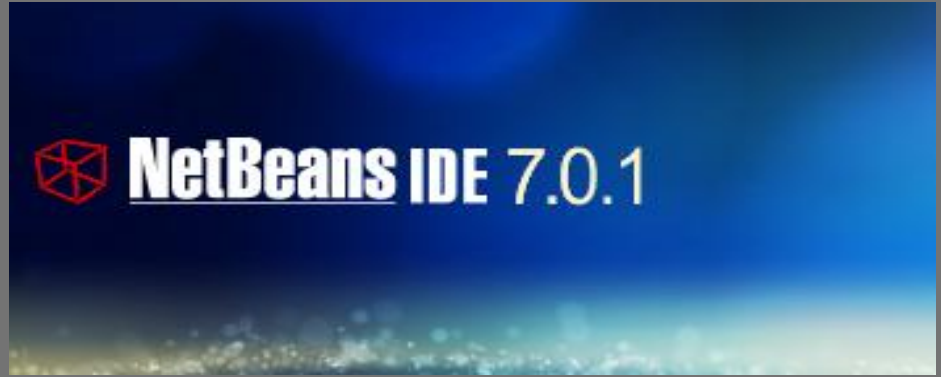


2014

تعلم الجافا الرسومية باستخدام



بواسطة : أحمد مناع

هذا الكتاب سيأخذ بيدك الى بداية أحتراف لغة الجافا بخطوات بسيطة و سهلة الفهم بعيدا عن التعقيد الاكاديمي ... يبدأ معك من بداية تجهيز حاسبك بمنصة الجافا ثم بيد في سرد الاساسيات بطريقة سهلة و مختصرة في الوقت ذاته ثم نبدا في التعرف على استخدام المكونات الرسومية باستخدام السحب و الاسقاط وكيفية التعامل مع أهم المكونات الرسومية و أكثرها شيوعا و إستخداما ... في باب آخر تبدأ في كيفية الإتصال و التعامل مع قواعد البيانات مثل أكسس , MySQL , java db وفي الجزء الاخير من الكتاب يعطيك فكرة عن كيفية برمجة تطبيقات الجوال من خلال NetBeans

كافة حقوق الاعداد محفوظة لمنتدى مصر التقني و مع ذلك لكل شخص يريد التعلم لك الحق في إعادة نسخ أو توزيع أو اقتباس أى من محتوى الكتاب بدون أى شروط تذكر



المحتويات

الباب الاول : الاساسيات 5

الفصل الاول : إعداد بيئة العمل

- 5..... التعرف على إصدارات الجافا •
- 6..... كيفية إعداد جهازك لإنشاء تطبيقات الجافا •
- 8..... تثبيت JDK البيئة الخاصة بلغة الجافا •

الفصل الثاني : الهيكل العام لبناء تطبيقات الجافا

- 11 التعرف على الفئة او الكلاس (Class) و أنواعها •
- 11 الهيكل العام لبناء أى تطبيق جافا •
- 12..... خطوات تنفيذ أى تطبيق جافا •
- 13..... إنشاء أول تطبيق باستخدام محرر نصوص بسيط NOTEPAD •
- 18 إنشاء أول تطبيق من خلال **NetBean** •

الفصل الثالث : أنواع البيانات وكيفية تمثيل الثوابت و المتغيرات

- 24 أنواع البيانات فى الجافا •
- 25 التعرف على الثوابت •
- 28 التعرف على المتغيرات •
- 30 العمليات الحسابية و المنطقية •
- 31 عامل الزيادة وعامل النقصان •

الفصل الرابع : طرق الادخال و الاخراج فى الجافا

- 32 طريقة إدخال البيانات باستخدام JOPTIONPANE •
- 35..... طريقة أخراج البيانات باستخدام JOPTIONPANE •
- 38 طريقة إخراج البيانات باستخدام Standard output •
- 40 طريقة إدخال البيانات باستخدام الكلاس Scanner •

الفصل الخامس : أدوات التحكم البنائى

- الجمل الشرطية
 - 44 جملة IF ○
 - 46 جملة **if...else** ○
 - 47 جملة if...else if...else ○
 - 48 جمل IF المتداخلة ○
 - 48 جملة switch ○

حلقات التكرار

- 51 الحلقة التكرارية while ○
- 51 الحلقة التكرارية do...while ○
- 52 الحلقة التكرارية for ○

53 الحلقة التكرارية ENHANCED FOR

الفصل السادس : المصفوفات ARRAYS

- 54 تعريف المصفوفة و انواعها
- 56 إدخال العناصر الى المصفوفة في البيئة غير الرسومية
- 57 قراءة عناصر المصفوفة
- 57 إدخال العناصر الى المصفوفة في البيئة الرسومية
- 60 ترتيب عناصر المصفوفة

الفصل السابع : جمل معالجة الاستثناءات :

- 62 أقسام جمل معالجة الاستثناءات
- 63 الصيغة الكاملة لكتلة معالجة الاستثناء
- 63 آلية تنفيذ كتلة الاستثناءات

الفصل الثامن : التعامل مع الفئات Classes و الكائنات Objects و الطرق Methods

- 66 إنشاء الفئات
- 66 الكائنات Objects
- 67 الطرق Methods

الباب الثاني : واجهة المستخدم الرسومية

الفصل الاول : الاطارات أو النوافذ

- 72 أنواع الاطارات و النوافذ في الجافا
- 72 إنشاء النوافذ التابعة للحزمة swing
- 77 التعرف على أهم خصائص jframe
- 77 أهم الاحداث الخاصة بانوافذ و الاطارات
- 79 إنشاء النوافذ الخاصة بالحزمة awt

الفصل الثاني : حقل التسمية JLabel و زر الامر JButton

- 81 العنصر الرسومي JLabel
- 81 خصائص العنصر الرسومي JLabel
- 82 العنصر الرسومي JButton زر الامر

الفصل الثالث : مكونات النصوص Text Components

- 86 العنصر الرسومي: JTextField
- 88 العنصر الرسومي JTextArea
- 88 العنصر الرسومي JPasswordField
- 89 العنصر الرسومي JFormattedTextField

الفصل الرابع : مربع الاختيار المتعدد jCheckBox و خانة الاختيار jRadioButton

- 91 مربع الاختيار المتعدد jCheckBox
- 94 خانة الاختيار jRadioButton

الفصل الخامس : اشرطة القوائم Menu Bar

- 105 التعامل مع المربع الحوارى jFileChooser

الباب الثالث قواعد البيانات

الفصل الاول : قواعد بيانات Java DB

- 109..... إنشاء قاعدة بيانات Java DB من خلال NetBeans
- 111 إنشاء جدول داخل قاعدة البيانات
- 117 إدخال بيانات الى جدول قاعدة البيانات يدوياً
- 118 الاتصال بقاعدة البيانات
- 123 قراءة محتويات جدول البيانات
- 125 مثال للاتصال بقاعدة البيانات فى الواجهة الرسومية

الفصل الثانى : قواعد بيانات MySQL

- 132..... إنشاء قاعدة بيانات MySQL من خلال NetBeans على السرفر المحلى
- 138 كيفية إنشاء قاعدة بيانات mysql من خلال NetBeans على مقلم بعيد (الانترنت)
- 144 الاتصال بقواعد بيانات mysql

الفصل الثالث : قواعد بيانات Access

الفصل الرابع : قواعد بيانات Microsoft SQL Server

الباب الرابع : برمجة تطبيقات الجوال

- 155..... الفصل الاول : إنشاء المشاريع Building MIDlets
- 178 الفصل الثانى : التعامل مع المكونات التابعة للتويب Displayable الجزء الاول!
- 186 الفصل الثالث : التعامل مع المكونات التابعة للتويب Displayable الجزء الثانى!
- 196 الفصل الرابع : التعامل مع المكونات التابعة للتويب Items
- 207 الفصل الخامس : التعامل مع نظام إدارة السجلات RMS
- 218 الفصل السادس : الاتصال بالويب web

الباب الاول : الاساسيات

الفصل الاول : إعداد بيئة العمل

أخوانى الكرام أهلاً وسهلاً بكم فى اول درس من دروس دورة الجافا الرسومية و التى يقدمها منتدى مصر التقنى لكل متهم بالجافا ...والدروس ان شاء الله تعالى ستكون سهلة و سلسلة ليس لكل مبتدئ فى الجافا فقط بل لكل من ليس لديه أى خبرة فى البرمجة و لم يتعامل مع أى لغة برمجة أخرى... ولكن عليك قبل الشروع فى تعلم لغة الجافا أن تكون متأكدًا من رغبتك فى تعلمها ... فمن ليست لديه الرغبة فلن ينجح ولن يكمل .. ومن عنده الرغبة ولو كان أقل نشاطًا من سابقه فسوف يجد سهولة فى تعلم اللغة...

وقبل المضى قدما فى شرح هذا الدرس أحب أن أنوه أنه ليس هناك ما يسمى بلغة الجافا الرسومية و الجافا غير الرسومية و انما ما قصدته من العنوان) دورة الجافا الرسومية (أنا سنقوم بإنشاء برامج ذات واجهة مستخدم رسومية عن طريق الجافا... كما يجب عدم الخلط بين الجافا والجافا سكريبت فرغم تشابه اللغتين فى السياق والتركييب إلا أنهما لغتان مختلفتان ولهما أهداف مختلفة... أما من ناحية أخرى أريد أن أهدى لكم كل واحد منكم... أننى لن أحدثكم عن مميزات تلك اللغة... ولن أقول لكم من هو مؤسسها .. ولن أشرح لكم مراحل تطورها .. ولن أقارن بينها وبين أى لغة برمجة أخرى .. فأنتم تعلمون قبلى جميع هذه التفاصيل .. وأنا متأكد أن أغلبكم قد سئم منها لكثرة ما تعاد وتكرر فى منتديات أخرى.....

التعرف على إصدارات الجافا

و قبل البدء فى هذا الدرس هو إعطائك نبذة مختصرة عن أنواع أو إصدارات الجافا و فيما يستخدم كل إصدار و ذلك على النحو التالى:

JAVA SE : الإصدار القياسي من الجافا وهو اختصار لكلمة (Java Standard Edition) هو الجزء الخاص ببرمجة برامج سطح المكتب و الذى نحن بصدد (STANDALONE APPLICATIONS)

JAVA EE : اصدار الإنترنت راييس بالإنجليزية (Java Enterprise Edition) هو يختلف عن الإصدار القياسي فى أنه خاص ببرمجة الخوادم والتطبيقات الموزعة وبرمجة الويب

JAVA ME : اصدار المايكرو بالإنجليزية (Java Micro Edition) وهو اصدار خاص بالرمجة على الهواتف المحمولة والأجهزة المحدودة فى المصادر عموما وتعتبر الخيار الأول لمبرمجي ألعاب الهواتف المحمولة.

كيفية إعداد جهازك لإنشاء تطبيقات الجافا

لكي تتمكن أخي الكريم من إنشاء تطبيقات بلغة الجافا يجب عليك أولاً تثبيت بيئة العمل الخاصة بلغة الجافا على جهازك و هي تسمى JAVA DEVELOPMENT KIT وتختصر الى **JDK** وتم تطوير تلك البيئة من قبل شركة SUN MICROSYSTEMS المملوكة لشركة أواركل حالياً

وبعد تثبيت **JDK** على جهازك تستطيع لكتابة أكواد الجافا بمحرر نصوص بسيط مثل Notepad المرفق مع الويندوز. أما أنظمة التشغيل الأخرى فهي أيضاً تحتوي على محرراتها فعلى سبيل المثال هناك VI على UNIX، وكذلك على linux، هناك أيضاً Text edit على الماكنتوش...

فيواسطة هذا البرنامج الصغير (المحرر) يمكنك كتابة برنامج جافا بأكمله بجميع أكواده، بعد ذلك يجب عليك حفظ هذا الكود في ملف بإمتداد File.java :

لكن هناك مشكلة أساسية لن يحلها هذا المحرر البسيط؛ وهي أنك إن أردت كتابة برنامج كبير، فلن يتحقق لك ذلك إلا بمشقة الأنفس ! لماذا ؟ .. ببساطة لأن المحرر الذي يأتي مثلاً مدمجاً في الويندوز لا يحتوي على ألوان تفرق لك كلمات الكود لتسهيل قراءته وكذلك لتسهيل كتابته، ولا يحتوي على نوافذ متعددة يمكنك من كتابة أكواد عديدة في نفس النافذة .

كما أنه لن يظهر لك العناصر الرسومية مثل مربعات النصوص و زر الامر ... الخ لكي تقوم بوضع هذه العناصر على النافذة بكل سهولة عن طريق السحب و الافلات مما يجعل كتابة برنامج كبير أمر شبه مستحيل!

إذاً ماهو الحل؟؟

الحل ببساطة هو اننا نحتاج بيئة التطور المتكاملة Integrated Development Environment ويتم اختصارها الى IDE والتي تسهل علينا كتابة الاكواد لإنشاء زر أمر مثلاً أو حتى إنشاء نافذة فعن طريق خطوة واحدة يتم فيها إنشاء ما تريده من العناصر الرسومية عن طريق السحب و الافلات كما تعودنا في الفجوال ببسك

ومن أمثلة تلك البيئات JBulide / Eclipse/ netbeans. ونحن هنا في هذه الدورة سوف نتكلم عن واحد من أقوى بيئات التطوير (IDE) وهو برنامج نت بينز... (netbeans) وذلك لكونه مدعوم من قبل الشركة المنتجة للغة الجافا ORACLE

لذلك أرجو أن تتبع الخطوات التالية لتحميل البيئة الخاصة بلغة الجافا JDK بالإضافة الى بيئة التطور المتكاملة وهو برنامج: NETBEANS

1_ الدخول على الموقع التالي

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2_ سنقوم بأختيار الخيار الثالث وهو تحميل JDK + NETBEANS كما في الصورة التالية:

Java SE Downloads

[Latest Release](#)

[Next Release \(Early Access\)](#)

[Embedded Use](#)

[Previous Releases](#)



Java Platform (JDK) 7u4



JavaFX 2.1



JDK 7u4 + NetBeans



JDK 7u3 + Java EE

2_ ستظر لك الصفحة التالية... قم بالموافقة على شروط الترخيص ثم بعد ذلك حدد نظام التشغيل على جهازك و سيتم بدء عملية التحميل....

JDK 7u4 with NetBeans 7.1.2

This distribution of the JDK includes the [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 7u4 and NetBeans 7.1.2 Cobundle License Agreement](#) to download this software.

Accept License Agreement Decline License Agreement

لا بد من تنشيط هذا الخيار للموافقة على شروط الاستخدام

Java SE and NetBeans Cobundle (JDK 7u4 and NB 7.1.2)		
Product / File Description	File Size	Download
Linux (32-bit)	165.88 MB	jdk-7u4-nb-7_1_2-linux-ml.sh
Mac OS X v10.7 (64-bit)	160.43 MB	jdk-7u4-nb-7_1_2-macosx-ml.dmg
Solaris SPARC (32-bit)	167.65 MB	jdk-7u4-nb-7_1_2-solaris-sparc-ml.sh
Solaris x86 (32-bit)	158.86 MB	jdk-7u4-nb-7_1_2-solaris-x86-ml.sh
Windows (32-bit)	176 MB	jdk-7u4-nb-7_1_2-windows-ml.exe

3_ أما اذا لم يتم بدء عملية التحميل وظهرت الرسالة التالية...

Thank you for accessing Oracle Technology Network.

In compliance with U.S. and applicable Export laws we are unable to process your request. Please contact RPLS-Ops_ww@oracle.com if you believe you are receiving this notice in error.

- Back to Previous Page
- Site Map
- Product Index
- Software Download Index
- To search for your page, try our Search function.

search site Technology Network Refine Search

فمعنى ذلك أن الدولة التي تقطن فيها عليها عقوبات مفروضة من قبل الولايات المتحدة الأمريكية مثل السودان و سوريا فأعلم أنك لن تستطيع التحميل من هذا الموقع ... لذلك سيقوم منتدى مصر التقنى برفع JDK + NetBeans على سرفر خاص حتى يتثنى للجميع تحميلية.

4-بعد التحميل سيظهر لك ملفان ...الاول خاص ببيئة تشغيل الجافا على جهازك... JDK أما الثاني هو الخاص ببرنامج... NetBeans كما فى الصورة التالية:



تثبيت JDK البيئة الخاصة بلغة الجافا

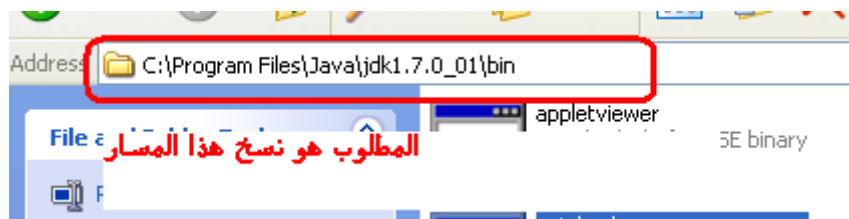
يجب عليك أحي الكريم تثبيت JDK أولا قبل تثبيت برنامج NetBeans إن لم يحدث غير ذلك سوف يظهر لك رسالة خطأ عند تثبيت NetBeans مفداها أنه لا يوجد على جهازك بيئة الجافا. JDK

وكيفيك لتثبيت JDK الضغط على زر Next حتى يتم الانتهاء من التثبيت ... لكن بقى شىء أخير يجب فعله ...أتبع التالي:

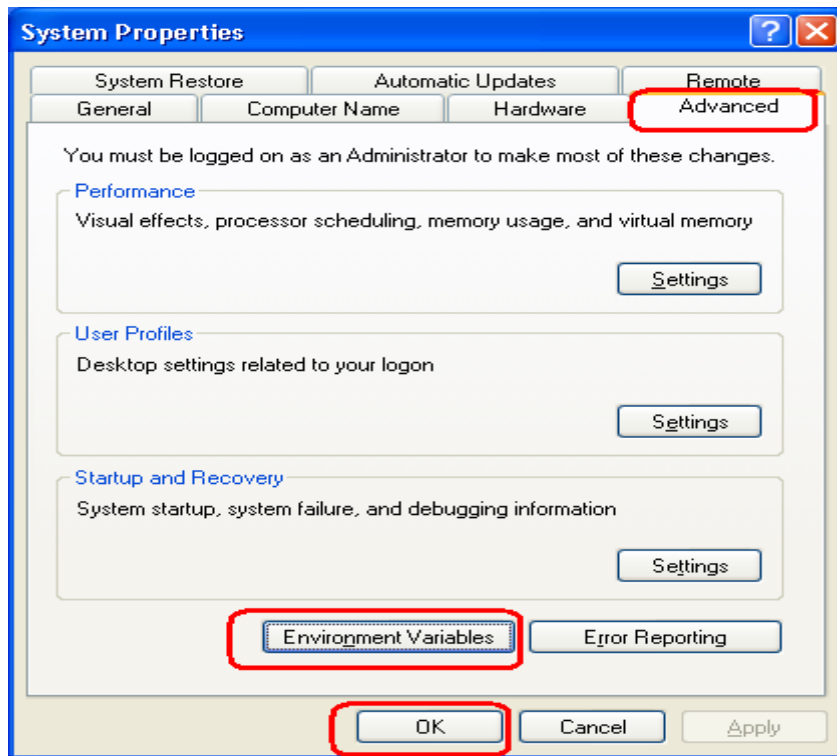
1_الدخول على المجلد الذى تم تثبيت فيه..... JDK فمثلا أنا قمت بالتثبيت على القرص C و بالتالى فأن مجلد التثبيت سيكون فى مجلد C:\Program File

2_عند الدخول فى هذا المجلد.... C:\Program Files ستجد مجلد بأسم... JAVA قم فتح هذا المجلد ...ستجد فيه مجلدان آخران ... واحد يسمى JDK1.7.0_01 و الاخر يسمى JRE7

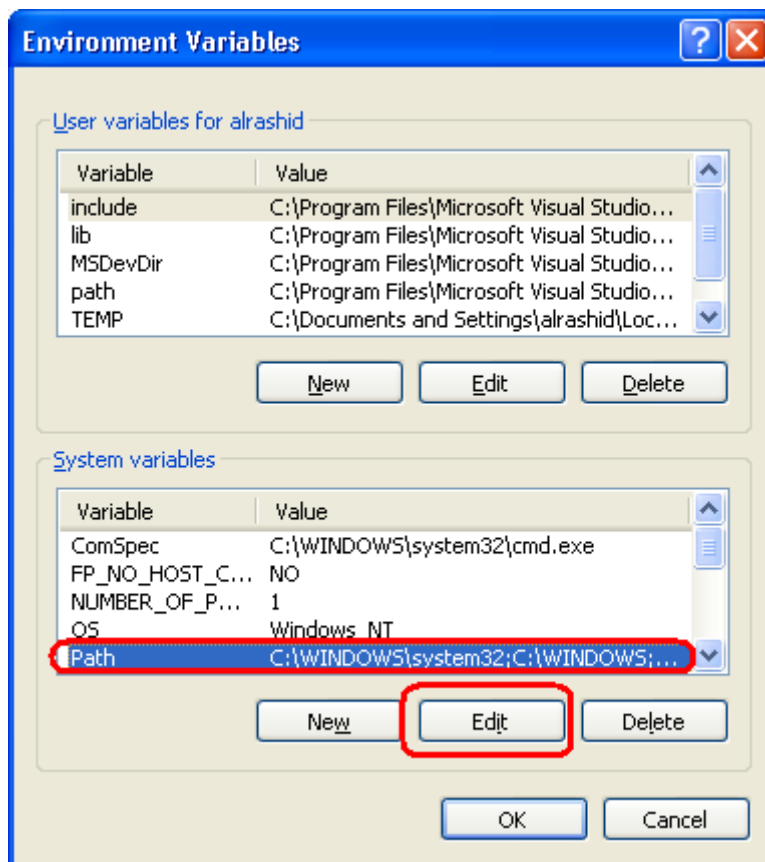
3_قم بفتح المجلد الذى يسمى.. jdk1.7.0_01 ستجد به مجلد يسمى BIN. قم فتحه هو ايضا ... وعندما تكون بداخله أنسخ المسار الموجود فى شريط العنوان كما فى الصورة التالية:



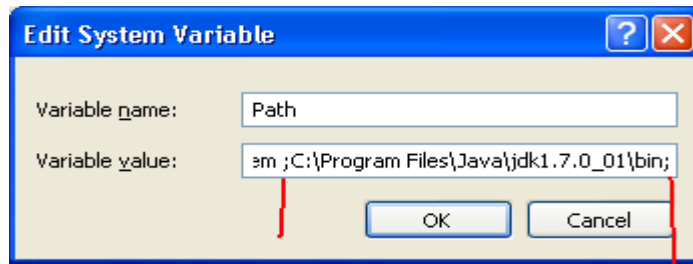
4_ثم بعد ذلك أذهب الى My Computer ثم كليك شمال ستظهر لك نافذة الخصائص ... أضغط على التويب... Advanced ...ستجد فى أخر النافذة زر الامر.... Envirment Variables كما فى الصورة التالية:



5_ ستظهر لك النافذة التالية.... كل ما عليك هو تحديد path ثم بعد ذلك أضغط Edit كما في الشكل التالي :



6_ ستظهر لك نافذة صغيرة... في مربع النص VARIABLES VALUE ستجد عدة سطر كلمات... كل ما عليك أن تضع مؤشر الكتابة نهاية هذا السطر... وعندما تتأكد أن المؤشر فعلا نهاية السطر قم بوضع فاصلة منقوطة ; ثم ضع المسار الذي قمنا بحفظه سابقا. (C:\PROGRAM FILES\JAVA\JDK1.7.0_01\BIN) ثم ننهي السطر أيضا بالفاصلة المنقوطة... ; كما في الصورة التالية :



لاحظ أن الجملة بدأت و أنتهت بالفاصلة المنقوطة

7_ ثم بعد ذلك أضغط OK_OK_OK_OK الى أن تنهي جميع النوافذ عندك.... وهكذا تكون قد جهزت بيئة عمل الجافا بنجاح على جهازك.....

