمة المبيعات حيث انها تساوي تساوي سعر الوحدة *نسبة الخفض * الكمية بمعنى لو اننا نحسب سعر الوحدة فنفترض ان العميل اشترى 60 قطعة وسعر القطعة 55 فان المعادلة تساوي
  سعر الوحدة = 55 للوحدة
  نسبة الخفض = 10% لان الكمية بين 51-100
  ثم الكمية تساوي 60
  فيكون قيمة المبيعات يساوي 55*10%*60 فيكون الناتج 300
- اما نسبة الزيادة تضرب في قيمة المبيعات التي تساوي سعر الوحدة وليكن 50 بعد حساب نسبة الخصم في الكمية ولتكن 60 فيكون قيمة المبيعات تساوي 300 وعند ضربها في نسبة الزيادة فاننا ننظر الى الشريحة التي

تحدد النسبة فنجد ان العميل اشترى 60 قطعة فان نسبة الزيادة او الفائدة تساوي 7% ونفترض ان المستخدم سيسدد الايراد بعد 3 اشهر فيتم حساب الايراد الكلي كلاتي

قيمة المبيعات = 50*60 = 300

المدة تساوي = 3اشهر يعني 3/12 من السنة لان المدة تحسب على اساس السنة فيكون قيمة الشهر في البسط و الاثنى عشر شهر في المقام من الكسر

نسبة الزيادة = 7%

فتكون المعادلة كلاتي 300*(3/12)*07.

- سعر الوحدة الواحدة يحدد من الادارة

فكر في هذه المسئلة فقم بتحديد المدخلات وهي سعر الوحدة و الكمية و تاريخ الدفع و تاريخ الشراء

اما المخرجات فتجد انه قيمة الايراد و سعر الوحدة بعد الخصم ومدة الاجل وهي المدة بين تاريخ الشراء وتاريخ الدفع وهل الدفع نقد ام اجل

ستجد ان لديك دوال ... دالة لحساب شراء نسبة الزيادة و دالة لحساب نسبة الخصم ودالة لحساب الايراد الكلي الذي يجب على العميل بدفعه في تاريخ الدفع و دالة لحساب سعر الوحدة ودالة لتحديد مدة الاجل

هذه تحيلي للبرنامج قم بكتابة الكود وعليك ان تستخدم الاحداث لتنبيه بعدم زيادة مدة الاجل عن 6 اشهر

تدريب خامس New Textbox

قم ببناء صندوق نص خاص بك بحيث انك في هذا الصندوق بكون به الخصائص الاتية

خاصية لتحديد لون الخلفية عند الدخول للكتابة عليه

خاصية لتحديد لون الخلفية عند تركه

خاصية لتحديد لون الخلفية عندما يكون الصندوق للقراءة فقط

خاصية منع كتابة اي قيمة غير قيمة رقمية

الى هنا اكون قد انتهيت من هذا الفصل واعطيتك ما يكفي من التمارين لكي تتدرب عليه وانصحك قراءة الفصل السابق عدة مرات لكي تستوعبه جيدا ويجب ان تنتبه ان كل حل كل التمارين التي اعطيتها لك موجود في هذا الكتاب فانصحك ان تبحث عنه كما انصحك ان تقرى هذه الفضاءات الاسماء الاتية الى ان اشرحا لك في الكتب الاتية في السلسلة

System.Drowing فضاء اسماء مسئول عن رسم باستخدام البرمجة وسوف اشرحه في الكتاب الثاني
System.Data هذا فضاء الاسماء يحسب كيفية التعامل مع الدالة وسوف اشرحها في الكتاب الثالث
System.oi هذا الفضاء يتعامل مع الملفات وسوف اشرحها في الكتاب الرابع
مع وان شاء الله سوف افرد كتاب لحل كل التمارين المتروكة للقارئ في الاربعة كتب مع تمارين

اخرى

والى هنا اكون قد اتممت هذا الكتاب واسئل الله العلي القدير ان تكون قد استفدت من ما قدمته لك حل لكل المشاكل التي تبحث عنها فان كان ذلك من احسان فمن الله وان كان هناك تقصير فمن نفسي ومن الشيطان

واذا اردت ان تتعرف اكثر عن تصنيفات البرمجة انصحك ان تذهب لمن صنع هذه التصنيفات ... اذهب لشركة مايكروسوفت وذلك بزيارة موقعها الاساسي لتعليم الnetframwark واليك الرابط الاتي

http://msdn.microsoft.com

ت المضيئة