و أخير قم بتنصيب برنامج... NETBEANS وكيفيك لذلك الضغط على زر NEXT حتى الانتهاء من التنصيب

....

الفصل الثانى : الهيكل العام لبناء تطبيقات الجافا

أهلا وسهلا أخواني أعضاء و زوار منتدى مصر التقنى فى ثانى درس من دروس دورة الجافا الرسومية بأستخدام NetBeans... اليوم إن شاء الله تعالى سنتعرف سويا الهيكل العام لبناء أى تطبيق جافا... وقبل البدء فى الشرح يجب أن نقف على بعض المصطلحات فى الجافا .

كل برنامج فى لغة الجافا مكون من فئة CLASS أو أكثر....أذن ما هى الفئة CLASS ؟

الفئة او الكلاس (Class) فى البرمجة كائنية التوجه (Object-Oriented Programming)، تمثل المخطط الأولي لإنشاء الكائنات (Objects)، حيث يحتوي هذا المخطط على مجموعة الخصائص (Attributes) والوظائف (Methods)، التي سوف تتشارك فيها جميع الكائنات التي سيتم إنشاؤها من هذه الفئة :

وهناك نوعين من الفئات الكلاس :

1_ فئة أساسية أو رئيسية: main يجب توفرها في أي برنامج.

2_ فئة ثانوى يتم استعمالها من طرف الكلاس الاساسى main، وهي تحتوي بشكل عام على الوظائف التي يجب على البرنامج القيام بها .

ولكى يتم فهم ما تم ذكره تعالوا نتعرف على الشكل العام لبناء أى برنامج فى الجافا .

الهيكل العام لبناء أى تطبيق جافا

بعدما مررنا بتعريف بسيط للفئة Class و معرفة أنواعها وقلنا أن أى تطبيق فى جافا لابد أن يحتوى على فئة Class على الاقل ، ندخل مباشرة لمعرفة الهيكل العام لبناء تطبيق الجافا...



شرح الهيكل العام لبناء أى برنامج جافا .

- **في السطر الاول** يتم إنشاء كلاس Class... وذلك عن طريق الامر Public ثم يتبعه الكلمة Class وهي من الكلمات المحجوزة في لغة الجافا والتي تقوم بتعريف أو إنشاء الفئة..... ثم يتبعها أسم الفئة ولقد تم تسميته هنا بالاسم الذي نريده Welcome
- **السطر الثاني** : تم فتح { قوس بداية الفئة Welcome أو كلاس Welcome .
- **السطر الثالث** : `public static void main(String args)` ... فهو يمثل جزءا من كل تطبيق جافا (Java Application) حيث يبدأ تنفيذ البرنامج من هنا أى أن هذا السطر هو نقطة بداية البرنامج... معنى ذلك أن أى تطبيق بلغة الجافا لابد و أن يحتوى على نقطة البداية ولا لن يتم تنفيذ شيئا..... ويتم إنشاء كما فى الشكل السابق
- **السطر الرابع** : تم فتح { قوس بداية الدالة main
- **أما السطر الخامس** فهو فارغ لوضع كود برمجي أى ما تريد أن يقوم به البرنامج مثل طباعة سطر معين ، حساب عدد معين.... الخ

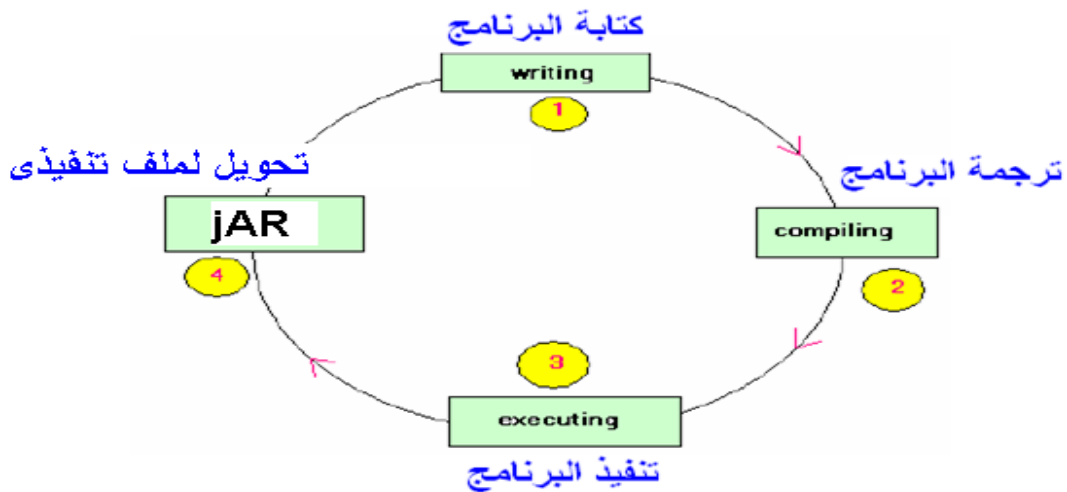
بعد ذلك يتم إنهاء البرنامج بقوسي النهاية حيث يمثل القوس الأول نهاية الوظيفة method للدالة main والقوس الآخر يمثل نهاية الفصيلة class

ملاحظات هامة :

- لابد وأن تنتهي كل جملة (سطر أو أمر) بعلامة الفاصلة المنقوطة ;
- عند تعريف الفئة class يجب أن يبدأ بكلمة public .
- يجب أن يبدأ أسم الفئة class بحرف لا برقم ، ولا يحتوى على أى مسافات فارغة ولا يكون أحد الاسماء المحجوزة في اللغة
- يفضل أن يبدأ أسم CLASS بحرف كبير مثل Welcome ، كذلك إذا كان يتكون من أكثر من أسم فإن كل أسم يبدأ بحرف كبير مثل SampleClassName .

خطوات تنفيذ أى تطبيق جافا

بعد ما تعرفنا على الهيكل العام لإنشاء تطبيق جافا... نأتى الان الى مراحل أو خطوات تنفيذ التطبيق :



- **الخطوة الاولى :** كتابة أكواد البرنامجابن؟...بعيدا عن بيئة NetBeans سنتحدث أولا عن كيفية كتابة أكواد الجافا فى محرر نصوص عادى مثل Notepad و إنشاء تطبيق جافا بواسطته ...ثم بعد ذلك ننشئ نفس التطبيق بواسطة NETBEANS لنرى كيف ساعدنا فى كتابة الكود و انشاء التطبيق بكل سهولة و يسر ...
- **الخطوة الثانية:** ترجمة البرنامج ... لماذا ؟ ...بمعنى تحويل و ترجمة شيفرات مكتوبة بلغة معينة أو source code إلى لغة يفهمها الحاسوب و نظام التشغيل...وذلك لكون كل ما كتبته من أكواد هو عبارة عن نص لا يفهمه الحاسوب، وانما تفهمه أنت ...وفي حالة لغة جافا يقوم المترجم بترجمة النص المكتوب إلى ما يسمى بالبايتكود bytecode حتى يفهمه الحاسب.
- **الخطوة الثالثة:** تشغيل و تنفيذ البرنامج لنرى مخرجات برنامجنا
- **الخطوة الرابعة :** تحويل برنامجنا الى ملف تنفيذى لنستطيع نقل وتشغيل برنامجنا على الاجهزة الاخرى فى حالة الجافا سيكون برنامجنا فى صورته النهائية بامتداد JAR

تطبيق عملي بأستخدام محرر نصوص بسيط NOTEPAD

اكتب برنامج يقوم بأظهار رسالة ترحيب ...؟؟؟؟

الحل

الخطوة الاولى: كتابة التطبيق نقوم بفتح محرر النصوص NOTEPAD ...وكتابة الكود التالى فيه :

```
import javax.swing.JOptionPane;

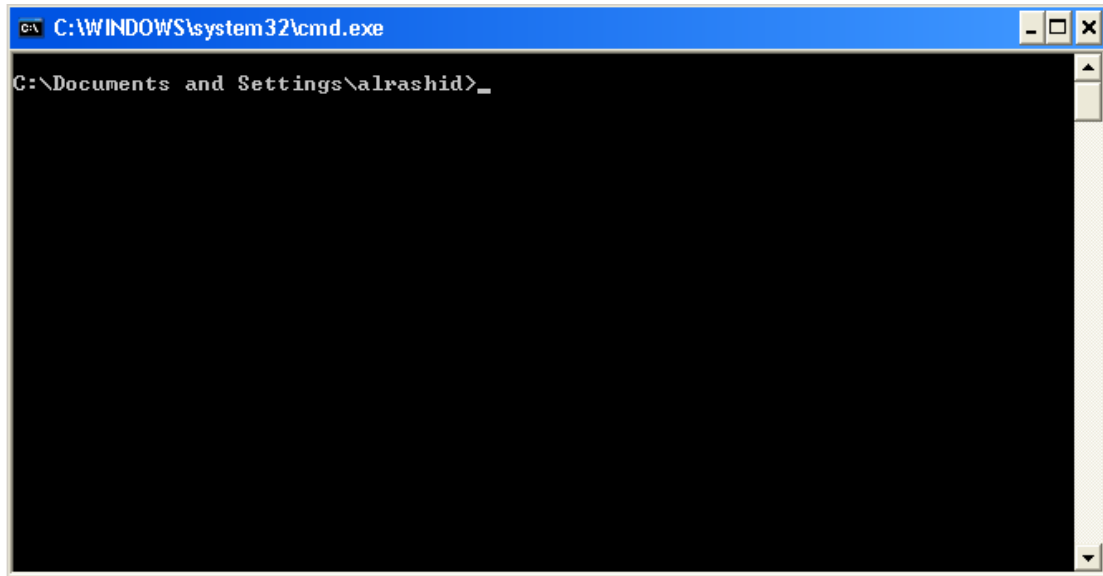
public class Welcome
{
public static void main( String args[ ] )
{
JOptionPane.showMessageDialog(null, "" );
}
}
```

شرح الكود السابق :

- ❖ السطر الاول ...قمنا بأستدعاء وإستيراد المكتبة swing_وهى الخاصة بالعناصر الرسومية وسوف يتم شرحها بالتفصيل فى دورس لاحقة_وحددنا العنصر الرسومى JOptionPane وهو المستخدم بعرض صناديق الحوار_ولها درس آخر منفصل_ وكل ما نريده الان هو كيفية أستيراد العنصر الرسومى ...
- ❖ السطر الثانى ...أنشأنا CLASS يسمى Welcome
- ❖ السطر الثالث ...قوس { للتعبير عن بداية الكلاس Welcome
- ❖ السطر الرابع ...أنشأنا الدالة الرئيسية main للبرنامج أو نقطة البداية و التى سيبدأ من خلالها الحاسب بتنفيذ الأوامر
- ❖ السطر الخامس ...قوس { للتعبير عن بداية الدالة MAIN
- ❖ السطر السادس ...كتبنا المطلوب تنفيذة من برنامجنا هو أظهار رسالة للمستخدم عن طريق الامر JOptionPane.showMessageDialog .تحتوى على النص _مرحبا بكم فى منتدى مصر التقنى _ويجب ملاحظة أننا فتحنا قوس () ثم كتبنا الكلمة المحجوزة فى لغة الجافا NULL بعد ذلك وضعنا فاصلة ، ثم علامتى اقتباس بينهما النص المراد عرضه ..و أنهينا السطر بالفاصلة المنقوطة ؛

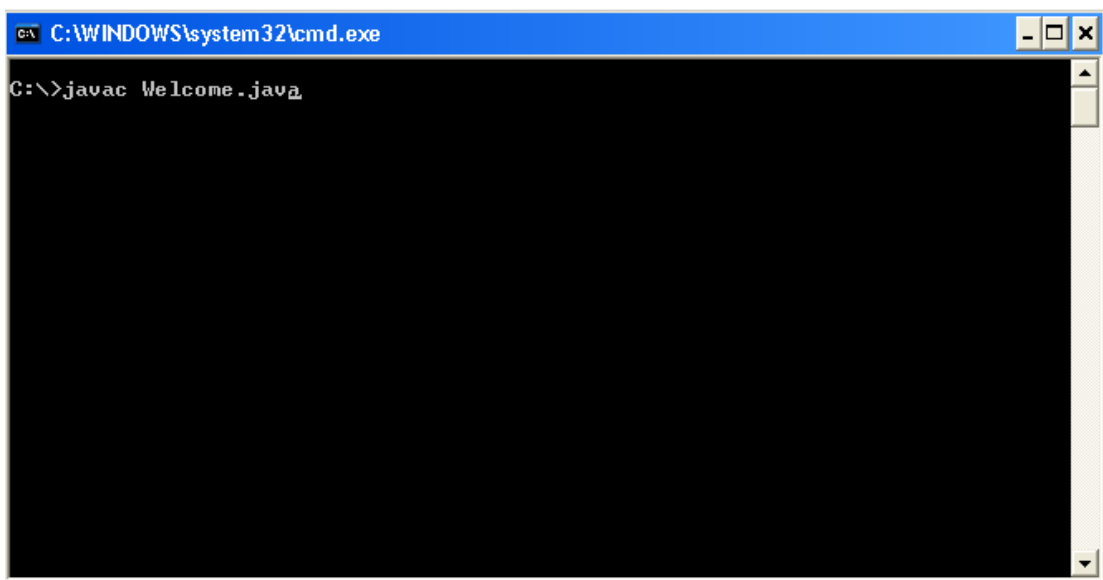
الان نقوم بحفظ الملف لكن بامتداد JAVA . وذلك عن طريق SAVE AS ولا بد أن يكون أسم الملف بنفس أسم CLASS وسوف يعطى المترجم خطأ عند الترجمة اذا لم يكن أسم الملف هو نفس أسم CLASS وكذلك إن لم يكن بامتداد JAVA .

الخطوة الثانية: ترجمة البرنامج... بعد ما قمنا بكتابة التطبيق و حفظناه بامتداد JAVA. بنفس أسم الكلاس الرئيسى WELCOME.. لنأتى الان لمرحة الترجمة... ونستطيع عمل ذلك من خلال نافذة الاوامر command lin... نذهب START ثم RUN ثم نكتب الامر CMD ثم أوكية... فتظهر لنا نافذة الاوامر... كما بالشكل التالى .



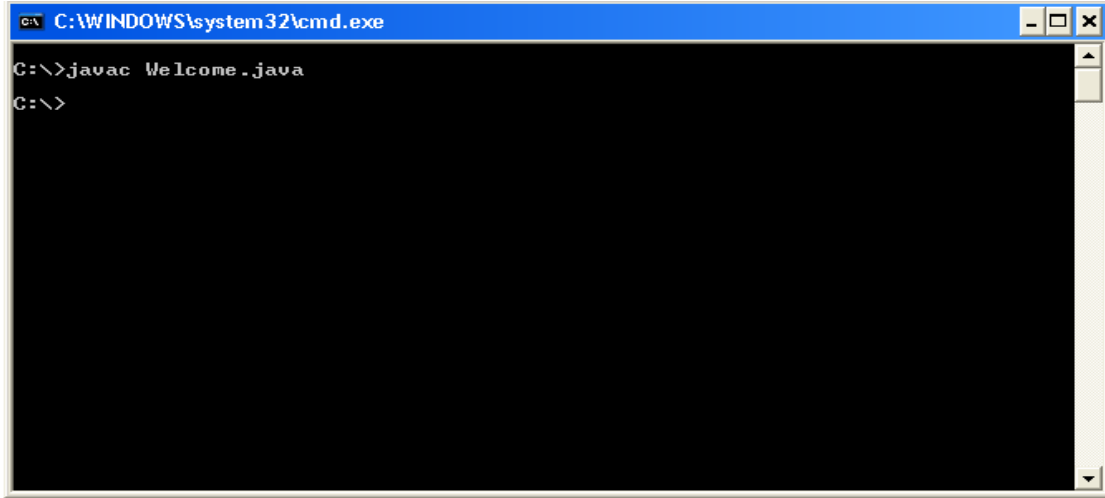
```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\alrashid>
```

أحدد المسار الذى حفظ فيه التطبيق Welcome.java... فى مثالنا هذا قمت بحفظه على C... بعد ذلك سنقوم بكتابة الامر javac Welcome.java كما فى الشكل التالى :



```
C:\WINDOWS\system32\cmd.exe
C:\>javac Welcome.java
```

ويجب ملاحظة أننا كتبنا في نافذة الاوامر **جافا سي** ثم مسافة ثم أسم التطبيق بالامتداد `Welcome.java`... الان قم بالضغط على زر `Enter` لتنفيذ الامر... أن ظهرت الشاشة التالية بدون أى `ERRO` أو سطور فمعنى ذلك أن البرنامج قد تم ترجمته بصورة صحيحة. كما فى الصورة التالية :

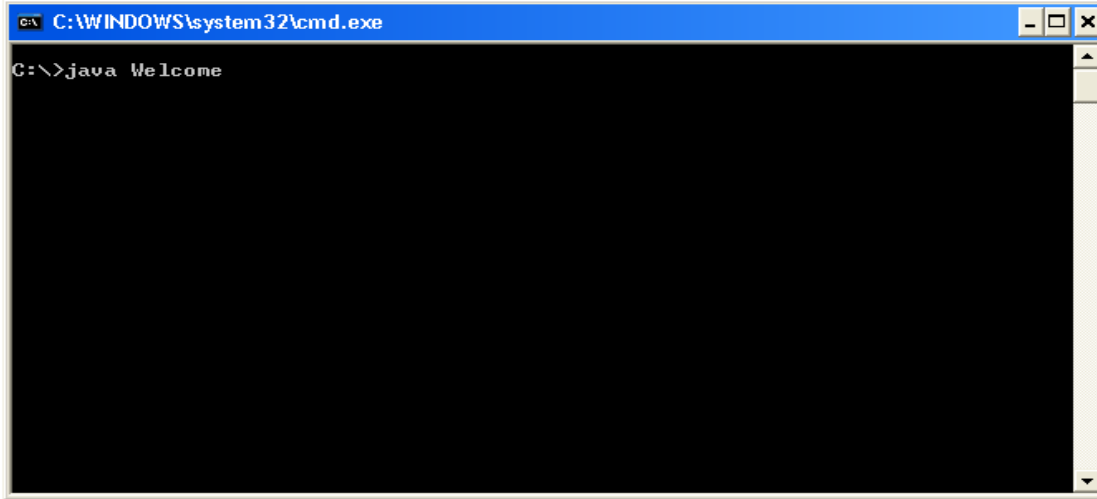


```
C:\WINDOWS\system32\cmd.exe
C:\>javac Welcome.java
C:\>
```

ولكى تتأكد أن التطبيق قد تم ترجمته...أذهب الى المجلد `C` ستجد ملف بجوار الملف `Welcome.java` يسمى `Welcome.class` كما فى الصورة التالية :



الخطوة الثالثة : تنفيذ و تشغيل البرنامج.... كل ما عليك فعله الان لى تتمكن من تشغيل التطبيق هو كتابة الامر `JAVA WELCOME`.... كما فى الصورة التالية :



الآن قم بالضغط على زر ENTER... ستجد برنامجك يعمل جيدا و تظهرك رسالة كما فى الصورة التالية :



الخطوة الرابعة و الاخيرة... تحويل التطبيق الى ملف JAR

بعد الانتهاء من كتابة أكواد التطبيق و ترجمته و تشغيله... يبقى لك خطوة أخيرة لكى يعمل البرنامج بشكل صحيح على الاجهزة الاخرى هو تحويله الى ملف تنفيذى من نوع JAR.... ويتم ذلك عن طريق نافذة الاوامر وذلك بكتابة الامر التالى :

```
JAR CF JAR_FILE_NAME DIRECTORY_NAME
```

ويجب ملاحظة أن ما هو مكتوب باللون الاحمر و الازرق مجرد متغيرات لشرح الامر فقط ونستطيع تغييرها... JAR_FILE_NAME... هو لوضع أسم للتطبيق بأمتداد JAR ونستطيع وضع أى أسم بغض النظر عن أسم CLASS... أما ال DIRECTORY_NAME هو وضع مسار مجلد التطبيق... فمثلا فى برنامجنا هذا سأقوم بكتابة الامر التالى :

```
JAR CF MYAPP.JAR C:/125
```

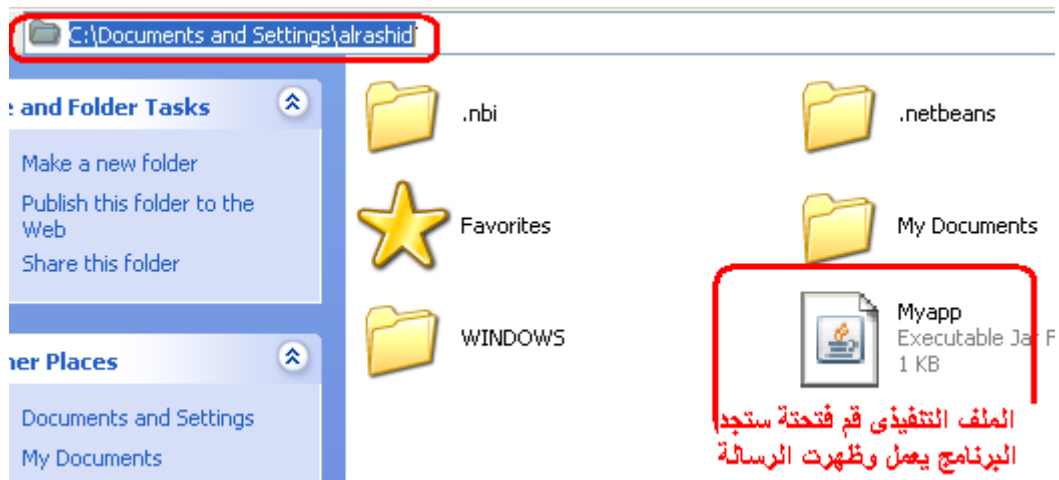


```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\alrashid> jar cf Myapp.jar c:/125
```

يجب ملاحظة أن ناتج التحويل ستجدة في هذا المسار

المسار الموجود في ملفات المشروع من نوع كلاس

هنا قمت بكتابة الامر JAR CF ثم وضعت أسم لبرنامجي هو Myapp.jar بالامتداد ثم وضعت المسار الذي يتواجد فيه ملفات المشروع وهو في مجلد ١٢٥ على C....النتيجة أننا سأجد ملف myapp في المسار الذي كنت متواجد فيه في نافذة الاوامر عندما قمت بأشائة كما في الصورة التالية :



و الى هنا ننتهى من الجزء الاول من هذا الدرس أما في الجزء الثاني سنقوم بإنشاء نفس المشروع لكن في بيئة NETBEANS لنرى كم الجهد الذي سيوفره لنا

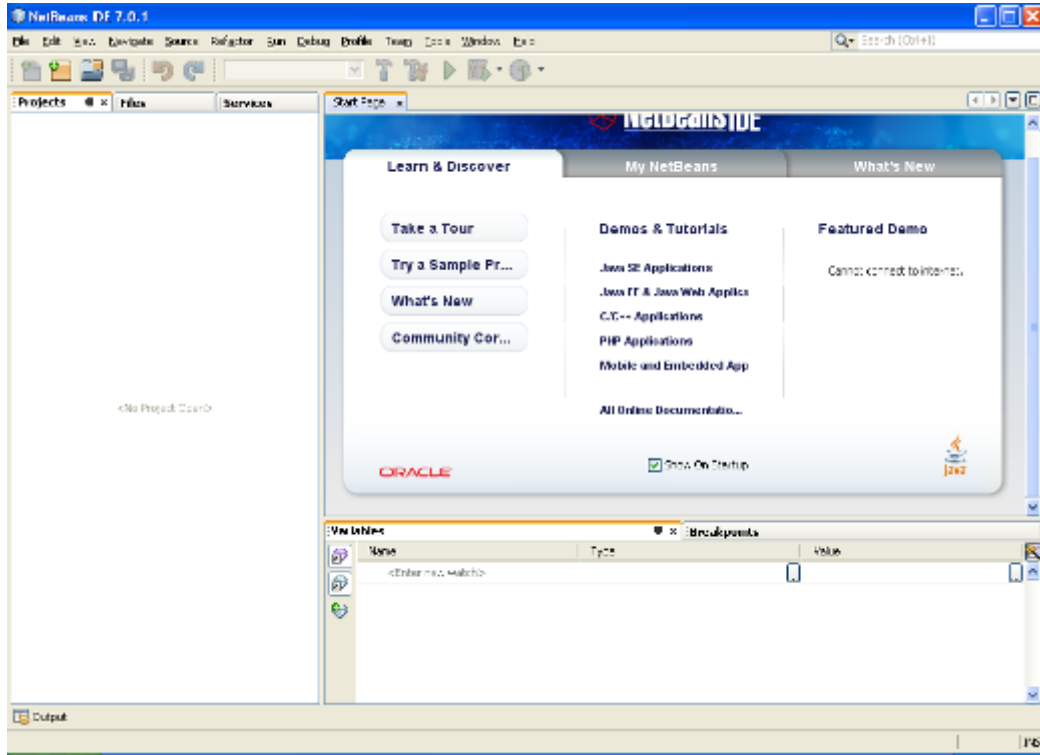
إنشاء أول تطبيق من خلال NetBean

تعرفنا في الجزء الاول على كيفية بناء تطبيق جافا... وقلنا أن أى تطبيق جافا يتكون من كلاس أو أكثر... وطبعاً عرفنا ماذا يعنى ماذا يقصد بالكلاس ويجب أن يحتوى على نقطة البداية وهى الدالة main .

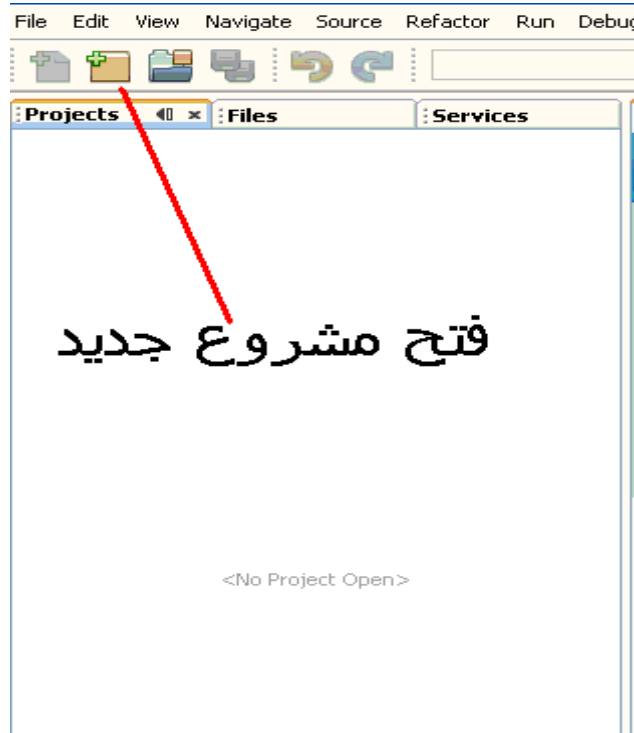
و تعرفنا سوياً على خطوات إنشاء تطبيق جافا من كتابة الكود فى محرر بسيط Notepad ثم حفظ الملف بامتداد JAVA. وتعلمنا أنه لابد من ترجمة الملف باستخدام الأمر JAVAC عن طريق نافذة الاوامر... ثم بعد ذلك قمنا بتشغيله عن طريق الامر JAVA... واخيراً قمنا بتحويله الى ملف تنفيذى JAR لكي يعمل بمجرد الضغط عليه و نقله على الاجهزة الاخرى ..

أما فى هذا الجزء سنتعلم سوياً كيفية إنشاء نفس التطبيق ولكن باستخدام NetBeans لذلك ارجوا أن تتبع الخطوات التالية :

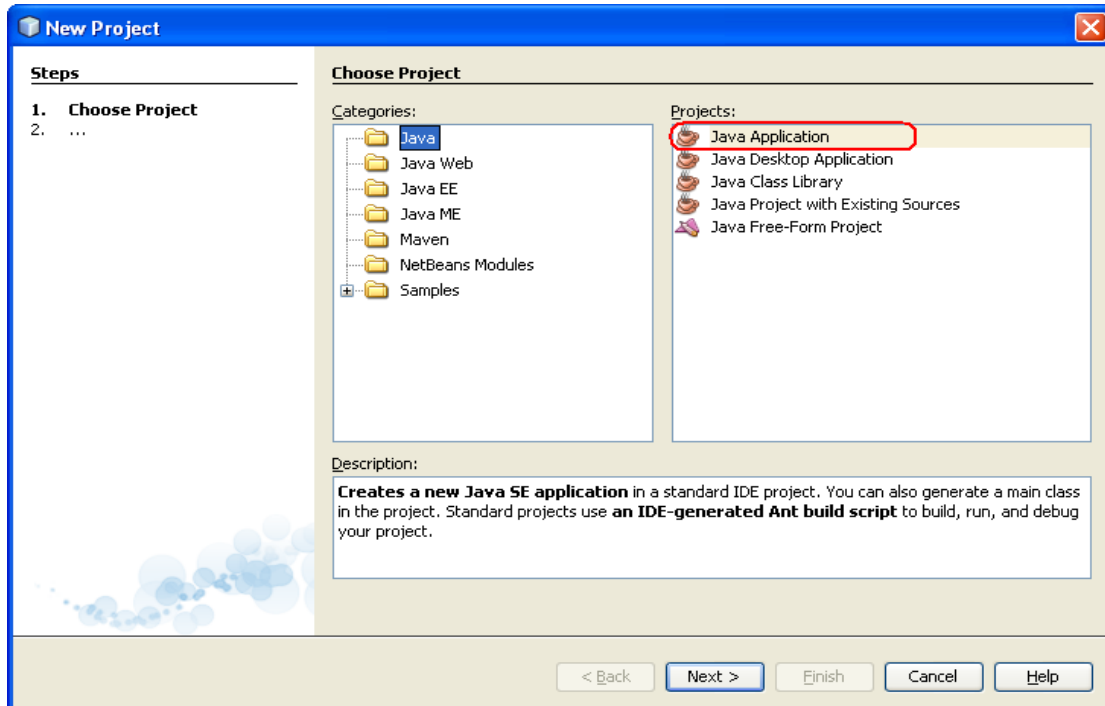
الخطوة الاولى : قم بتشغيل NetBeans.... ستجد واجهة البرنامج كما فى الشكل التالى :



الخطوة الثانية: الضغط على أيقونة إنشاء مشروع جديد.... كما في الصورة التالية :

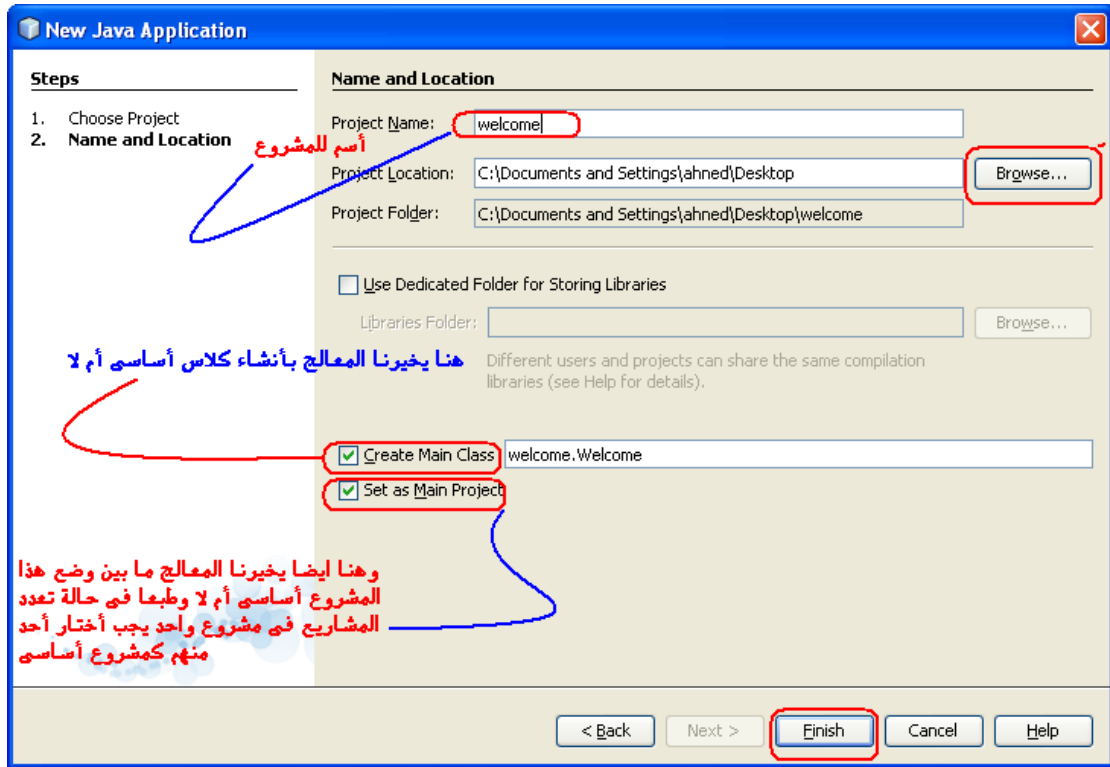


سوف يظهر لنا مربع حوارى لاختيار نوع المشروع.... كما في الصورة التالية :



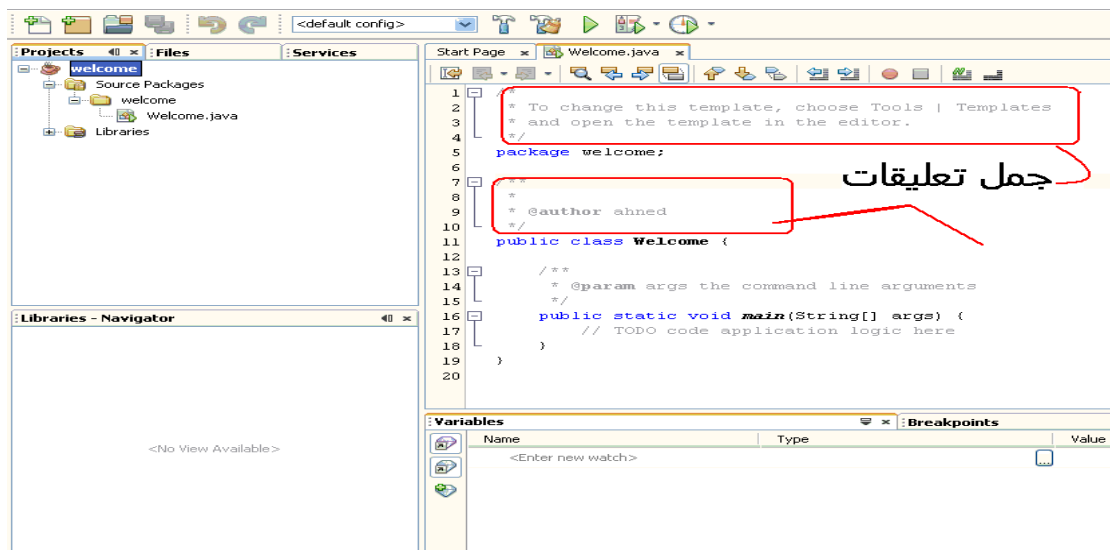
ستلاحظ أحي الكريم من الصورة السابقة في التوبيو Categories أنواع من الجافا فمنها java EE وقلنا أنه خاص ببرمجة الخوادم والتطبيقات الموزعة و Java ME و هو الخاص ببرمجة تطبيقات الموبايل... وسوف نتحدث عنه في

دورس متقدمةالذى يهنا الان هو التصنيف JAVA و عند الضغط عليه ستجد به عدة مشاريع .. JAVA application و java Desktop application الخقم بأختيار مشروع من نوع java application كما فى الصورة السابقة ثم أضغط NEXTستجد المربع الحوارى التالى :



الخطوة الثالثة: وضع أسم للمشروع كما فى الصورة السابقةوستجد ايضا أن المعالج يحدد لك المسار الذى سيتم وضع ملفات المشروع فيه ولكن تستطيع تغيير هذا المسار كما تريد من خلال زر Browse ... أكتب أسم للمشروع فى الخانة المخصص له ويجب أن يكون بالانجليزية

فى نهاية المعالج ستجد مربعين أختيار الاول: create main class: وهو الخاص بإنشاء الكلاس الرئيسى لبرنامجنا سنقوم بتحديدة أما الخيار الثانى: set as main project: وطبعا هذا فى حالة وجود عدة مشاريع فى حال مثالنا هذا سواء قمنا بتحديدة أم لم نقم بذلك فلا فائدة فيه وذلك لكون التطبيق الذى نحن بصددة مكون من مشروع واحد فقط المهم الان قم بالضغط على إنهاء لتظهر لنا شاشة كتابة الكود كما فى الصورة التالية :



كما تلاحظون من الصورة السابقة بأن البرنامج NetBeans قد أنشئ لنا تلقائياً الكلاس welcome كما أنه أنشئ الدالة MAIN أو نقطة بداية التطبيق وكل ما علينا الآن فعله هو كتابة المطلوب تنفيذ من التطبيق وهو إظهار رسالة الترحيب .

أما ما هو مكتوب بالخط الرمادي مجرد جملة تعليق مثل الجملة الخضراء التي تجدها في محرر الفجوال بيسك ٦ لا يلتفت إليها البرنامج أثناء التنفيذ وتستطيع حذفها إن شئت أو تستطيع كتابة جملة أخرى وذلك بوضع علامة // في حالة كان التعليق سطر واحد... ووضع علامة /**/ ثم جملة تعليق في أكثر من سطر ثم إنهاء جملة التعليق بوضع العامة /* كما في الصورة السابقة .

الخطوة الرابعة : كتابة التطبيق الآن سنقوم بكتابة الكود الخاص بإظهار الرسالة وذلك عن طريق استدعاء المكتبة SWING والعنصر الرسومي JOptionPane أولاً.... ثم بعد ذلك إعطاء الأمر لإظهار الرسالة كما في الصورة التالية :

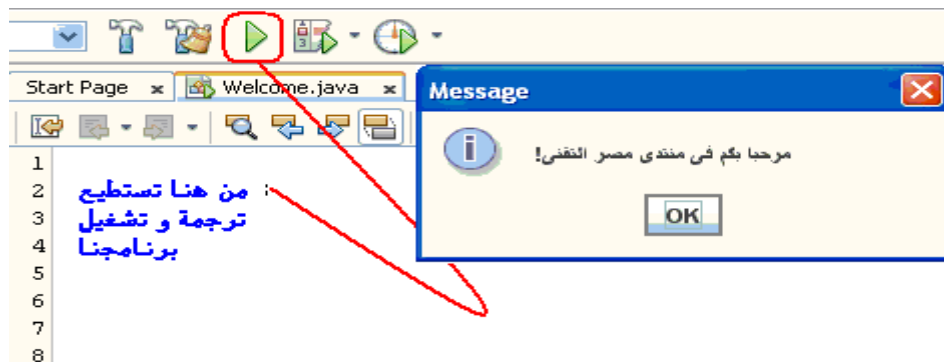
```
Start Page x Welcome.java x
package welcome;
import javax.swing.JOptionPane;
public class Welcome {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "مرحباً بكم في منتدى مصر التقني");
    }
}
```

```
Start Page x Welcome.java x
package welcome;
import javax.swing.JOptionPane;
public class Welcome {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "مرحباً بكم في منتدى مصر التقني");
    }
}
```

أستعداد المكتبة بالإضافة إلى العنصر الرسومي المطلوب

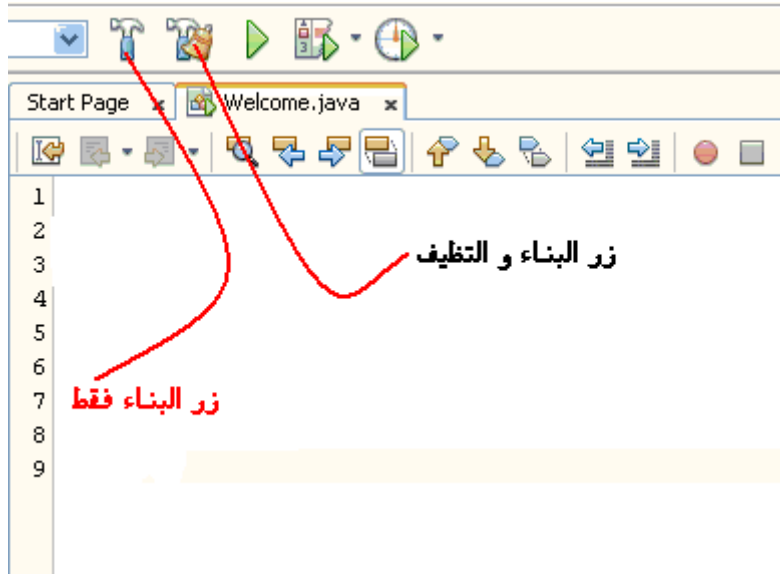
أمر إظهار الرسالة

الخطوة الخامسة : تشغيل التطبيق بعدم ما قمنا بكتابة التطبيق في بيئة NetBeans سنقوم الآن بتشغيل التطبيق ليس من خلال نافذة الأوامر كما فعلنا في الجزء الأول و إنما من خلال NetBeans قم الآن بالضغط على أيقونة RUN الموجودة في شريط مهام NetBeans كما في الصورة التالية :



الخطوة السادسة و الاخيرة : تحويل البرنامج لملف تنفيذي من نوع JAR... لكي تقوم بهذه الخطوة في بيئة NetBeans عليك أحي الكريم تشغيل التطبيق أولاً... لماذا؟! لان عندما نقوم بتشغيل البرنامج يقوم NETBEANS بترجمة التطبيق ثم بعد ذلك يتم تشغيله.... و بالتالي فلو قمنا بعملية التحويل تلك قبل عملية التشغيل سيكون التطبيق لم يتم ترجمته و لن نستطيع تحويله الى ملف تنفيذي JAR

لنأتى الان لكيفية تحويل التطبيق الى ملف تنفيذي JAR من خلال NetBeans.... لكي نقوم بذلك نذهب الى شريط مهام NetBeans و نضغط على زر clean and build أو زر build فهذا أو ذاك يقوم بعملية بناء المشروع الى ملف تنفيذي JAR لكن الزر الاول يقوم بعملية تنظيف لبعض الملفات الغير ضرورية للتطبيق ثم البناء.... كما فى الصورة التالية :



قم بالضغط على أحد منهم و أنتظر لحظات ستجد NETBEANS يقوم بعملية البناء الى ملف تنفيذي و ستجد تطبيقك كامل فى المسار الذى سيوضحه لك NETBEANS كما فى الصورة التالية :

```

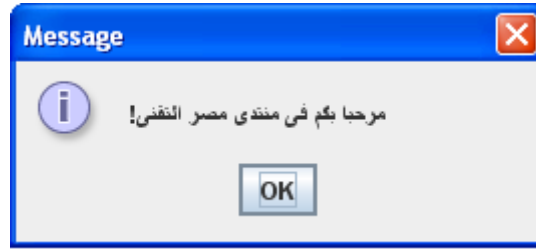
Output - welcome (clean,jar)
init:
deps-clean:
Updating property file: D:\welcome\build\build-clean.properties
Deleting directory D:\welcome\build
clean:
init:
deps-jar:
Created dir: D:\welcome\build
Updating property file: D:\welcome\build\build-jar.properties
Created dir: D:\welcome\build\classes
Created dir: D:\welcome\build\empty
Created dir: D:\welcome\build\generated-sources\ap-source-output
Compiling 1 source file to D:\welcome\build\classes
compile:
Created dir: D:\welcome\dist
Copying 1 file to D:\welcome\build
Nothing to copy.
Building jar: D:\welcome\dist\welcome.jar
To run this application from the command line without Ant, try:
java -jar "D:\welcome\dist\welcome.jar"
jar:
BUILD SUCCESSFUL (total time: 4 seconds)

```

المسار الموجود فيه الملف التنفيذي jar

العملية تمت بنجاح

أدخل المسار الموجود في الملف التنفيذي للتطبيق... ثم قم بفتحة و تشغيله من خلال الضغط عليه ستجد الرسالة قد ظهرت بهذا الشكل :



الفصل الثالث : أنواع البيانات وكيفية تمثيل الثوابت و المتغيرات

فى الدرس السابق تعلمنا سويا كيفية إنشاء تطبيق جافا عن طريق محرر النصوص NOTEPAD و ايضا عن طريق NETBEANS ...

اليوم إن شاء الله تعالى سنتكلم عن درس مهم جدا وهو عن كيفية تمثيل البيانات و العناصر الاساسية و التى تستخدم فى بناء تطبيقات الجافا مثل الثوابت و المتغيرات و غيرهما و وسوف نقوم بشرح هذه المكونات من خلال أمثلة مكتوبة بلغة الجافاوقبل البدء تعالوا نتذكر سويا أنواع البيانات التى ستم تمثيلها!!

أنواع البيانات فى الجافا

تحتوى لغة الجافا الحروف الأبجدية (LETTERS) سواء كانت الحروف الكبيرة (CAPITAL LETTERS) من A إلى Z وكذلك الحروف الصغيرة (SMALL LETTERS) من A إلى Z.بالضافة الى الأرقام العددية (DIGITS) من ٠ إلى ٩

وايضا الحروف الخاصة (SPECIAL CHARACTERS) وهى تلك الحروف التى ليست بأعداد أو بحروف أبجدية ولكنها تكون على هيئة رموز كالآتى(,=,;,;,!,[],!,|,(),%,#,\$,>,/,//,-,+):

لكن تعالى الان عزيزى القارئ تصنف تلك البيانات فى لغة الجافا حسب طولها والسعة التخزينية لها فى الذاكرة كما يلى :-

1_بيانات من النوع المنطقى: BOOLEAN

و يأخذ القيم صفر او واحد ويكون حجمة فى الذاكرة BIT ويستخدم فى حالة المتغيرات التى تكون احدى قيمتين معلومتين مثلا (نعم و لا) أو بمعنى أدق (TRUE OR FALSE) والقيمة الافتراضية فى حالة الاعلان عن متغير وعدم إعطاؤه قيمة تكون FALSE

2_بيانات من النوع: BIT

وهى البيانات التى تكون قيامها اعداد صحيحة سالبة او موجبة محصورة بين - ١٢٨ كحد أدنى و ١٢٨ كحد أقصى مثل (٥, ١٠, -١٠, -١١)

3_بيانات من النوع: CHAR

وهى التى تكون قيمها عبارة عن حرف واحد او قيمة لاحدى متسلسلات الهروب او رقم وهذا الرقم يعبر عن حرف ولكن بنظام الاسكى كود حيث فى هذا النظام كل حرف يمثل برقم معين وحجمها فى الذاكرة.

4_بيانات من النوع: INT

وهي تمثل بقيم صحيحة موجبة أو سالبة محصورة بين -٢١٤٧٤٨٣٦٤٨ كحد أدنى و٢١٤٧٤٨٣٦٤٧ كحد أقصى وهو أكثر أنواع البيانات استخداماً في الجافا.

5_بيانات من النوع: FLOAT

هي تمثل بقيم الاعداد حقيقية اى اعداد بها علامة عشرية او اعداد كسرية او اعداد اسية

6_بيانات من النوع : DOUBLE

وهي تمثل ايضا الاعداد الحقيقية مثل FLOAT إلا أن الفرق هو أن البيانات من نوع DOUBLE تكون كبيرة جدا جدا

و الان أخى الكريم بعد ما تعرفنا على البيانات فى لغة الجافا و أنواعهاتعالوا نتعلم سويا كيفية تمثيلها

أولا : الثوابت:

هي عبارة عن قيم ثابتة يراد الاحتفاظ بها طوال فترة تشغيل البرنامج ولا تتغير قيمتها ابدا و تنقسم الى نوعين:

الاول ثوابت رقمية: NUMERIC CONSTANTS

هي تلك الاعداد ذات القيم الصحيحة والحقيقية التى تحوى تسلسلا من الارقام سواء كانت اعداد سالبة او اعداد موجبة او صفروهي بدورها تنقسم الى :

- **ثوابت رقمية صحيحة... INTEGER** وهي التى تحتوى على ارقام سواء كانت سالبة أو موجبة لكن لا تحتوى على كسور أو علامة عشرية
- **ثوابت رقمية حقيقية .. FLOATING CONSTANT** وهي تلك الاعداد السالبة أو الموجبة لكنها تحتوى على كسور أو علامة عشرية

الثانى ثوابت رمزية: NON-NUMERIC CONSTANTS

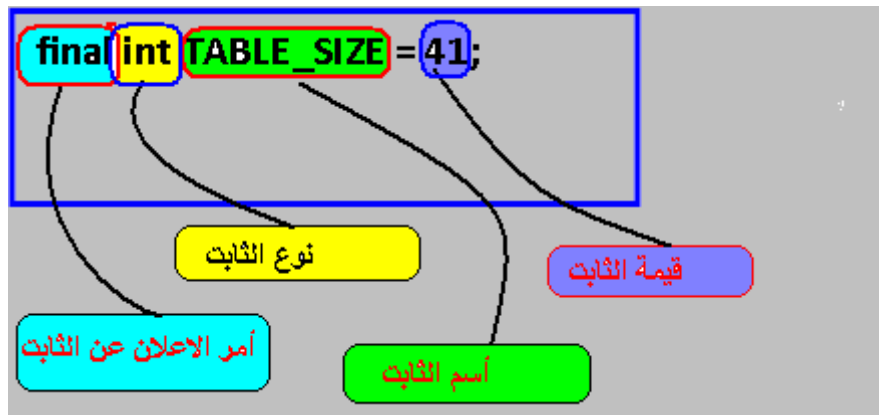
وهي عبارة عن رموز اللغة وتتكون من الحروف والأرقام وتكون بين علامتي تنصيص أو اقتباس ومن الأمثلة على الثوابت الرمزية مثل...(("30+40" - "12345" - "K" - "NAME") وهي بدورها تنقسم الى:

- **ثوابت حرفية.. CHARACTER CONSTANTS** وهي التى تتكون رقم او حرف او رمز من متسلسلات الهروب
- **ثوابت السلسلة STRING CONSTANT** يحتوى ثابت السلسلة على اى عدد من الرموز او الفراغات وتكون محصورة بين علامتى تنصيص . " "

كيفية الاعلان عن الثوابت فى الجافا:

للاعلان عن أى قيمة فى تطبيق جافا على أنها ثابت فأنا نستخدم الامر FINAL وذلك لاجبار الحاسب الالى

بحجز مكان في الذاكرة لحفظ تلك القيمة وانها ستظل ثابتة طوال تنفيذ البرنامج.... ثم بعد نحدد نوع الثابت ثم وضع أسم للثابت



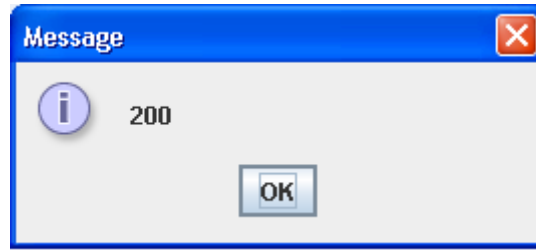
FINAL DATATYPE CONSTANT_NAME =CONSTANT_VALUE

مثال : قم بإظهار رسالة للمستخدم مع تخزين الرقم ٢٠٠ على أنه ثابت ... الحل كما يلي:

FINAL INT EYG=200;

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7         final int eyg=200;
8         JOptionPane.showMessageDialog(null, eyg);
9     }
10 }
```

ويجب أن تلاحظ أي الكريم أننا استخدمنا الامر FINAL متبوعة بنوع البيانات التي سأجعلها ثابتة وبما أن المثال فرض علينا الاعلان عن ثابت رقمي صحيح لذلك حددنا نوعه INT وذلك لان الرقم الذي سوف نخزنه في هذا الثابت رقم صحيح ليس به علامة عشرية .أما مخرجات البرنامج ستكون كما يلي:



مثال آخر.....قم بإظهار رسالة للمستخدم مع تخزين الرقم ٥,٥ على أنه ثابت..الحل

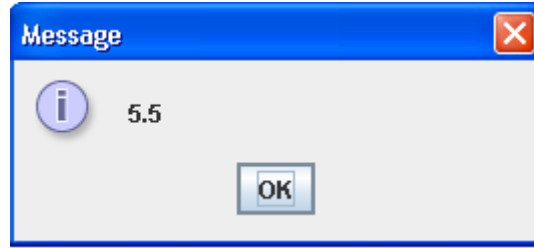
FINAL FLOAT EYG=(FLOAT) 200.25;

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7         float egy=(float) 5.5;
8
9
10
11         JOptionPane.showMessageDialog(null, egy);
12
13     }
14 }
```

ويجب ملاحظة أحي الكريم أننا حددنا نوع الثابت هنا على FLOAT وذلك لأنه رقم يحتوى على علامة عشرية وبالتالي فهو من الأعداد الحقيقية كما أننا في تلك الحالة وضعنا الأمر FLOAT بين قوسين عندما أعطينا قيمة للثابت..... كما يمكنك أيضاً الإعلان عن الأعداد الحقيقية كما يلي:

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7         final double eyg=5.5;
8         JOptionPane.showMessageDialog(null, eyg);
9     }
10 }
```

لاحظ أحي الكريم أننا استخدمنا DOUBLE للإعلان عن الثابت EGY ثم وضعنا قيمة تساوي ٥,٥.... ومخرجات البرنامج في كلا الحالتين كما في الصورة التالية:



ثانيا : المتغيرات :

هى عبارة عن قيم أيضا يتم تخزينها فى ذاكرة الحاسب ولكنها يمكن تغييرها اثناء فترة تشغيل البرنامج وتنقسم الى:

أ -متغيرات رمزية (حرفية)... وهى التى تتضمن الحروف بكافة أشكالها والرموز والفراغات (مسافة فارغة :)

ب - متغيرات عددية... وهى التى تتضمن الارقام سواء إن كانت أرقام صحيحة _الارقام التى بدون علامة عشرية حتى الرقم ٣٢٧٦٧... أو كانت أرقام حقيقية وهى جميع الارقام التى تحتوى على علامة عشرية

كيفية الاعلان عن المتغيرات :

توجد طريقتان للاعلان عن المتغيرات

الطريقة الاولى تكون بهذا الشكل:

```
DATATYPE VARIABLES_NAME;  
VARIABLES_NAME=VARIABLES_VALUE ;
```

Datatype Variables_Name;

نوع المتغير

اسم المتغير

Variables_Name = Variables_Value ;

اسم المتغير

قيمة المتغير

فكما ترون في تلك الطريقة يتم الاعلان عن المتغير ثم بعد ذلك يتم تخصيص هذا المتغير بوضع قيمة له....ويمكنك اختصار تلك الطريقة كما يلي:

Datatype Variables_Name=Variables_Value ;

وهذه هي الطريقة الثانية يتم الاعلان و التخصيص في سطر واحد

أمثلة توضيحية....

```
char a,b;
```

```
a= 'a';  
b='/';
```

في المثال السابق قمنا بالاعلان على عدد ٢ متغير حرفي ثم بعد ذلك خزنا في المتغير الاول a القيمة a ويجب ملاحظة أننا وضعنا القيمة بين علامتي إقتباس كما يمكننا استخدام هذا المثال كما في الشكل التالي:

```
char a="a",b="/";
```

أما الاعلان عن المتغيرات الرقمية الصحيحة تكون بهذا الشكل

```
int a;  
A=100;
```

أو

```
int a=100;
```

ويجب ملاحظة أن قيمة المتغير رقم صحيح ١٠٠ أما اذا كان ذو علامة عشرية فإنه بذلك أصبح عدد حقيقي فيكون الاعلان عنه بهذا الشكل:

```
float x;  
x=5.2;
```

أو

```
float x= 5.2
```

وهنا يجب علينا الانتباه لجملة الإعلان والتخصيص السابقة , float x= 5.2 إلا أن الحاسب يعتبره من النوع double أي يحجز له مكان ٦٤ بت وللتغلب على المشكلة السابقة يتم الإعلان والتخصيص كالتالي:

```
float x= 5.2 f
```

أي يتم وضع حرف (f) بعد الرقم لكي يتم حجز مكان له في الذاكرة مقداره ٣٢ بت وبذلك نكون قد وفرنا في الذاكرة المستخدمة....أما المتغيرات العددية الحقيقية الطويلة Double هي نفس المتغيرات العددية الحقيقية float ولكن يمكن تمثيلها في خمسة عشرة خانة وتكتب على الشكل التالي:

double x;

قواعد تسمية الثوابت و المتغيرات فى الجافا:

1_ يفضل أن يكتب أسم الثابت بالحروف الكبيرة CAPITAL LETTERS فى حين أن أسماء المتغيرات يفضل أن تكون بالاحرف الصغيرة.. SMALL LETTERS ويجب ملاحظة أذى الكريم أننا نقول يفضل وليس يجب معنى ذلك إن لم تلتزم بهذه القاعدة فإن البرنامج سيعمل معك بشكل صحيح.

2_ يمكن أن يبدأ الاسم فى الثوابت و المتغيرات بالحرف أو الشرطة السفلية UNDER SCORE (_) أو علامة الدولار (\$) ولكن لا يمكن بدء التسمية برقم ولكن يمكن أن نضع رقم بعد الحرف. -لا يمكن تسمية المتغيرات أو الثوابت بإحدى الكلمات المحجوزة

العمليات الحسابية و المنطقية

معظم برامج الكمبيوتر تستخدم العمليات الحسابية والشكل التالي يوضح كيفية تمثيل تلك العمليات فى لغة الجافا ولاحظ الفرق بين التعبير فى الجافا و التعبير الجبرى ويمكن تلخيصها فيما يلى:

العملية	العامل	التعبير الجبرى	التعبير بالجافا
جمع	+	$f+7$	$f+7$
طرح	-	$f-7$	$f-7$
ضرب	*	bm	$b*m$
قسمة	/	x	x/y
موديلاس	%	$r \text{ mod } s$	$r\%s$

- علامة + والغرض منها عملية الجمع بين العوامل
- علامة - والغرض منها عملية الطرح بين العوامل
- علامة * والغرض منها ضرب العوامل
- علامة / والغرض منها القسمة بين العوامل وعند استخدام هذا المؤثر نحترس حيث يمكن ان يكون العاملين المستخدمين اعداد صحيحة وينتج عن العملية عدد عشري $b=15$, $a=3.75$ مثال $a/b=3.75$ صحيحة فيكون الناتج عدد صحيح فيكون المتوقع ولن الناتج سيكون غير ذلك حيث ان العاملان $a/b=3.75$ صحيحة فيكون الناتج عدد صحيح فيكون الناتج 3 فقط دون العدد الكسرى او العشرى ولكن ماذا لو ارادنا القسمة الصحيحة فى هذه الحالة يجب ان يكون احد العاملين عدد عشري بمعنى وهى نفس القيم القديمة ولكن اختلف نوعها وفى هذه الحالة يكون $b=4$, $a=10$ ان تكون $10,0$ الناتج $3,75$ لذلك احترس عزيزى القارئ عند استخدامك هذا المؤثر.

- علامة % ويكون ناتج هذه العملية هو باقى القسمة ونلاحظ هنا انه يجب ان تكون العوامل اعداد صحيحة لضمان صحة الناتج وتكون اشارة الناتج هى نفس اشارة العامل الايسر...

مثالقم بعرض رسالة بها مجموع العددين ٥ و ٦.....الحل كما يلى:

تكبير الصورة معاينة الأبعاد الأصلية.

```

1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[]
7
8         int a=5;
9         float b=6.5f;
10        float c=a+b;
11        JOptionPane.showMessageDialog(null,"المجموع هو "+ c);
12
13    }
14 }

```

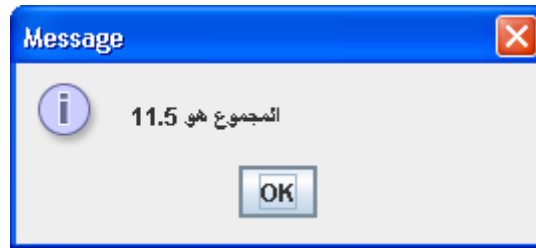
الاعلان عن المتغير الاول

الاعلان عن الرقم الثانى

الاعلان عن المتغير الذى سيتم تخزين نتيجة العملية الحسابية فيه لحين عرضها

تم وضع علامة + وذلك لعرض جملة (المجموع هو) بالإضافة الى نتيجة العملية الحسابية

مخرجات البرنامج كما بالشكل التالى:



عامل الزيادة وعامل النقصان:

تمندنا لغة الجافا بعامل الزيادة ++ ويستخدم لزيادة المتغير أو الثابت بمقادير واحد... فمثلا إذا كان قيمة المتغير $a = 5$ فإذا وضعت عامل الزيادة قبل المتغير هكذا $a++$ فمعنى ذلك أن قيمة هذا المتغير أصبحت ٦..... وعامل النقصان -- ويستخدم لنقص المتغير أو الثابت بمقدار واحد صحيح فى المثال السابق إذا وضعت عامل النقصان قبل المتغير هكذا -- فمعنى ذلك أن قيمة هذا المتغير أصبحت ٤...

ويمكن وضع تلك العوامل قبل المتغير او بعده ولكن بالتأكيد هناك فرق بينهم

لاحظ معى اذا كانت $a=5$ وكانت هناك معادلة بهذا الشكل... $b=(a++)*3$ فإن الناتج سيكون $b=15$ ولكن اذا استخدمنا المتغير a فى عملية أخرى سيكون قيمته ٦.... أما اذا كانت المعادلة بهذا الشكل $b=(++a)*3$ فإن الناتج سيكون $b=18$ هنا الزيادة تمت على المتغير a و اصبحت قيمته ٦ قبل العملية الحسابية

اذن اذا كان عامل الزيادة أو النقصان قبل المتغير فنتم الزيادة أو النقص فى نفس السطر او فى نفس الخطوة فى البرنامج لكن اذا كان بعد المتغير فنتم الزيادة بعد الانتهاء من هذه الخطوة...

الفصل الرابع : طرق الادخال و الاخراج فى الجافا

فى هذا الدرس إن شاء الله تعالى سنتعلم سويا ونتعرف على طرق ادخال و اخراج البيانات فى لغة الجافا ..وهذا الدرس مهم جدا حيث لا يخلو أي برنامج من طريقة لادخال البيانات و استخراجها وهي الوسيلة الوحيدة التي يستطيع فيها المستخدم إدخال القيم عن طريق لوحة المفاتيح للحاسب ، حتى يقوم بمعالجة هذه القيم سواء كان البرنامج (برنامج حسابات – قاعدة بيانات.).

والحقيقة أن لغة الجافا تحتوي على أكثر من طريقة لإدخال البيانات و أخراجها منها ما هو مناسب للتطبيقات الرسومية (تطبيقات الويندوز)ومنها ما هو مناسب للتطبيقات غير الرسومية _تطبيقات الدوس (Console Application) و سوف يتم شرح تلك الطرق فى البيتين.

أولا :طريقة إدخال وإخراج البيانات فى التطبيقات الرسومية:

ويتم ذلك عن طريق اوامر الكلاس المسمى JOptionPane حيث يمدنا بنوافذ حوارية مختلفة تساعدنا على إدخال و اخراج البيانات من والى البرنامج من خلال تلك النوافذ...

1_طريقة إدخال البيانات بأستخدام: JOptionPane

أولا : يجب تحميل الكائن أو الكلاس المسمى JOptionPane من الحزمة المسماة **javax.swing** وهذه الحزمة تحتوي على كثير من الكائنات الاخرى _ستيم شرحها فى حينها _ويتم ذلك عن طريق الكود التالى:

الرمز:

```
import javax.swing.JOptionPane;
```

ثانيا : يجب تعريف المتغيرات التي سيتم تخزين ما يقوم المستخدم بإدخاله عن طريق لوحة المفاتيح ولا بد أن تكون من النوع النصى String وذلك لان كل بيانات يتم ادخالها من المستخدم بواسطة هذا الأمر يعتبر.... String

الرمز:

```
String name;
```

ثالثا :نقوم بإظهار المربع الحوارى لادخال البيانات من المستخدم ويتم تخزينها فى المتغير String

الرمز:

```
name=JOptionPane.showInputDialog(" enter name ");
```

رابعا :إنهاء البرنامج وذلك بإستخدام الامر

الرمز:

`System.exit(0);`

و كقاعدة عامة يجب استخدام هذا الامر لانتهاء التطبيق في التطبيقات الرسومية.... وذلك لان التطبيق في حالة استخدام هذا الامر سيكون نشط في ذاكرة الجهاز.....

اذن الكود العام لادخال البيانات باستخدام الكلاس `JOptionPane` سيكون كما في الشكل التالي:

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7
8         String name;
9
10        name=JOptionPane.showInputDialog("من فضلك أدخل اسمك");
11        System.exit( 0 );
12    }
13 }
```

الآن انتهينا من كيفية ادخال البيانات بتلك الطريق لكن ماذا لو كنا نريد من المستخدم إدخال بيانات رقمية و ليست نصية و نريد استخدامها كأرقام في عمليات حسابية فقد أخبرنا سابقا بأن كل شيء يتم أخذه من المستخدم بواسطة هذا الأمر (يعتبر `String` سيتم شرحها لاحقاً بالتفصيل) ، بمعنى أن البرنامج سيعتبرها كلمات أو جمل أو نصوص، ولذلك لن يتمكن من التعامل معها في المعادلات الرياضية والحسابية حتى لو أدخل المستخدم فيها أرقام. بمعنى لو أدخل المستخدم رقم سيتم تخزينه في البرنامج بين علامتي التنصيص بهذا الشكل: "٩٩"

وجوده بين علامتي التنصيص يعني انه لايعتبر من أنواع المتغيرات التي يمكن التعامل معها في المعادلات الرياضية و الحسابية مثل `int` و `double` و `byte` وغيرها.....
ولو لم نقم بعملية تحويلها إلى بيانات رقمية ، وحاولنا أن نقوم بعمليات حسابية لها مباشرة ، سيحدث خطأ في البرنامج ، لأنه لن يستطيع التعامل مع الـ `String` في العمليات الحسابية.

لذلك توجد أوامر لتحويل المدخلات من بيانات نصية إلى أنواع البيانات الرقمية التي نريدها.... وذلك على النحو التالي:

```
byte b = Byte.parseByte("123");
short s = Short.parseShort("123");
int i = Integer.parseInt("123");
long l = Long.parseLong("123");
float f = Float.parseFloat("123.4");
double d = Double.parseDouble("123.4e10");
```

كيفية تحويل النصوص المدخلة الى أنواع من متغيرات رقمية

وهذا الشكل يوضح الاوامر المستخدمة لتحويل القيم النصية المدخلة بواسطة المستخدم الى قيم رقمية و ذلك لاستخدامها في العمليات الحسابية.....

مثال : بقم بعرض رسالة للمستخدم لآخذ بيانات نصية..... الحل كما يلي:

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7
8         String no;
9
10        no=JOptionPane.showInputDialog("من فضلك أدخل الرقم");
11        System.exit( 0 );
12    }
13 }
```

ستجد أذى الكريم أن هذا المثال هو نفس المثال السابق بدون أى تغير.... وفى تلك الحالة أى نص أو حرف أو حتى أرقام أدخلها المستخدم فأنها سوف يتم حفظها فى المتغير (no) على أنها بيانات نصية و ليست رقمية.... بمعنى لو أدخل المستخدم الرقم (5) ستعبر البرنامج أن تلك القيمة نصية و لن تستطيع استخدامها فى العمليات الحسابية وسيحدث خطأ للبرنامج لو استخدمتها.....

مثال آخر.... بقم بعرض رسالة للمستخدم لآخذ بيانات رقمية..... الحل كما يلي:

```
1
2 package m;
3 import javax.swing.JOptionPane;
4 public class M {
5
6     public static void main(String[] args) {
7
8         String no;
9         int number;
10        no=JOptionPane.showInputDialog("من فضلك أدخل الرقم");
11        number=Integer.parseInt("no");
12        System.exit( 0 );
13    }
14 }
```

لاحظ أى الكريم... التغير الذى حدث فى المثال الثانى عن المثال الاول... فقط قمنا بإضافة سطرين جديدين... الاسطر الاول هو الاعلان عن المتغير number على أنه من نوع بيانات رقمية صحيحة... int أما السطر الاخر هو أننا قمنا بتحويل القيمة النصية المخزنة فى المتغير no الى بيانات رقمية صحيحة من نوع int و تم تخزينها فى المتغير number..... وهكذا نكون قد حولنا القيمة البيانات النصية من المستخدم الى بيانات رقمية و تستطيع الان إجراء العمليات الحسابية عليها....

كما يرجى ملاحظة اننا حولنا القيم النصية الى قيم رقمية من نوع... int أما اذا كان نريد أنواع رقمية أخرى غير ذلك فاذا كانت البيانات المتوقعة ادخالها من المستخدم من نوع byte اذن كود التحويل سيكون كما يلي:

الرمز:

```
number = Byte.parseByte("no");
```

وإذا كانت البيانات المتوقعة أدخلها من المستخدم من نوع long اذن كود التحويل سيكون كما يلي :

الرمز:

```
number= Long.parseLong("no");
```

وإذا كانت البيانات المتوقع أدخلها من نوع Float اذن كود التحويل سيكون كما يلي:

الرمز:

```
number = Float.parseFloat("no");
```

وهكذا بالنسبة للبيانات من نوع **double** أو **short** وبما أن لغة الجافا لحالة الاحرف فهناك خطأ شائع وهو عدم الالتزام بنفس الجمل السابقة من حيث الاحرف الكبيرة و الصغيرة.

2_ طريقة أخراج البيانات بأستخدام: JOptionPane

طريقة اخراج البيانات بأستخدام الكلاس المسمى JOptionPane كنا قد تعودنا على أستخدامها في الدروس السابقة...وقلنا أننا اذا أردنا أن نخرج بيانات للمستخدم على هيئة نافذة رسالة يكون عن طريق جملة الاخراج التالي:

الرمز:

```
JOptionPane.showMessageDialog(null, "");
```





وكانت هذه ابسط رسالة يمكن أظهارها للمستخدم و تحتوى بيانات ..لكن يمكننا اظهارها بصورة أكثر تنسيق باستخدام جملة الاخراج التالية:

```
showMessageDialog(null, message, title, messageType)
```

وجملة الاخراج السابقة فكما هو واضح تتطلب أربع مدخلات:

- ❖ المدخل الاول : الكلمة المحجوزة **null** وفائدتها وضع صندوق الحوار في وسط الشاشة.
- ❖ المدخل الثانى **message**: نص الرسالة الذى سوف يظهر للمستخدم ويجب وضع علامتى اقتباس على النص المراد أخراجة للمستخدم
- ❖ المدخل الثالث **title**: عنوان الرسالة الذى سوف يظهر في سطر العنوان للرسالة
- ❖ المدخل الرابع **messageType**: وهو نوع الرسالة حيث يوجد مجموعة من الرموز التى التى يمكن أظهارها في صندوق الحوار التى تساعد المستخدم في معرفة نوع صندوق الحوار و الرسالة التى تظهر فيه

❖ والشكل التالي يوضح تلك الرموز:

الوصف	الرمز	نوع رسالة صندوق الحوار
عرض صندوق حوار يبين رسالة خطأ للمستخدم		JOptionPane. ERROR_MESSAGE
عرض صندوق حوار مع رسالة للمستخدم		JOptionPane. INFORMATION_MESSAGE
رسالة تحذيرية للمستخدم		JOptionPane. WARNING_MESSAGE
سؤال للمستخدم يجب الإجابة عليه بنعم أو لا		JOptionPane. QUESTION_MESSAGE
يظهر رسالة في الصندوق بدون رموز	لا يوجد رمز	JOptionPane. PLAIN_MESSAGE

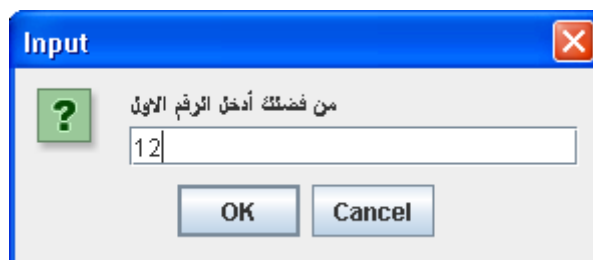
مثال : قم بعمل برنامج يقوم بحساب عددين يدخلهم المستخدم و إظهار الناتج في مربع حوارى.....الحل:

```

1 package m;
2 import javax.swing.JOptionPane;
3 public class M {
4
5     public static void main(String[] args) {
6
7         String no1,no2;
8         int number1,number2;
9         long total;
10
11         no1=JOptionPane.showInputDialog("من فضلك أدخل الرقم الاول");
12         no2=JOptionPane.showInputDialog("من فضلك أدخل الرقم الثاني");
13
14         number1=Integer.parseInt(no1);
15         number2=Integer.parseInt(no2);
16
17         total=number1+number2;
18         JOptionPane.showMessageDialog(null,"الناتج هو" + total, "egy_tech", 2);
19
20         System.exit( 0 );
21     }
22 }

```

مخرجات البرنامج:





شرح الكود المستخدم في البرنامج:

السطر رقم ٧ : قمنا بتعريف متغيرين no1, no2 من نوع نصي String وذلك لتخزين القيم التي يتم أخذها من المستخدم... وطبعاً قلنا أن تلك القيم يعتبرها البرنامج من النوع النصي حتى و أن كانت أرقام

السطر رقم 8 : قمنا بتعريف متغيرين number1, number2 من النوع int وذلك لتخزين القيم المدخلة من المستخدم فيهما بعد تحويلها إلى قيم رقمية من نوع int.

السطر رقم ٩ : قمنا بالإعلان عن المتغير total وذلك لتخزين ناتج العملية الحسابية فيه..

السطر العاشر و الحادي عشر : قمنا بإظهار مربعين أو رسالتين أدخل للمستخدم عن طريق الامر showDialog للكلاس المسمى JOptionPane وذلك لاختار القيم من المستخدم و تخزينها في المتغيرين no1, no2 على التوالي.

السطر الرابع عشر و الخامس عشر : قمنا بتحويل القيم المخزنة في المتغيرين no1, no2 إلى قيم رقمية من نوع int عن طريق الامر Integer.parseInt وتخزين ناتج التحويل في المتغيرين number1, number2 الذي قمنا بالإعلان عنهم سابقاً...

السطر السابع عشر : جعلنا المتغير total يساوي قيمة المتغير number1 بالإضافة إلى قيمة المتغير number2

السطر الثامن عشر : أظهرنا مخرجات البرنامج عن طريق مربع حوارى باستخدام الامر showMessageDialog للكلاس المسمى JOptionPane ويرجى ملاحظة أننا استخدمنا المدخلات الأربعة سالفة الذكر بين قوسين في إظهار المربع الحوارى... فاستخدمنا المدخل (null) لإظهار المربع الحوارى في منتصف الشاشة... و مدخل نص الرسالة وفي هذا المثال هو " الناتج هو + total " وهنا تم استخدام العامل (+) وذلك للاحاق القيمة المخزنة في المتغير total بعد النص "الناتج هو " ثم بعد ذلك مدخل عنوان الرسالة وهو في مثالنا هذا..... "egy_tech" أما الرقم ٢ الموجود في الكود هو المدخل الرابع الذى يحدد نوع أيقونة الرسالة وقد تم تغييرها في مثالنا هذا كما هو ظاهر في مخرجات البرنامج...

ثانياً: طرق الإدخال و الإخراج في البيئة غير الرسومية...

1_ طريقة أخراج البيانات في البيئة غير الرسومية:

الطريقة التي تستخدم لإخراج البيانات في البيئة غير الرسومية تسمى Standard output ويتم ذلك عن طريق استخدام الكلاس System و ذلك باستخدام الكود التالي:

الرمز:

```
System.out.print("Egy_tech!");
```

ويجب ملاحظة أن حرف S كإبتال في كلمة... System والنص المطلوب عرضة Egy_tech في قوسين وبين علامتي إقتباس... ونلاحظ أنه عند تنفيذ هذا البرنامج ستظهر العبارة (Egy_tech) على سطر واحد.... أما إذا أردنا أن تكون المخرجات على سطرين مختلفين فيتم إضافة حرفي (ln) على العبارة print ومعناها الانتقال إلى سطر جديد (new line) ويتم ذلك في السطر الثالث كالتالي:

الرمز:

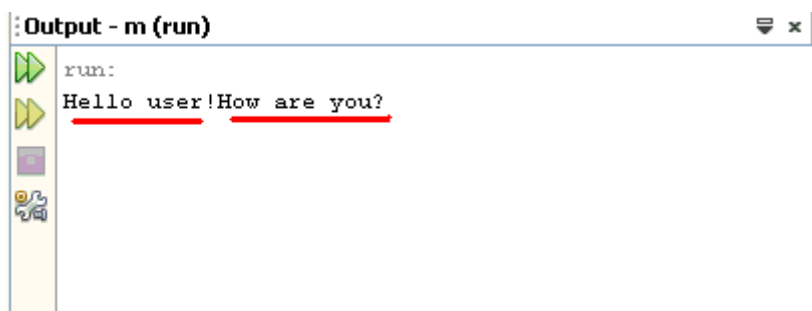
```
System.out.println("Egy_tech");
```

الفرق بين الأول والثاني ، هو أن الثاني سيجعل أمر الكتابة الذي بعده يكون في سطر جديد ، ولكن الأول لن يقوم بهذا. مثلاً لو كتبنا هذه الأمرين:

الرمز:

```
System.out.print("Hello user!");  
System.out.print("How are you?");
```

ستكون النتيجة كذلك

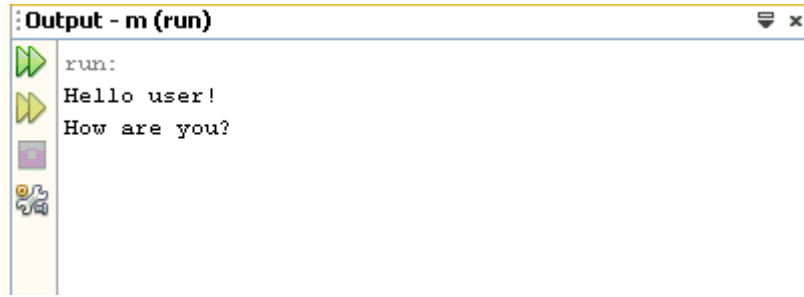


فكما تلاحظ أخي الكريم أن الجملتين في سطر واحد.... أما إذا استخدمنا جملة الإخراج كما يلي :

الرمز:

```
System.out.println("Hello user!");  
System.out.println("How are you?");
```

ستكون النتيجة كما يلي:



```
Output - m (run)
run:
Hello user!
How are you?
```

ويجب ملاحظة الفرق هنا... فقد تم كتابة الجملتين كل جملة في سطر.... وهذا هو الفرق بين `println` و `print` في إخراج البيانات.... ويمكن تنفيذ نفس شكل المخرجات السابقة بسطر واحد وذلك عن طريق إضافة `(\n)` وتعني الانتقال إلى سطر جديد ويكون شكل البرنامج كالتالي:

الرمز:

```
System.out.print("Hello user!\n How are you?");
```

فتكون المخرجات نفس المخرجات السابقة بهذا الشكل



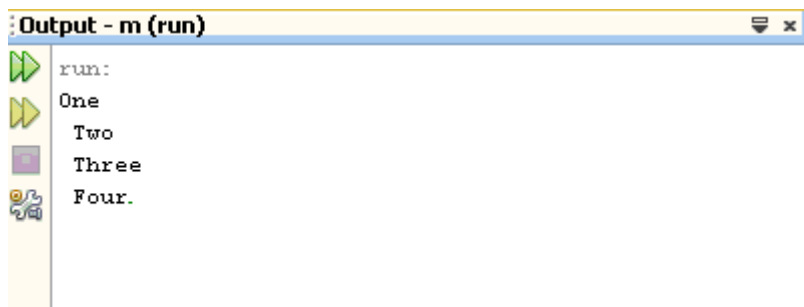
```
Output - m (run)
run:
Hello user!
How are you?
```

أما إذا أردنا طباعة عدة أسطر متتالية بأمر طباعة واحد فيتم ذلك بتكرار `(\n)` كالتالي:

الرمز:

```
System.out.print ("One\n Two \n Three \n Four") ;
```

فتكون المخرجات كالتالي



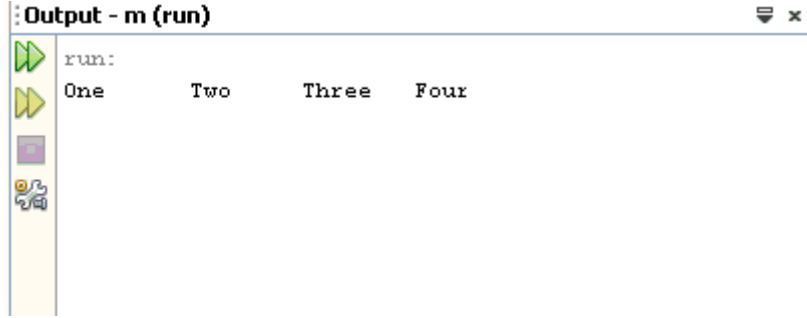
```
Output - m (run)
run:
One
Two
Three
Four.
```

أما اذا أردنا أن تكون المخرجات على مسافات أفقية متساوية فأنا نستخدم (t\) كالتالي:

الرمز:

```
System.out.print ("One\t Two \t Three \t Four");
```

وتكون المخرجات كالتالي:



وكل هذه الاوامر تسمى **متسلسلات الهروب** والتي يتم إضافتها لجملته للطباعة للحصول على عدة اشكال للمخرجات.... واليكم الجدول التالي الذي يوضح متسلسلات أخرى:

الحرف الخاص	الوصف
\n	سطر جديد. يضع المؤشر في بداية السطر التالي
\t	مسافة أفقية. تحريك المؤشر مسافة معينة إلى النقطة التالية في السطر
\r	carriage return. يضع المؤشر في بداية السطر الحالي ولا يتقدم إلى السطر التالي ، وأي حرف يطبع يتم طباعته على حرف سابق تم كتابته في نفس السطر
\\	شرطة خلفية. إظهار \" في الخرج
\"	علامة تنصيص مزدوجة. إظهار علامة التنصيص المزدوجة

2_ طريقة إدخال البيانات في البيئة غير الرسومية:

من الطرق التي تستخدم في إدخال البيانات في لغة الجافا هي استخدام الكلاس ([Java.util.Scanner](#)) ، وهي قوية جداً حيث أنه يمكنها ليس فقط قراءة أسطر الإدخال التي يقوم المستخدم بكتابتها، بل يمكنها أيضاً قراءة محتوى الملفات... الكلاس Scanner وكما يدل على ذلك اسمها، فيمكن أن نترجمها بالعربي إلى الماسح، وكلنا نعرف أداة الماسح الضوئي التي تقوم بمسح وثيقة معينة وتحويلها لوثيقة رقمية. هذه الكلاس تقوم بنفس العمل ! فهي تقوم بمسح كل ما تمت كتابته من طرف المستخدم ليتم تخزينه في متغير للإشغال به في البرنامج.... الطريقة هي سهلة الاستخدام وذلك لأن جميع وظائفها معرفة من طرف الجافا مسبقاً يكفي فقط تتبع الخطوات البسيطة المشروحة هنا.

أولاً: يجب استدعاء الكلاس Scanner وهي الكلاس التي تحتوي على وظائف القراءة من سطر الإدخال من مكتبة الجافا بالأمر التالي:

الرمز:

```
import java.util.Scanner;
```


ثانياً: نقوم بتعريف كائن له نفس خصائص الكلاس Scanner ليتم تخزين فيه ما قام المستخدم بكتابته عن طريق لوحة المفاتيح و القاعدة العامة لتعريف هذا الكائن تكون كما في الشكل التالي:

```
Scanner variable = new Scanner(System.in) ;
```

اي اسم متغير
(a , b , sum ,.....)

فمثلاً إذا أردت أن أنشئ الكائن `input` له نفس خصائص الكلاس `Scanner` يكون الامر على النحو التالي:

الرمز:

```
Scanner input = new Scanner(System.in);
```

ثالثاً: نقوم بقراءة ما أدخله المستخدم على حسب ما نود تخزينه من بيانات... فمثلاً إذا كنا نريد قراءة رقم صحيح من نوع `int`... فإننا سنستخدم الكود التالي للقراءة:

الرمز:

```
input.nextInt();
```

أما إن كانت نوع البيانات غير ذلك.... وكما نعلم هناك عدة أنواع من الأعداد (أعداد طبيعية، عشرية ...) ولكل نوع هناك نوع بيانات معين (كما رأينا في الدرس السابق : أنواع البيانات والمتغيرات). هذه الأعداد، كما يتم تخزينها في نوع معين من أنواع البيانات، يتم أيضاً قراءتها بنوع معين من أنواع البيانات... فالأعداد الطبيعية يتم قراءتها بواسطة السطور التالية:

- 1.nextInt
- 2.nextByte
- 3.nextShort
- 4.nextLong

:

أما الأعداد العشرية يمكن قراءتها بالسطور التالية

- 1.nextFloat
- 2.nextDouble

أما قراءة الكلمات تكون بهذا الامر `nextLine`

رابعا: يفضل دائماً أن نقوم بإنهاء هذا النوع من الكلاس عندما ننتهي من استعمالها بالسطر التالي:

الرمز:

```
input.close();
```

يجب مراعاة كتابة Scanner بحرف S كبير وكذلك كتابة nextInt ب | كبيرة مع عدم نسيان الفاصلة المنقوطة في سطر تعريف الكائن وكذلك في سطر القراءة بواسطة nextInt

مثال توضيحي: أكتب برنامج لجمع رقمين على أن يتم إدخال الرقمين من لوحة المفاتيح ثم طباعة الناتج على الشاشة.

الحل:

```
1 import java.util.Scanner ; ← استدعاء المكتبة التي تحتوي على جملة الإدخال
2 public class magdy{
3 public static void main(String s[] ){
4 int a,b,sum; ← الإعلان عن ثلاث متغيرات
5 Scanner Keyboard=new Scanner(System.in); ← تعريف كائن عملية الإدخال
6 System.out.println("Enter first number");
7 a=Keyboard.nextInt(); ← هنا يتم قراءة الرقم الأول وتخزينه في المتغير a
8 System.out.println("enter seconde number");
9 b=Keyboard.nextInt(); ← هنا يتم قراءة الرقم الثاني وتخزينه في المتغير b
10 sum=a+b;
11 System.out.println("the sum is="+sum);
}
```

الشرح:

❖ في السطر الأول تم استدعاء المكتبة التي تحتوي على جملة الإدخال Scanner

1 import java.util.Scanner ;

❖ أما في السطر الرابع فقد تم الإعلان عن ثلاث متغيرات من النوع integer وهي المتغير a لتخزين الرقم الأول والمتغير b لتخزين الرقم الثاني والمتغير sum لتخزين ناتج عملية الجمع .

4 int a,b,sum;

❖ أما السطر الخامس تعريف كائن له نفس خصائص الكلاس سكانر لاستقبال مدخلات من لوحة المفاتيح ولقد تم تسميته Keyboard ويمكن تسميته بأي اسم تريده ويتم فيه تخزين القيمة المدخلة مؤقتا تمهيدا لنقلها لمتغير آخر سيكون هنا a أو b ولاحظ كلمة (System.in) أصبح بجوارها كلمة in دلالة على عملية الإدخال

5 Scanner Keyboard=new Scanner(System.in);

❖ أما السطر السادس فوظيفته هي طبع رسالة على الشاشة تخبر المستخدم بإدخال الرقم الأول.

❖ أما في السطر السابع فيتم قراءة ما تم إدخاله في المخزن Keyboard و وضعها في المخزن a.

7 a=Keyboard.nextInt();

- ❖ أما في السطر الثامن فهو يكرر العملية لطلب الرقم الثاني.
- ❖ وفي السطر التاسع يتم إدخال الرقم الثاني كما سبق ولكن يتم تخزينه هذه المرة في المخزن b.
- ❖ أما السطر العاشر فيتم فيه عملية الجمع ووضع الناتج في المخزن sum.
- ❖ وفي السطر الحادي عشر يتم طباعة قيمة الجمع على الشاشة.

أما عيب هذه الكلاس (Scanner) أنها لا تحتوي على وظيفة لقراءة حرف واحد!..... لذلك ولقراءة حرف واحد يجب قراءة السطر بأكمله ثم بعد ذلك قراءة الحرف الأول من الكلمة أو الجملة التي قام المستخدم بادخالها. وذلك يتم كما في المثال التالي:

```
01. // هنا يتم استدعاء الكلاس سكائر
02. import java.util.Scanner;
03.
04. public class Input_char {
05.
06.     /**
07.      * @author Ayoub El Hobbi
08.      */
09.     public static void main(String[] args) {
10.
11.         // نقوم بتعريف كائن اسمه إنبوت له نفس خصائص الكلاس سكائر
12.         Scanner input = new Scanner(System.in);
13.
14.         // نطلب من المستخدم إدخال الحرف الأول من اسمه
15.         System.out.print("what is the first character of your name ? : ");
16.
17.         // نقوم بتخزين ما قام بكتابته في متغير ناچ
18.         String name = input.nextLine();
19.
20.         // تأخذ الحرف الأول ما أدخله المستخدم .. فحقي وإن أدخل اسمه بأكمله
21.         char firstChar = name.charAt(0);
22.
23.         // وهنا تتم طباعته
24.         System.out.println("the letter is : " +firstChar);
25.         input.close();
26.     }
27.
28. }
```

كما رأينا في هذا المثال فحتى وإن قام المستخدم بكتابة اسمه بأكمله يمكننا تخزين فقط الحرف الأول من الكلمة، وكما يمكن تخزين موضع الحرف الأول من الكلمة، يمكن أيضاً تخزين الموضع الثاني، الثالث ... وذلك بتعويض رقم ٠ في charAt بمثل رقم ١....لنأخذ مثلاً اسم Ayoub أول حرف من هذا الاسم هو a وهو يحمل الرقم ٠ في charAt، ثاني حرف يحمل الرقم ١، وهكذا ... يمكنك تجربة ذلك بنفسك

الفصل الخامس : أدوات التحكم البنائي

أهلاً وسهلاً أخواني الكرام في الدرس الخامس من دروس منتدى مصر التقني لتعلم الجافا الرسومية بأستخدام NetBeans ... درسنا اليوم هام جداً فلا يكاد يخلوا أى برنامج من أحد أدوات التحكم البنائي مثل الجمل الشرطية و حلقات التكرار....

أولاً : الجمل الشرطية:

ويقصد بها تنفيذ أمر ما عند تحقق شرط معين... فلننصّر مثلاً أننا نكتب برنامجاً صغيراً جداً. يخزن رقماً في متغير، ثم يقارن هذا المتغير بالرقم عشرة، فإذا كان المتغير أصغر نعرض على الشاشة جملة معينة. وإذا كان أقل من عشرة، سنعرض على الشاشة جملة أخرى.....

1_جملة IF وتنقسم الى عدة صيغ وذلك على النحو التالي:

- **الصيغة الأولى لجملة if** و تأخذ الصيغة العامة التالية:

```
الشرط
if (expression) {
    جملة1 // statement1;
    جملة2 // statment2;
}
```

كما هو موجود في الشكل السابق نبدأ جملة الشرط بالكلمة (if) و بعدها نفتح قوسين دائريين نضع في داخلها شرطاً ما. بعدها نفتح قوس مجموعة نضع في داخله ما نريد تنفيذه في حالة تحقق الشرط فقم بتنفيذ الجمل التي بين القوسين وفي حالة عدم تحقق فلا يتم تنفيذ جملة (if) وإنما استمر في تنفيذ بقية جمل البرنامج من بعد تخطي جملة (if)

ويجب ملاحظة أن تلك الصيغة أحتوت على عدة سطور برمجية يتم تنفيذها في حالة تحقق الشرط فتم نفتح قوسين دائريين أما في حالة تنفيذ جملة واحدة فقط بعد جملة (if) فإنه يمكن الاستغناء عن الأقواس وفي هذه الحالة تنتهي جملة الشرط بالفاصلة المنقوطة (؛) كما يلي:

```
الشرط
if (expression)
    statement1;
تنتهي جملة الشرط عند الفاصلة المنقوطة
```

و كما هو معلوم فإن الشرط يجب أن يحتوى على أحد المعاملات المنطقية التالية:

المؤشر	المعنى
==	يساوي
!=	يخالف
<=	اصغر او يساوي
>=	اكبر او يساوي
>	اكبر قطعاً
<	اصغر قطعاً

مثال للتوضيح : المطلوب كتابة برنامج يقوم بفحص رقم يتم إدخاله من لوحة المفاتيح وليكن (x) بحيث إذا كان أكبر من الصفر يقوم بطباعة العبارة (الرقم موجب)

الحل.....

```

1 package m;
2 import javax.swing.JOptionPane;
3 public class M {
4
5     public static void main(String[] args) {
6
7         String no;
8         int number;
9         no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10        number=Integer.parseInt(no);
11        if (number>0)
12            JOptionPane.showMessageDialog(null, "الرقم موجب", "Egy_Tech", 2);
13        System.exit( 0 );
14    }
15 }

```

شرح الكود:

- السطر رقم ٧ : قمنا بالاعلان عن المتغير no لتخزين ما يتم إدخاله من المستخدم فيه...وطبعاً كما قلنا في دروس سابقاً أنه من نوع نصي string
- السطر رقم ٨ : قمنا بالاعلان عن متغير رقمي من نوع int يسمى number و ذلك لتخزين القيمة الموجودة في المتغير no بعد تحويلها الى قيمة رقمية.
- السطر رقم ٩ : قمنا بإظهار نافذ للمستخدم لأخذ القيم منه و تخزينها في المتغير. no
- السطر رقم ١٠ : تحويل القيمة النصية المخزنة في المتغير no الى قيم رقمية و تخزينها في المتغير. number
- السطر رقم ١١ : استخدمنا جملة الشرط if للتأكد من أن القيمة المدخلة من المستخدم أكبر من الصفر...
- السطر رقم ١٢ : اذا كانت جملة الشرط صحيحة_بمعنى أن الرقم المدخل من قبل المستخدم أكبر من الصفر_يتم أظهار رسالة للمستخدم نصها" الرقم موجب"

وفي هذا المثال يجب ملاحظة أننا لم نستخدم الاقواس في جملة...if أما اذا كانت جملة if تحتوي على أكثر من سطر برمجى يجب وضع الاقواس....

مثال للتوضيح : كرر المثال السابق مع أظهار عدد ٢ رسالة الاولى لاطهار الرقم المدخل من المستخدم.... و الاخرى لعرض نوعية موجب أو سالب.

الحل:

```

1 package m;
2 import javax.swing.JOptionPane;
3 public class M {
4
5     public static void main(String[] args) {
6
7         String no;
8         int number;
9         no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10        number=Integer.parseInt(no);
11        if (number>0) {
12            JOptionPane.showMessageDialog(null, "الرقم المدخل هو" + number, "Egy_Tech", 2);
13            JOptionPane.showMessageDialog(null, "الرقم موجب", "Egy_Tech", 2);
14        }
15
16    }
17
18 }

```

نلاحظ أن جملة الشرط احتوت على أكثر من سطر
ولذلك وضعنا الأقواس

- الصيغة الثانية لجملة if و تأخذ الصيغة العامة التالية:



وكما تلاحظ من الشكل السابق أحي الكريم ...جملة if...else عبارة عن جملة إختيار يمكنك من القيام بعمل ما إذا تحقق شرط معين، و القيام بعمل آخر إذا لم يتحقق الشرط.

في مثالنا السابق، يمكننا أن نوسع الفكرة بحيث يقوم البرنامج بعرض رسالة بأن الرقم موجب إذا كان أكبر من الصفر و إظهار رسالة بأن الرقم سالب إن كان غير ذلك. كما يلي:

```

1 package m;
2 import javax.swing.JOptionPane;
3 public class M {
4
5     public static void main(String[] args) {
6
7         String no;
8         int number;
9         no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10        number=Integer.parseInt(no);
11        if (number>0)
12            JOptionPane.showMessageDialog(null, "الرقم موجب", "Egy_Tech", 2);
13        else
14            JOptionPane.showMessageDialog(null, "الرقم سالب", "Egy_Tech", 2);
15
16    }
17 }

```

و يمكننا في البرنامج السابق حذف الأقواس الموجودة بعد جملة if , وكذلك حذف الأقواس الموجودة بعد else وذلك لوجود جملة طباعة واحدة بعد كلاً منهما.

- **الصيغة الثالثة لجملة IF** و تأخذ الشكل التالي :

```

if(expression 1){
    // اذا كان الشرط الاول صحيح يتم تنفيذ الكود الذى هنا/
}else if(expression 2){
    // اذا كان الشرط الثانى صحيح يتم تنفيذ الكود الذى هنا/
}else if(expression ) {
    // اذا كان الشرط الثالث صحيح يتم تنفيذ الكود الذى هنا/
}else {
    // يتم تنفيذ الجملة التى هنا فى حالة عدم صحة الشروط السابقة/
}

```

وكما ترى من الصيغة السابقة فان جملة if...else if...else يستخدم فى التأكد من شروط عدة فإذا كان الشرط الاول صحيح يتم تنفيذ الشفرة التى بين القوسين ثم نستخدم التعليمة else if للتأكد من صحة الشرط الثانى فان كان صحيح يتم تنفيذة وهكذا أما اذا كانت كل الشروط غير صحيحة أو لم تتحقق فأننا نستخدم التعليمة else لتنفيذ كود معين

مثال للتوضيح :

```

public class Test {
    public static void main(String args[]){
        int x = 30;
        if( x == 10 ){
            System.out.print("Value of X is 10");
        }else if( x == 20 ){
            System.out.print("Value of X is 20");
        }else if( x == 30 ){
            System.out.print("Value of X is 30");
        }else{
            System.out.print("This is else statement");
        }
    }
}

```

مخرجات المثال السابق سيظهر رسالة Value of X is 30 ذلك لان البرنامج سيقوم بالمرور على كافة جملة if...else للتأكد من أن المتغير X = 30 و قد تم التحقق منه في السطر الثالث

- **الصيغة الرابعة لجملة IF** و هي عبارة جمل متداخلة و تأخذ الشكل التالي :

```
if(Boolean_expression 1){  
    // تنفيذ شفرة برمجية  
    if(Boolean_expression 2){  
        // تنفيذ شفرة برمجية  
    }  
}
```

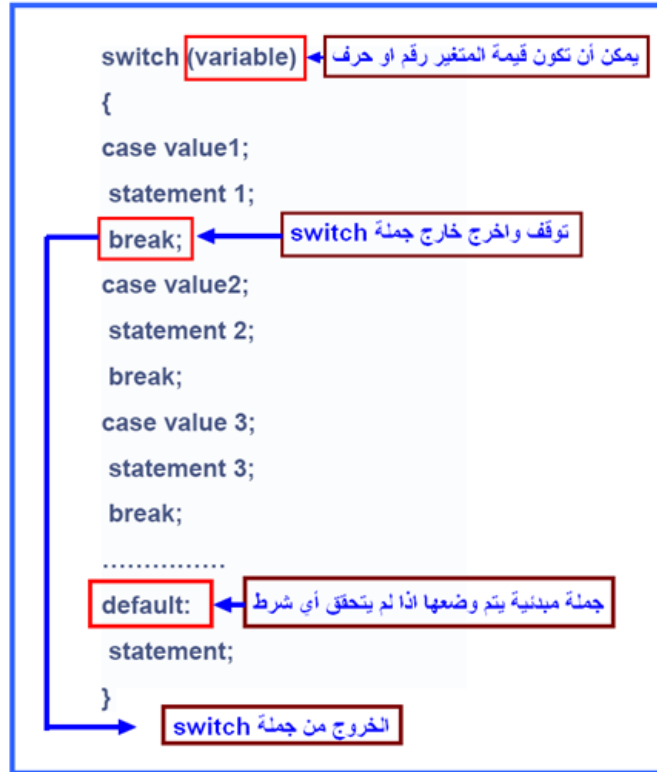
وكما ترى في الصيغة السابقة فهي عبارة عن جمل IF متداخل و تسمح لك بتنفيذ أكثر من جملة برمجية في وجود تحقق عدة شروط

مثال للتوضيح :

```
public class Test {  
    public static void main(String args[]){  
        int x = 30;  
        int y = 10;  
        if( x == 30 ){  
            if( y == 10 ){  
                System.out.print("X = 30 and Y = 10");  
            }  
        }  
    }  
}
```

2_جملة switch

تستخدم عبارة if إذا كان جواب الشرط عبارة عن احتمالين أو ثلاثة احتمالات على الأكثر، أما إذا زاد عدد الاحتمالات على ذلك فمن الأفضل استخدام عبارة switch وصيغتها العامة كالآتي:



ولكي نفهم كيفية عمل جملة switch نجري المثال التالي:

مثال للتوضيح : لنفرض أننا نريد إن يطبع الحاسب جملة ترحيب معينة إذا تم الضغط على أحد الأرقام في لوحة المفاتيح.
الحل:

```

1 package m;
2 import javax.swing.JOptionPane;
3 public class M {
4
5     public static void main(String[] args) {
6
7         String no;
8         int number;
9         no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10        number=Integer.parseInt(no);
11        switch (number) {
12            case 1:
13                JOptionPane.showMessageDialog(null, "السلام عليكم", "Egy_Tech", 2);
14                break;
15            case 2:
16                JOptionPane.showMessageDialog(null, "صباح الخير", "Egy_Tech", 2);
17                break;
18            case 3:
19                JOptionPane.showMessageDialog(null, "مساء الخير", "Egy_Tech", 2);
20                break;
21            default:
22                JOptionPane.showMessageDialog(null, "الرقم الذي أدخلته غير موجود لدى البرنامج", "Egy_Tech", 2);
23            }
24        }
25    }

```

وعند تنفيذ هذا البرنامج فإنه تحدث إحدى الحالات الآتية:



- ❖ إذا تم إدخال الرقم 1 فإنه يطبع العبارة الأولى "السلام عليكم" ثم يجد عبارة break فيخرج خارج جملة switch وينتهي البرنامج.
- ❖ إذا تم إدخال الرقم 2 فإنه يطبع العبارة الثانية "صباح الخير" ثم يجد عبارة التوقف break فيخرج خارج جملة switch وينتهي البرنامج.
- ❖ وهكذا في حالة إدخال الرقم (3) فإنه يطبع الجملة الثالثة ثم break ثم يخرج.
- ❖ أما في حالة إدخال أي رقم غير موجود في البرنامج وليكن (4) مثلاً، فإن البرنامج يطبع العبارة الموجودة في جملة "default عفووا الرقم غير موجود".... ثم ينتهي البرنامج.

ملحوظة:

يجب مراعاة أن جملة switch لها قوسي بداية ونهاية....و يجب دائماً أن تنتهي كل حالة case من حالات switch بالعبارة break حاول أن تقوم بإلغاء هذه العبارة وتجربة البرنامج ... ماذا تجد؟؟.جرب إدخال حرف بدلا من الرقم ماذا تجد؟؟

ثانيا : حلقات التكرار

تستخدم الحلقات التكرارية او جمل الدوران في كل لغات الحاسوب لتساهم في أداء العمليات التي تتكرر في البرنامج ... في عبارة عن بناء برمجي يحتوي علي جمل برمجية قابلة للتنفيذ مرارا تحت تأثير شرط معين ..بمعنى أنه طالما الشرط محقق يتم تنفيذ الكود البرمجي الذي بداخل تلك الحلقة وتحتوى لغة الجافا على ما يلي من الحلقات التكرارية:

- while
- do...while
- for

• الحلقة التكرارية while

و تستخدم تلك الحلقة في تكرار تنفيذ كود معين لعدد معين من المرات و يتم بنائها على النحو التالي :

```
while (expression)
{
    الكود البرمجي المراد تنفيذه //
}
```

ومن خلال الجملة السابقة فان الكود سيتم تنفيذه داخل الحلقة طالما أن الشرط expression صحيح

مثال للتوضيح :

```
public class Test {
    public static void main(String args[]) {
        int x = 0;
        while( x < 5 ) {
            System.out.print("ahmed");
            x++;
        }
    }
}
```

من خلال المثال السابق فان البرنامج سيقوم بطباعة الاسم ahmed ثم يقوم بإضافة القيمة 1 الى المتغير x حيث أن الجملة ++x تعنى زيادة عدد واحد للمتغيرثم يعاود الكرة حتى يصبح قيمة المتغير x أكبر من 5 حينها سيصبح الشرط غير صحيح $x < 5$ و بالتالي ستتوقف الحلقة التكرارية عن التكرار و النتيجة أنه سيتم طباعة الاسم ahmed خمس مرات

• الحلقة التكرارية do...while

وتلك الحلقة تشبه كثيرا الحلقة while و الفرق بينهما أن الاولى تقوم بالتحقق من قيمة الشرط قبل تنفيذ ما بداخلها من تعليمات في حين أن الثانية تقوم بتنفيذ التعليمات مرة واحدة أولا ثم تتحقق من الشرط و يتم بنائها على النحو التالي :

```
do
{
    التعليمات البرمجية //
}while (Boolean_expression);
```

مثال للتوضيح :

```
public class Test {  
    public static void main(String args[]){  
        int x = 0;  
        do{  
            System.out.print("ahmed");  
            x++;  
        }while( x < 5 );  
    }  
}
```

وهنا سيتم أيضا طباعة الاسم أحمد خمس مرات وهي نفس النتيجة التي توصلها إليها في الحلقة **while** وذلك لكون أن الشرط صحيح 5 أكبر من قيمة المتغير **x** لكن في حالة أن الشرط غير صحيح بمعنى أن 5 أقل من القيمة **x** فإن في تلك الحالة تختلف النتيجة ما بين الحلقة الكرارية الأولى **while** و الحلقة التكرارية الثانية **do...while** فالأولى لن يتم تنفيذ أي كود برمجى بداخلها فى حين فى الثانية سيتم طباعة الاسم **ahmed** مرة واحدة ثم ينعدم الشرط فتقف عن الدوران

- الحلقة التكرارية **for**

و تستخدم تلك الحلقة التكرارية فى حالة معرفة عدد مرات الدوران أو التكرار المطلوب تنفيذها و يتم بنائها على النحو التالى :

```
for(initialization; expression; update)  
{  
    //التعليمات البرمجية المراد تنفيذها  
}
```

وكما ترى أختي الكريم فإن جملة التكرار **for** تتطلب ثلاث معطيات على الأقل وهما :

Initialization المتغير الذى سنحقق به الشرط
Expression الشرط
Update معدل الزيادة و النقصان

مثال توضيحي :

```
public class Test {  
    public static void main(String args[]) {  
        for(int x = 0; x < 5; x = x+1) {  
            System.out.print("ahmed");  
            System.out.print("\n");  
        }  
    }  
}
```

• الحلقة التكرارية enhanced For

من إصدار الجافا رقم 5 دعمت نوع جديد من الحلقات التكرارية كوسيلة أبسط لتكرار عناصر تعبير (مجموعات البيانات) كالمصفوفات أو ArrayList..... الخ غير ذلك و يتم بنائها على النحو التالي :

```
for (مجموعة البيانات : متغير من نفس نوع مجموعة البيانات)
{
    //التعليمات البرمجية
}
```

مثال للتوضيح :

```
public class Test {
    public static void main(String args[]){
        int [] numbers = {10, 20, 30, 40, 50};
        for(int x : numbers ){
            System.out.print( x );
            System.out.print(",");
            System.out.print("\n");
        }
    }
}
```

مثال آخر :

```
public class Test {
    public static void main(String args[]){
        String [] names ={"James", "Larry", "Tom", "Lacy"};
        for( String name : names ) {
            System.out.print( name );
            System.out.print(",");
        }
    }
}
```

الفصل السادس : المصفوفات Arrays !!

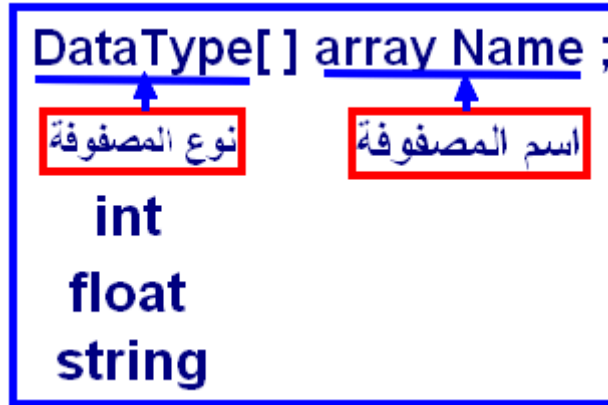
أولا : تعريف المصفوفة و انواعها:

المصفوفة هي عبارة عن مواقع يتم تخزين البيانات فيها لمدة مؤقتة طويلة فترة تنفيذ البرنامج فقط ..أو بمعنى عامى عبارة عن جدول يتم تخزين فيه البيانات و لفترة معينة.....يوجد نوعان من المصفوفات:

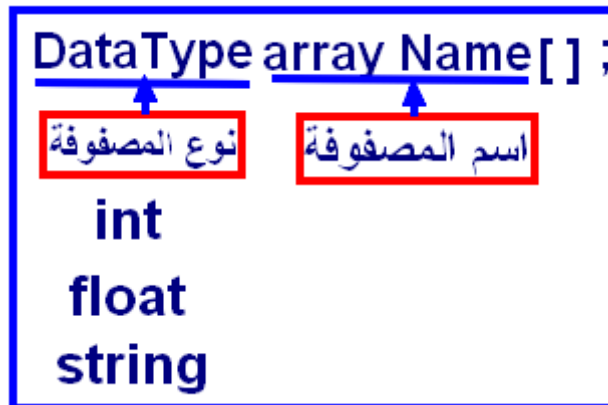
- المصفوفة الأحادية: وهي مكونة من بعد واحد فقط.
- المصفوفة متعددة الأبعاد: وهي مكونة من عدد من الصفوف والأعمدة (ليس شرطا ان تكون بعدين.)

ثانيا : كيفية انشاء المصفوفات فى لغة الجافا

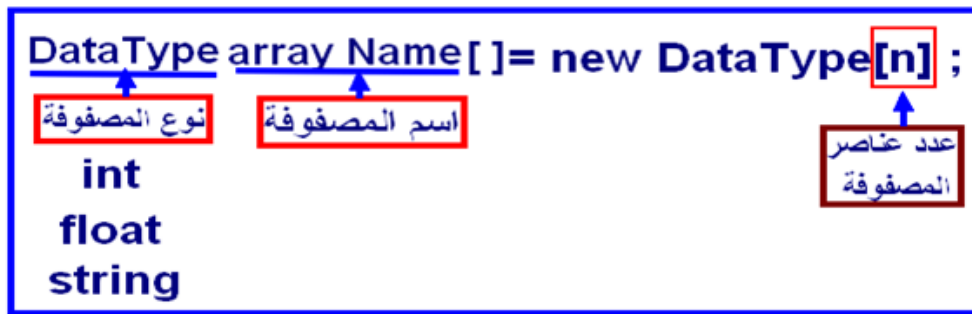
- المصفوفة ذات البعد الواحد....والصيغة العامة للإعلان عن المصفوفة ذات البعد الواحد كالتالي:



أو يمكن الإعلان عنها بنفس الصيغة السابقة مع وضع الأقواس بعد اسم المصفوفة كالتالي:



وبعد الإعلان عن المصفوفة لابد من تحديد عدد عناصرها ويتم ذلك كالتالي:



فمثلا لعمل مصفوفة رقمية من النوع int خاصة بدرجات عشرة طلاب مثلا يتم ذلك كالتالي:

```
int array1[];
array1[] = new int[10];
```

أو يمكن الإعلان عن المصفوفة وتحديد عدد عناصرها في سطر واحد كالتالي:

```
int array1[] = new int[10];
```

أو بهذا الشكل التالي:

```
int[] array1 = new int[10];
```

أذن كل الخيارات متاحة امامك في الاعلان و انشاء مصفوفة ذات بعد واحد

مثال للتوضيح : المطلوب عمل مصفوفة ذات بعد واحد تحتوي على درجات خمس طلاب وطباعة الناتج على الشاشة. يفترض أن درجات الطلاب (١٠، ٢٠، ٣٠، ٤٠، ٥٠).

الحل و كتابة البرنامج كالتالي :

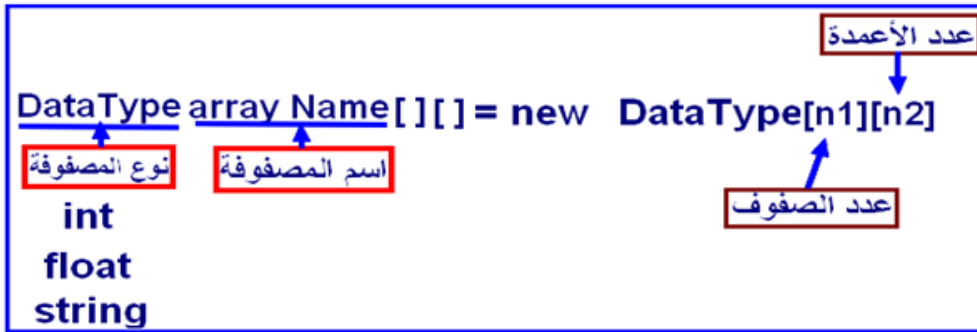
```
import java.util.Scanner;
public class array{

public static void main(String s[] ){
    int student[]={10,20,30,40,50};
    for(int i=0;i<=4;++i)
        System.out.println(student[i] );
    }
}
```

:

• المصفوفة متعددة الأبعاد (ذات البعدين) Multidimensional array

ويمكن القول بأن المصفوفة ذات البعدين هي عبارة عن جدول يحتوي على صفوف وأعمدة. والصيغة العامة لهذه المصفوفة كالتالي



ثالثا : كيفية إدخال العناصر الى المصفوفة من لوحة المفاتيح في البيئة غير الرسومية

في البيئة غير الرسومية :نقم بأستخدام دالة الإدخال.... Scanner

فمثلا لنفرض أن هناك مصفوفة ذات بعدين يراد فيها إدخال درجات ٦ طلاب عن طريق لوحة المفاتيح يتم ذلك كالتالي:

سنقوم بتسمية المصفوفة student وسوف نستخدم دالة الإدخال Scanner لإدخال قيمة صحيحة إلى هذه المصفوفة وهذا هو شكل البرنامج

```
import java.util.Scanner;
public class array{

    public static void main(String s[ ] ){

1  int student[ ][ ]=new int[3][2]; ← الأعلان عن مصفوف ذات بعدين
2  Scanner Keyboard=new Scanner(System.in);
3  for(int row=0;row<3;row++){ ← يتم استخدام حلقتين واحدة للصف
4  for(int column=0;column<2;column++) ← والأخرى للعمود
5  student[row][column]=Keyboard.nextInt(); ← يتم هنا ادخال عناصر المصفوفة
    }
    }
}
```

في السطر الأول يتم الإعلان عن مصفوفة ذات بعدين من النوع int وعدد عناصرها ٦ عناصر.

السطر الثاني سبق وتم شرحه في جملة الإدخال.

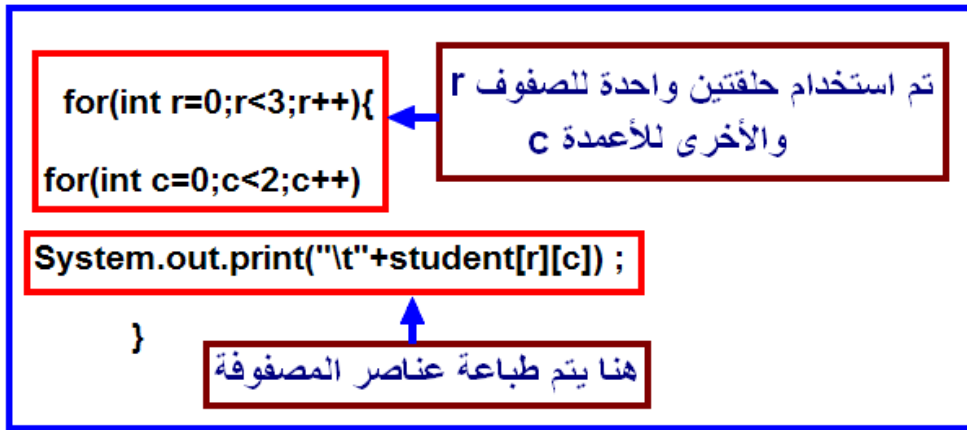
السطر الثالث والرابع تم عمل حلقتين ، الحلقة الأولى الخارجية للإشارة إلى رقم الصف row والحلقة الثانية للإشارة إلى رقم العمود .column. وطبعاً لاحظ ان الحلقة الخارجية قد تم إضافة أقواس لها .

أما السطر الخامس فيتم استخدام دالة الإدخال كما شرحناها سابقاً. ولكننا هنا استخدمنا المصفوفة student كمتغير يتم تخزين عناصر المصفوفة المدخلة فيه.

وعند تشغيل البرنامج لن تظهر لك أي رسالة . لأننا لم نضع رسالة للإدخال يمكنك أنت إضافتها ولكن، سيظهر المؤشر في أقصى يسار الشاشة منتظراً إدخال قيم عناصر المصفوفة فيتم إدخال ٦ عناصر والضغط على مفتاح الإدخال كل مرة. وبعد تمام الإدخال للستة عناصر تظهر علامة المحث للدرس دلالة على انتهاء الإدخال.

رابعا : كيفية قراءة عناصر المصفوفة

بعد أن يتم إدخال عناصر المصفوفة وإجراء أي عمليات عليها كالعلاقات الحسابية مثلاً، بهمنا أن نرى المخرجات على الشاشة. وفي هذا البرنامج سوف نتعرف على كيفية قراءة عناصر المصفوفة وطباعة هذه العناصر كما هي على الشاشة دون أي تغيير، ويتم ذلك عن طريق البرنامج التالي:



خامسا : إدخال العناصر الى المصفوفة في البيئة الرسومية

في بيئة النوافذ أو البيئة الرسومية نستخدم JOptionPane والتي من خلالها نقوم بإنشاء مربع حوارى للمستخدم يستطيع من خلاله ادخال البيانات

فلنتفرض مثلاً أننا نريد انشاء مصفوفة تتكون من خمس عناصر يتم تخزين فيها مجموع الدرجات للطلاب ثم يقوم بعض النتائج التي أكبر من ٦٠ درجةالحل سيكون الكود على النحو التالي :

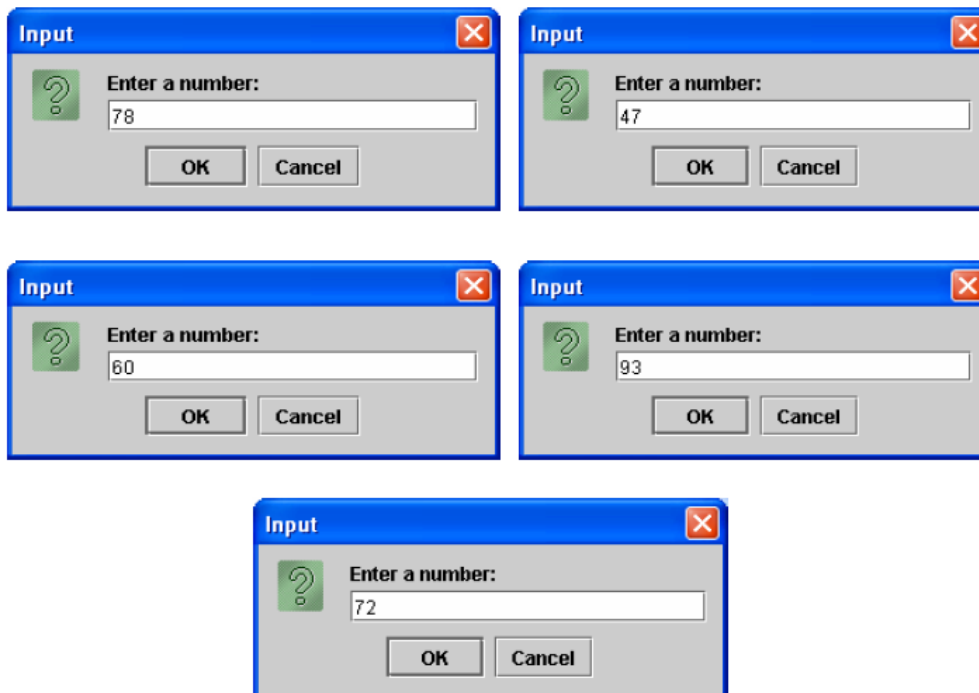
```

package array2;
import javax.swing.*;
public class Array2 {

    public static void main(String[] args) {
        int b[]=new int[5];
String s;
for(int i=0; i<5; i++){
s=JOptionPane.showInputDialog("Enter a number:");
b[i]=Integer.parseInt(s);
} // end for
for(int i=0; i<5; i++)
    if(b[i]>=60)
System.out.println(b[i]);
    }
}

```

شرح المثال : فى السطر رقم (٤) قمنا بتعريف مصفوفة أسمها b وتم حجز ٥ مواقع فى تلك المصفوفة... وفى الاسطر من (٦ الى ٩) قمنا بعمل حلقة تكرارية لإخراج المربع الحوارى input لادخال بيانات من المستخدم و ذلك لتخزينها فى مواقع المصفوفة... بعد ذلك يقوم البرنامج بطاعة الدرجات التى تساوى أو تزيد عن الرقم ٦٠ و يكون مخرجات البرنامج كما يلى :



مثال آخر:

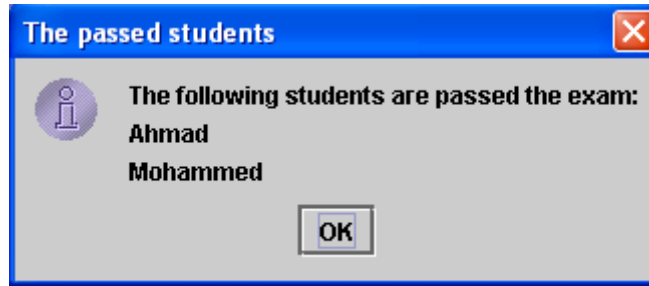
```
package array2;
import javax.swing.*;
public class Array2 {

    public static void main(String[] args) {
        double marks[] = new double[6];
        String names[] = new String[6];
        String s;

        for(int i=0; i<6; i++){
            s=JOptionPane.showInputDialog("أدخل اسم الطالب");
            names[i]=s;
            s=JOptionPane.showInputDialog("أدخل درجة الطالب");
            marks[i]=Double.parseDouble(s);
        }
        String title = "The passed students";
        String results="The following students are passed the exam:\n";
        for(int i=0; i<6; i++){
            if(marks[i]>=60)
                results=results+names[i)+"\n";
        }
        JOptionPane.showMessageDialog(null, results, title,
        JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}
```

الشرح : فى السطر رقم (٧) قمنا بإنشاء مصفوفة اسمها marks من نوع double بها ٦ مواقع وذلك لتخزين درجات الطلاب فيها ... أما فى السطر رقم (10) انشأنا مصفوفة أخرى اسمها names من نوع String لتخزين اسماء الطلاب ٦ بها ... من بداية السطر رقم (١١) استخدمنا الدوران لادخال اسماء و درجات الطلاب الستة و هذا طبعا عن طريق الامر showInputDialog الخاص بالكلاس و من خلال الدوران ايضا يتم اضافة الطلاب التى تزيد أو تساوى درجاتهم الرقم (٦٠) الى المخرجات و طباعة المخرجات عن طريق الامر showMessageDialog الخاص ايضا بنفس الكلاس JOptionPane





خامسا : ترتيب عناصر المصفوفة :

في كثيرا من التطبيقات نحتاج الى ترتيب المصفوفات أما ان يكون ترتيبا تصاعديا من الاكبر الى الاصغر أو ترتيبا تنازليا من الاصغر الى الاكبر.... وتوجد عدة طرق لذلك:

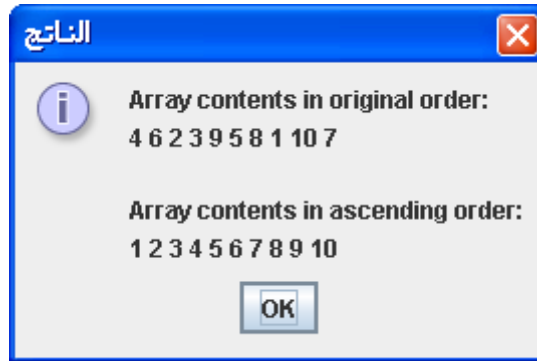
الترتيب المصفوفة باستخدام خوارزمية الترتيب الفقاعي:

وتتم عملية الترتيب الفقاعي للمصفوفة بمقارنة كل عنصر مع العنصر التالي له، فإذا كان العنصر الأول أكبر من العنصر التالي تتم عملية تبديل العنصرين باستخدام متحول وسيط، وفي كل مرحلة نحصل على العنصر الأكبر في نهاية المجموعة المقارنة، وتكرر العملية هذه $(n-1)$ مرة باعتبار أن (n) هو عدد عناصر المصفوفة. والبرنامج التالي يبين ذلك :

```
1
2 package array2;
3 import javax.swing.*;
4
5 public class Array2 {
6
7     public static void main(String[] args) {
8
9         int num[] = new int[] {4, 6, 2, 3, 9, 5, 8, 1, 10, 7};
10        int temp;
11        String output="Array contents in original order:\n";
12        for(int i=0; i<num.length; i++)
13            output+=num[i]+" ";
14        for(int i=1; i<num.length; i++)
15            for(int j=0; j<num.length-1; j++)
16                if(num[j]>num[j+1]){
17                    temp=num[j];
18                    num[j]=num[j+1];
19                    num[j+1]=temp;
20                }
21        output+="\n\nArray contents in ascending order:\n";
22        for(int i=0; i<num.length; i++)
23            output+=num[i]+" ";
24        JOptionPane.showMessageDialog(null, output, "النتائج",
25        JOptionPane.INFORMATION_MESSAGE);
26        System.exit(0);
27    }
28 }
```

في الاسطر من ١٢ الى ٢٠ يوجد عمليتان دوران متداخلتان يتم من خلالها ترتيب عناصر المصفوفة num

حيث يتم فحص القيم في السطر رقم ١٦ و اذا تحقق هذا الشرط يتم تبديل عنصرين من عناصر المصفوفة بحيث يأخذ كل واحد من العنصرين مكان الاخر داخل المصفوفة...وتكون مخرجات البرنامج كما يلي :



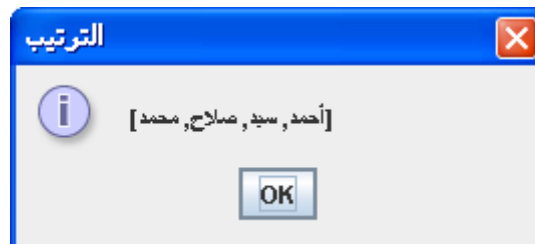
ترتيب المصفوفة باستخدام الكلاس. `util`.

الامر سهل جدا في ترتيب عناصر مصفوفة ما باستخدام الكلاس المسمى `util` كل ما عليك فعله هو استيراد هذا الكلاس الى مشروعك و البرنامج التالي يبين ذلك :

```
1
2 package array2;
3 import javax.swing.*;
4 import java.util.Arrays;
5 public class Array2 {
6
7     public static void main(String[] args) {
8
9         String names[] = {"صلاح", "سيد", "محمد", "أحمد"};
10        Arrays.sort(names);
11        JOptionPane.showMessageDialog(null, Arrays.toString(names), "الترتيب",
12        JOptionPane.INFORMATION_MESSAGE);
13        System.exit(0);
14    }
15 }
```

الشرح:

في السطر رقم (٤) قمنا باستيراد الكلاس... `util.Arrays` أما في السطر رقم ٩ قمنا بإنشاء المصفوفة `names` و اعطاء القيم لها...أما رقم رقم ١٠ قمنا بترتيب عناصر المصفوفة باستخدام الامر `Arrays.sort` و وضعنا اسم المصفوفة المراد ترتيب عناصرها بين قوسين...أما مخرجات البرنامج عن طريق المربع الحوارى `showMessageDialog` وهو يحتوى على عناصر المصفوفة مترتبة كما في الشكل التالي :



واخيرا تبقى لنا في هذا الدرس كيفية البحث الخطى و البحث الثنائى فى المصفوفات فالى اللقاء

الفصل السابع : جمل معالجة الاستثناءات

من الثابت بقواعد الرياضيات انه لا يجوز القسمة على صفر بأي حال من الأحوال لذلك لم يتمكن البرنامج من تنفيذ العملية المطلوبة وستظهر للمستخدم رسالة خطأ تنفيذي لا يمكنه تفاديها ولعل الأغلب أنها ستسبب بإغلاق البرنامج.

ولمعالجة أخطاء التنفيذ هذه وتحجيمها قبل حدوثها وفرت لنا لغات البرمجة وعلى رأسها الجافا آلية تتمثل بوضع جمل استثناءات خاصة يتوقع من خلالها المبرمج أنواع أخطاء التنفيذ التي قد تحدث ويقوم ببرمجة حدث برمجي بكل استثناء ليتم تنفيذه بدلاً من خطأ التنفيذ. الخالصة إن الاستثناءات هي عبارة عن آلية برمجية لمعالجة أخطاء التنفيذ المحتمل وقوعها.... وهذا ما يقصد بمعالجة الاستثناءات في لغة الجافا.

أقسام جمل معالجة الاستثناءات

تقسم جملة معالجة الاستثناءات إلى ثلاثة أقسام رئيسية، هي:

الاول القسم try

وهي عبارة عن كتلة برمجية خاصة يكتب بداخله المبرمج التعليمات البرمجية التي يتوقع حدوث خطأ تنفيذي خلالها وتكون صيغتها العامة على النحو الآتي:

الرمز:

```
try
{
    java code statements;
}
```

الثاني القسم catch

ويختص هذا القسم باحتواء التعليمات البرمجية التي من شأنها معالجة الأخطاء التي قد تحدث داخل القسم try وبما ان أخطاء التنفيذ كثيرة متعددة فيمكننا صياغة أكثر من تعليمة catch لتحتوي كل واحدة منها على نوع معين من أخطاء التنفيذ وتكون عادة جميع جمل catch مرتبطة بجملة try تكتب جمل catch وفق الصيغة العامة التالية:

الرمز:

```
catch (ExceptionType name)
{
    java code statments;
}
```

يجب تضمين جملة catch نوع الخطأ المتوقع حدوثه Exception Type ومنح الخطأ اسماً حتى تتمكن من التعامل معه. ويمكننا داخل كتلة catch صياغة تعليمات برمجية ليتم تنفيذها عند حدوث الخطأ لتضمن عند تنفيذ البرنامج تحجيم نسبة أخطاء التنفيذ وإعادة توجيه سير تدفق البرنامج إلى التعليمات المدرجة داخل catch دون ظهور رسالة خطأ التنفيذ أو ان يؤدي ذلك إلى إغلاق البرنامج.

يعتبر القسم catch إجباري فلا يمكننا استخدام معالجة الاستثناءات بدونها ولكن يمكننا إدراجه بصورته العامة دون صياغة أي تعليمة برمجية بداخله.. وهنا بمجرد حدوث الخطأ التنفيذي المحدد داخل جملة catch الفارغة فان البرنامج سيتجاهل تنفيذ التعليمة البرمجية التي تحتوي الخطأ والمدرجة داخل القسم try ليتنقل مباشرة إلى تنفيذ التعليمات التي تلي معالجة الاستثناء كما سنبين ذلك في الأمثلة القادمة.

الثالث القسم **finally** علما بان الصيغة العامة للقسم **finally** تكون على النحو التالي.

الرمز:

```
finally
{
    java code statements;
}
```

بالرغم من إن القسم **finally** يعتبر اختبارا طالما أن القسم **catch** موجود إلا انه يعتبر أهم أقسام كتلة معالجة الاستثناء فمن خلاله يمكن للمبرمج تحديد أي من جمل الاستثناء **catch** التي سيتم تنفيذها إذا كان هناك أكثر من جملة. كما يمكن للمبرمج صياغة تعليمات برمجية داخل هذا القسم ليتم تنفيذها بصورة مستقلة عن التعليمات المدرجة داخل جمل **catch** كما يمكننا الاعتماد على القسم **finally** كبديل للقسم **catch** خاصة إذا كنا لا نريد تحديد نوع الخطأ المتوقع.

الصيغة الكاملة لكتلة معالجة الاستثناء

الشكل التالي يبين الأقسام الثلاث لكتلة معالجة الاستثناء

الرمز:

```
try
{
    java code statments;
}
catch (ExceptionType1 name)
{
    Exception 1;
}
catch (ExceptionType2 name)
{
    Exception 2;
}
finally
{
    java code statements;
}
```

آلية تنفيذ كتلة الاستثناءات

لمعرفة طريقة تنفيذ الكتل الاستثنائية، ادرس البرنامج التالي.

الرمز:

```
public class MyMain2
{
    public static void main(Stringargs[])
    {
```

```

system.Out.Println("Arabic SEO");
int X=10;
float b=X/0;
System.Out.Println("The answer is"+b);
}}

```

عند ترجمة البرنامج السابق سنحصل على نتيجة ايجابية مفادها خلو البرنامج من الأخطاء القواعدية والتركيبية. ولكن النتيجة بعد تنفيذ البرنامج حتما ستكون مغايرة إذ سيظهر لنا معالج الأخطاء التنفيذية رسالة Run Time Error لاحظ ان البرنامج قام بتنفيذ تعليمة الطباعة Arabic SEO وبمجرد انتقال تدفق البرنامج إلى تعليمة القسمة float b= X/0 وجد معالج أخطاء التنفيذ أن عملية القسمة غير الشرعية فلا يجوز القسمة على صفر ولهذا السبب اظهر لنا رسالة خطأ التنفيذ وبصورة متزامنة أوقف تنفيذ البرنامج ومنع تعليمة الطباعة الثانية من التنفيذ. لنعمل الآن على إعادة صياغة البرنامج باستخدام القسم try من معالجة الاستثناءات، كما في الشكل التالي:

الرمز:

```

public class MyMain2
{
    public static void main(String args[])
    {
        System.Out.Println("Arabic SEO");
        try
        {
            int X=10;

            float b=X/0;
            System.Out.Println("The answer is"+b);
        }
    }
}

```

لاحظ عند ترجمة البرنامج سنحصل على خطأ قواعدي ينص على إننا استخدمنا تعليمة الاستثناء try دون احد الأقسام الأساسية لها مثل catch أو finally لذلك علينا الالتزام بالتركيب العام لجمل الاستثناء وإعادة صياغة البرنامج بتضمينها القسم catch لتصبح على النحو الآتي:

الرمز:

```

public class MyMain2
{
    public static void main (String args[])
    {
        System.Out.Println("Arabic SEO");
        try
        {
            int X=10;

            float b=X/0;
            System.Out.Println("The answer is"+b);
        }
    }
}

```



```

catch (Exception e)
{
    System.Out.Println("Not Allow to Dived by 0");
}
}
}

```

استخدمنا القسم catch لمعالجة خطأ القسمة على صفر، لاحظ أننا صرحنا عن نوع الخطأ التنفيذي بصورة عامة Exception دون أي تحديد. وقمنا بصياغة تعليمة طباعة لتظهر للمستخدم ولتوضح له إن القسمة على صفر غير مسموحة لها. عند تنفيذ البرنامج والوصول إلى تعليمة القسمة سيكتشف معالج التنفيذ وجود الخطأ وبالتالي سينتقل مباشرة إلى القسم catch لينفذ ما بداخله من تعليمات. وبالتالي يمكننا إعادة صيغة البرنامج السابق بتوظيف القسم finally لطباعة النص Sorry أو أي كود برمجي آخر، كما في النص التالي.

الرمز:

```

public class MyMain2
{
    public static void main(String args[])
    {
        System.Out.Println("Arabic SEO");
    try
    {
        int X=10;
        float b=X/0;
        System.Out.Println("The answer is"+b);
    }
    catch(Exception e)
    {
        System.Out.Println(" ");
    }
    finally
    {
        System.Out.Println("Sorry");
    }
}
}
}

```

الفصل الثامن : التعامل مع الفئات Classes و الكائنات Objects و الطرق Methods !

في هذا الدرس نتناول حجر الزاوية في لغة الجافا ألا وهو الفئة class حيث تعتبر الفئات الوحدات الأساسية في لغة البرمجة الكائنية نظرا لما تحتويه من كائنات وطرق وبيانات ويعتبر هذا الدرس من أهم الدروس في هذه الدورة حيث أن البرمجة الكائنية التوجه كان من المفروض أن تسمى البرمجة القائمة على الفئات ولكن ليس هذا وقت التعديل في مسميات أصبحت مشهورة في العالم بهذا المسمى الذي يبخس حق الفئات وينصف أحد مكوناتها وهي الكائنات ولكن ما علينا - هيا بنا نبدأ -

أولا ...تعريف الفئة: Classes

الفئة هي القالب البرمجي للغة الجافا الذي يحدد شكل الكائنات لما تحتويه من بيانات ومعاملات والكائن ينتمي للفئة وليس هو كل الفئة وبالتالي للفئة عناصر منها المتغيرات والنهج (الدوال - الطرق) إذا الفئة هي المكان الذي يستطيع المبرمجون أن يجمعوا فيه بين البيانات والطرق (الدوال) تحت سقف واحد

ثانيا ...إنشاء الفئات:

يتم إنشاء الفئة باستخدام الكلمة المفتاحية [class] وهي من الكلمات المحجوزة في الجافا ويكون الشكل العام للفئة كما يلي:

```
class Nameclass  
{  
  
}
```

هذا اذا كان برنامجنا يحتوى على عدة فئات مجتمعة في ملف واحد و بالتالي فإن الفئة الرئيسية التي تحتوى على الدالة main تبدأ بمحدد الوصول public class أى أن الفئة عامه ...وتبدأ جميع الفئات الاخرى بالكلمة المحجوزة class فقط كما أعلاه....

أما اذا كان برنامجنا يحتوى على عدة فئات كلا في ملف منفصل فلا بد من كل الفصائل تبدأ public class

ويجب أن تحتوى أى فئة على طريقة واحدة أو متغير واحد أو كلاهما وحين نريد أن نصمم فئة يجب أن نراعي نوع البيانات ومدى توافقها معا داخل الفئة بمعنى أن فئة مصممة لتخزين أسماء موظفين وأرقام هواتفهم وعناوينهم لن يكون من المناسب أن أضع في نفس الفئة بيانات عن حالة الطقس وحالة البحر ومعدل سقوط المطر ولذلك يجب أن نراعي الناحية المنطقية عند تصميم الفئة من حيث علاقة البيانات التي نريد التعامل معها في نفس الفئة

ثالثا ... الكائنات Objects

هي عناصر تمثيل استخدام الفئة في البرنامج وهي تأخذ نفس بيانات الفئة من دوال وطرق و متغيرات ... الخ و الطريقة الوحيدة لإنشاء الكائن تكون باستخدام الامر new باستخدام الشكل العام التالي

```
classname objectname = new classname ();
```

وهذا يعنى أسم الفئة أسم الكائن = الكلمة المحجوزة (new) أسم الكلاس أو الفئة:();

مثال توضيحي / قم بإنشاء برنامج لحساب العددين ٥ و ٨ باستخدام عدد ٢ كلاس او فئة ؟

الحل:

الرمز:

```
package ex1;

class Egy{
    int no1=5,no2=8,total;
}

public class Ex1 {

    public static void main(String[] args) {

        Egy cal= new Egy();

        cal.total=cal.no1+cal.no2;
        System.out.println("Value= "+cal.total);
    }
}
```

- في السطر الثاني... قمنا بإنشاء الفئة أو الكلاس Egy عن طريق الامر class Egy ثم فتحنا قوسى البداية و النهاية... ويجب ملاحظة أننا استخدمنا الكلمة المحجوزة class لإنشاء تلك الفئة بدون محدد الوصول... public وذلك لكون جميع فئات برنامجنا فى ملف واحد فقط.
 - في السطر الثالث.... نلاحظ أننا عرفنا المتغيرات الثلاث وهي [no1] لتخزين الرقم الاول و [no2] لتخزين الرقم الثانى و [total] لتخزين عملية الجمع....
- وهكذا نلاحظ اننا قمنا بإنشاء ايسط كلاس او فئة لا تحتوى إلا على الاعلان عن المتغيرات فقط بدون اى طرق او دوال.
- في السطر الرابع... أنشأنا الكلاس او الفئة الرئيسية Ex1 و التى تحتوى على الدالة الرئيسية main و يجب ملاحظة أن تلك الفئة أو الكلاس بدأت بمحدد الوصول public
 - في السطر الخامس... قمنا بإنشاء الكائن cal من الفئة Egy باستخدام الكلمة المحجوزة new
 - في السطور الاخرى.... بعد ان انشأنا الكائن cal نتسطيع بسهولة استخدام المتغيرات الموجودة فيه وهى طبعا امتداد للبيانات و المتغيران من الفئة المنشىء منه هذا الكائن

رابعا : الطرق (الدوال) Methods

كما نوهنا سابقا عند الحديث عن الفئات بأن المكونات الرئيسية للفئة هي المتغيرات الأولية والطرق أو الدوال فإننا نتناول بالتفصيل في هذه الجزئية الحديث عن الطرق....

1_تعريف الطريقة [السلوك]

هي ذلك الجزء من الفئة الذي يختص بالتعامل مع البيانات وتعرف داخل الفئة و بها يتم التفاعل بين أجزاء البرنامج المختلفة والفئة و تحتوي الطريقة على جملة برمجية أو أكثر والكود البرمجي الجيد في لغة الجافا هو الذي يكتب بحيث تؤدي الطريقة مهمة واحدة ولكل طريقة اسم يتم استدعاء الطريقة به ويمكنك أن تطلق على طريقته الاسم الذي تفضله مع التنبيه على عدم استخدام الكلمات المحجوزة في لغة الجافا لهذه التسمية...

٢_ الشكل العام للطريقة

Access specifier return type method name(parameter type parameter name)

وقبل أن نناقش الشكل العام للطريقة بالتفصيل نذكر نوعا الطرق وهما

النوع الأول : هو النوع الذي يعود بقيمة ويستوجب استخدام return
النوع الثاني : هو النوع الذي لا يعود بقيمة void

والآن مع مناقشة الشكل العام المختصر للطريقة:

- **المقطع الاول Access specifier** محدد الوصول للطريقة أى نوع الوصول للطريقة فمثلا النوع العام الذي يتيح الوصول للطريقة من خارج الفئة ومن خارج البرنامج ويأخذ الكلمة المفتاحية **public** أما النوع الآخر وهو عكس النوع العام ويأخذ الكلمة المفتاحية **private** وهو يسمح بالوصول للطريقة من داخل الفئة فقط أما النوع الثالث فهو النوع المحمي ويأخذ الكلمة المفتاحية **protected**
- **المقطع الثاني return type** يحدد نوع القيمة المرتجعة من الدالة .
- **المقطع الثالث method name** وهو الذي يعطي للطريقة اسمها ومن الافضل استخدام أسم يدل على وظيفية الطريقة ودائما في لغة الجافا يبدأ الاسم بحرف صغير ...وفي حالة الاسم الذي يتكون من اكثر من مقطع يبدأ المقطع الاول بحرف صغير و المقطع الثاني بحرف كبير... علما بأن هناك طرقا تعرف بواسطة المبرمج وأخرى تكون جاهزة توفرها لغة الجافا وما على المبرمج سوى استدعاتها مثل الطرق الرياضية وحين تعرف طريقة يجب عليك إتباع الشروط الخاصة بالتعريف في لغة الجافا
- **المقطع الرابع -parameter** يوضع بين القوسين بيان بالعوامل الداخلة للطريقة ويفصل بين العامل والآخر فاصلة وفي حالة عدم وجود عوامل لا يوضع شيء بين القوسين
- المقطع الخامس جسم الطريقة. { أى الاكواد المراد تنفيذها من خلال تلك الطريقة.

مثلا توضيحي : قم بإنشاء برنامج لحساب العديدين ٥ و ٨ بأستخدام عدد ٢ كلاس او فئة مع إنشاء طريقة لحساب الناتج... الحل:

الرمز:

```
package ex1;

class Egy{
    int no1=5,no2=8,sun;

    void total(){
        sun = no1+no2;
        System.out.println("Value= "+sun);
    }
}
```

```
public class Ex1 {
    public static void main(String[] args) {
        Egy cal=new Egy();

        cal.total();

    }
}
```

دعنا الآن نلقي نظرة سريعة على هذا البرنامج ولنبدأ ببني القصيد ألا وهو إضافة الطريقة ومع السطر الأول

```
void total ( ) {
```

في هذا السطر نعلن عن طريقة بمسمى total بدون عوامل ولا ترجع قيمة void ولذلك لا ترجع قيمة لمن استدعاها ونهاية السطر توجد حاصرة للإذن ببدء جسم الطريقة الذي يتكون من سطرين هما

```
sun = no1+no2;
System.out.println("Value= "+sun)
```

ويتم تشغيل الطريقة في الدالة الرئيسية للبرنامج بعد إنشاء كائن من نفس الفئة عن طريق الامر التالي:

```
cal.total();
```

مثال توضيحي آخر.. قم بحل نفس السؤال السابق ذكرة مع استخدام العوامل في الطريقة total

```
class Egy{
    void total(int no1,int no2){
        int sun = no1+no2;
        System.out.println("Value= "+sun);
    }
}
```

```
public class Ex1 {
    public static void main(String[] args) {

        Egy cal=new Egy();

        cal.total(8,5);

    }
}
```

3_العائد من الطريقة Returning from aMethod

كما ذكرنا سابقا يوجد نوعان من الطرق هما:

النوع الأول : هو النوع الذي لا يعود بقيمة void method مثل الحالة في المثال السابق أعلاه

النوع الثاني : هو النوع الذي يعود بقيمة الطرق التي تعود بقيمة و هي الأكثر استخداما في مختلف مجالات البرمجة ويستوجب استخدام return و الشكل العام لها يكون على النحو التالي:

الرمز:

```
public int Meth03( ){  
  
    //Body of Method03  
    // طريقة لا تحتوي على عوامل وتعود بقيمة  
    return 0;  
}
```

مثال توضيحي: قم بحل نفس السؤال السابق ذكره مع جعل الطريقة total ترجع بقيمة العملية الحسابية

```
class Egy{  
    int no1=5,no2=8;  
  
    int total(){  
  
        return no1+no2;  
    }  
}  
  
public class Ex1 {  
  
    public static void main(String[] args) {  
  
        Egy cal=new Egy();  
  
        System.out.println("Value= "+cal.total());  
  
    }  
}
```

نلاحظ فيه أننا قمنا بإنشاء الطريقة (total) الذي سوف يستقبل القيمة الراجعة التي سوف تكون هنا من النوع [int] وذلك يستدعي أن تعود بقيمة من النوع الصحيح لمن استدعاها وعلى سبيل المثال إذا أردت طريقة تعود بقيمة من النوع [double] يجب عليك اختيار هذا النوع كما يلاحظ أننا استخدمنا الامر (return) لارجاع قيمة المعادلة (no1+no2)...



4_ المنشآت - البيانات Constructors

في المثال السابق تم كتابة المتغيرات بطريقة يدوية....وكانت هذه الطريقة للتدريب ولكن لمحترفي البرمجة يوجد شيء آخر لانجاز هذا الأمر ألا وهو استخدام البيانات Constructors ولكي نوضح هذا المفهوم إليك هذا المثال

```
class Egy{
    int no1,no2;
    Egy(){
        no1=0;
        no2=0;
    }
    int total(){

        return no1+no2;
    }
}
```

```
public class Ex1 {

    public static void main(String[] args) {

        Egy cal=new Egy();

        cal.no1=5;
        cal.no2=8;
        System.out.println("Value= "+cal.total());

    }
}
```

مما سبق نستنتج أن البيانات تشبه الطريقة من حيث القواعد وتأخذ نفس اسم الفئة وفيها الحل لمشكلة تحديد القيم الابتدائية وحيث أن اسم الباني هو نفسه اسم الفئة فإن الباني سوف يستدعى تلقائياً.....

في المثال أعلاه تم إنشاء فئة بسيطة class Egy وكان الباني فيها هو Egy وكما نلاحظ فإن اسم الفئة يشبه تماماً اسم الباني وعندما نقوم بإنشاء كائن cal حسب الجملة البرمجية (التعليمة)

```
Egy cal=new Egy();
```

يتم حجز مكان لهذا الكائن ثم يطلب الباني وسيتم تحديد قيمة ابتدائية له ونلاحظ هنا أن الباني قد خصص القيمة (0) لكل من المتغير no1,no2 وأن هذا الباني قد تم استدعاؤه بواسطة new عند إنشاء الكائن Egy

وبالتالي نستنتج فوائد المنشآت (البيانات) التي تعتبر (طريقة) غي عادية حيث لا تمتلك قيمة راجعة وتخلصنا من المشاكل البرمجية الخاصة بالفئات التي تنجم عن الكتابة بشكل يدوي كما في المثال قبل التعديل وأيضا تسهل قراءة وفهم مصدر البرنامج البرمجيو الخلاصة أن:

البناء - نهج (طريقة) خاص يتم تنفيذه عندما يبدأ كائن الفئة ويحتوي من الوسطاء ما يحدد شكل الكائن ويتم استدعاؤه تلقائياً لأنه يحمل نفس اسم الفئة والبيانات لا تمتلك قيمة راجعة

الباب الثانى : واجهة المستخدم الرسومية

الفصل الاول : الاطارات أو النوافذ

درس اليوم أن شاء الله تعالى سيكون عن واجهة المستخدم الرسومية والتي تتمثل فى النوافذ و الأزرار و مربعات النصوص و القوائم وغيرها..... وسنخصص هذا الدرس عن أهم العناصر الرسومية التي تستخدم فى برمجة التطبيقات بلغة الجافا وهى الاطارات أو النوافذ

أولا أنواع الاطارات و النوافذ فى الجافا

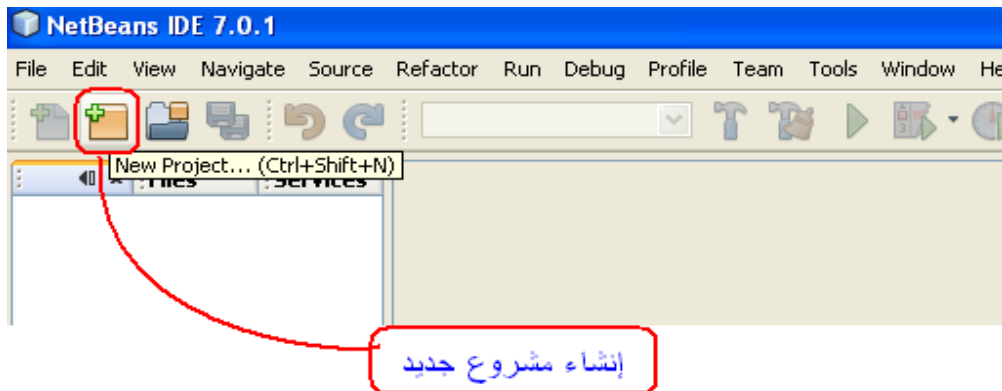
يوجد نوعين من الاطارات أو النوافذ فى لغة الجافا

- النوع الاول `awt` : وهو ينتمى للحزمة `awt`
- النوع الثانى `swing` وهو ينتمى للحزمة `swing`

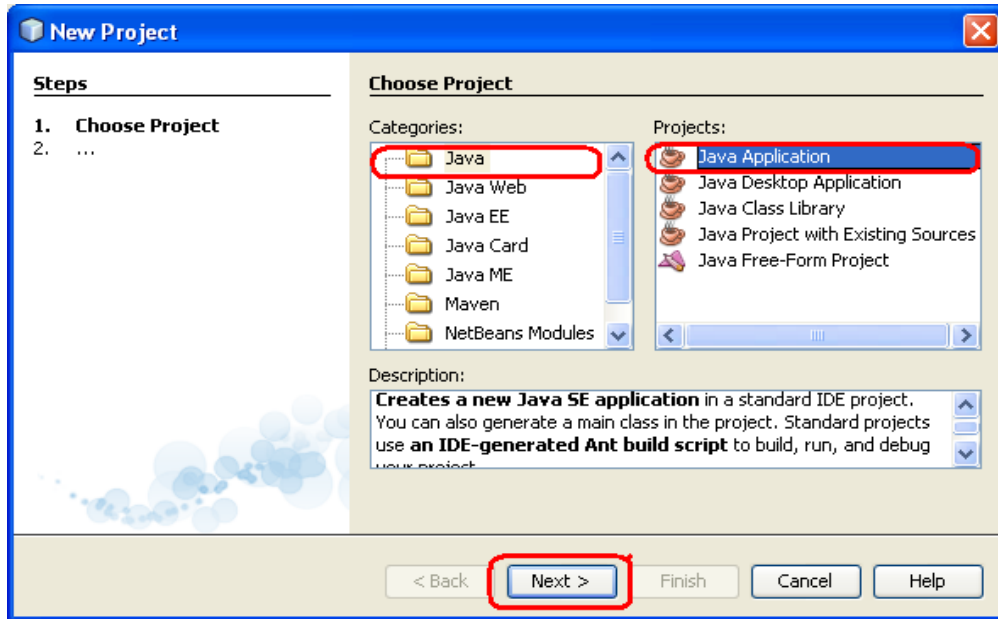
والفرق بينهما أن `awt` يأخذ شكل نظام التشغيل الذى يعمل عليه فى حين `swing` لا يتغير شكله بتغير نظام التشغيل

ثانياً_ كيفية إنشاء النوافذ التابعة للحزمة `swing`

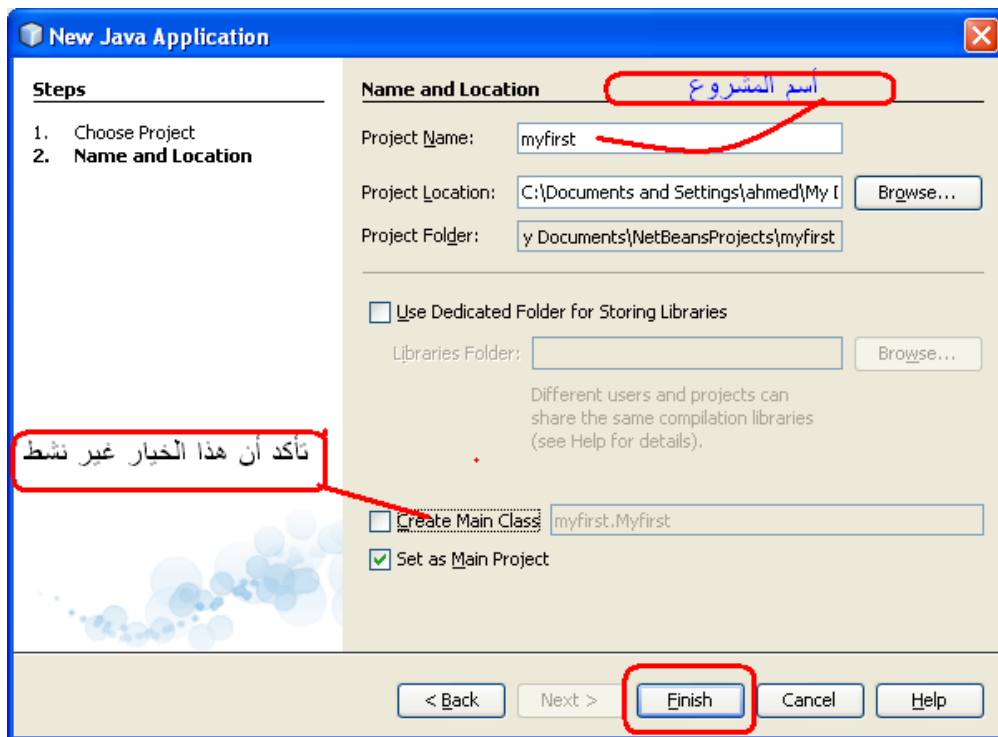
1_ قم بإنشاء مشروع جديد و ذلك من خلال الضغط على ايقونة مشروع جديد الموجودة فى شريط الادوات كما فى الصورة التالية:



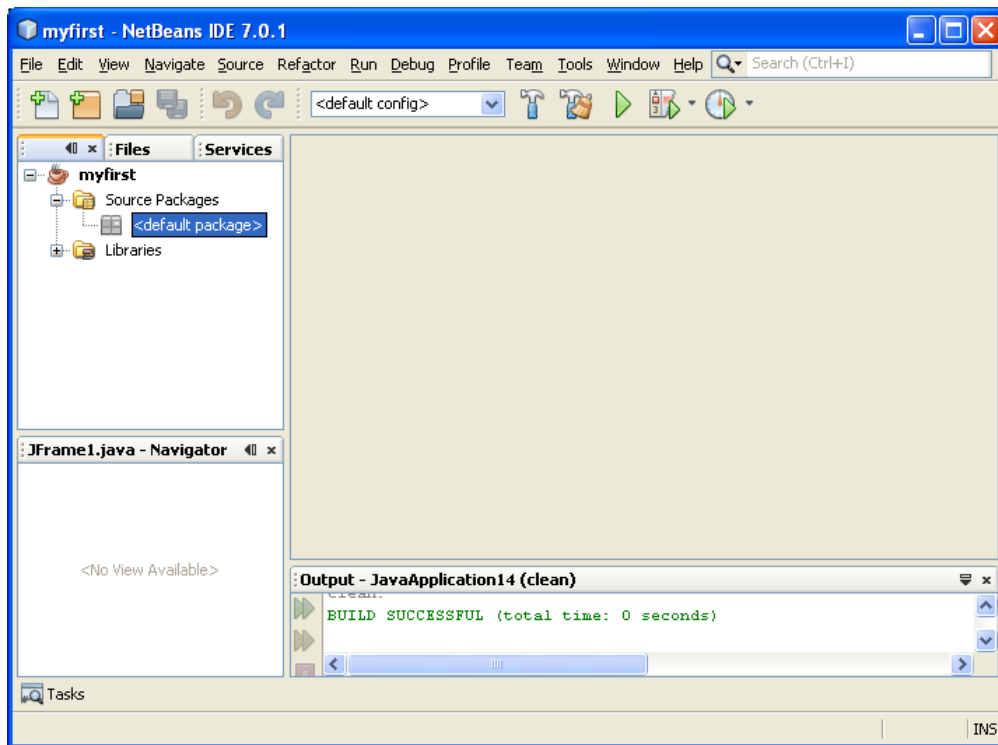
2_ سوف تظهر لك نافذة مشروع جديد فى جانب التصنيف أختار `java` ستجد أن القائمة مشاريع بها عدة مشاريع حدد نوع المشروع `java application` كما فى الصورة التالية ثم نضغط `Next`



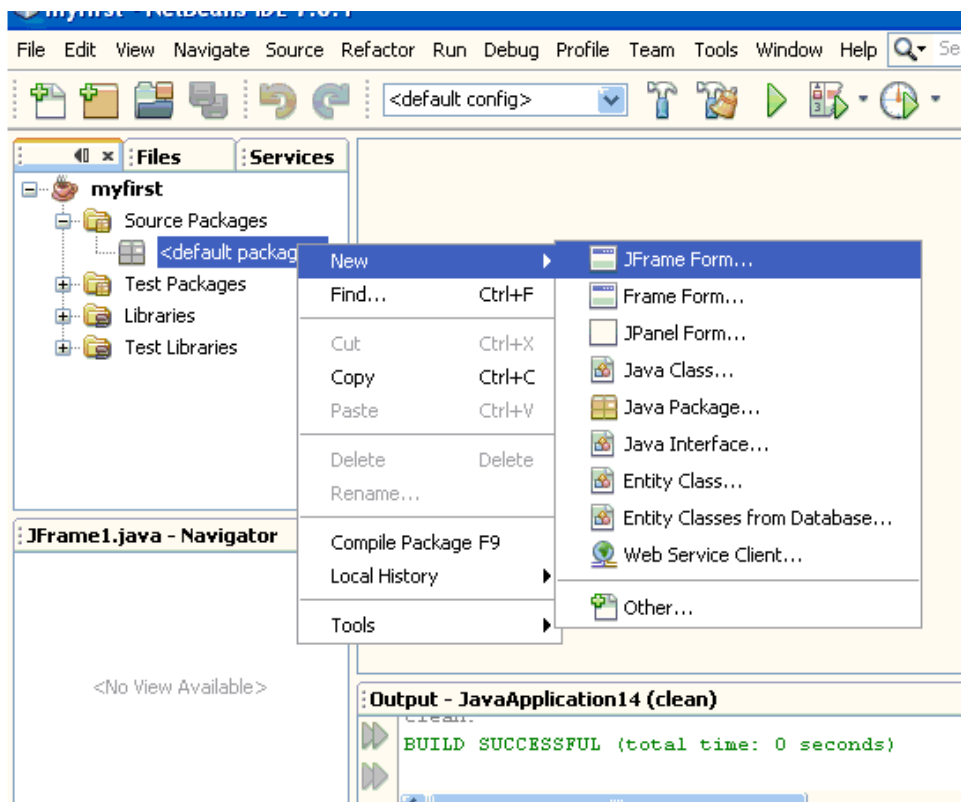
3_ في المربع الحوارى **New java application** قم بوضع اسم للمشروع وحدد المسار الذى تريد أن تحفظ فيه مشروعك... ثم تأكد أن الخيار **create main class** غير نشط ثم اضغط **finish**



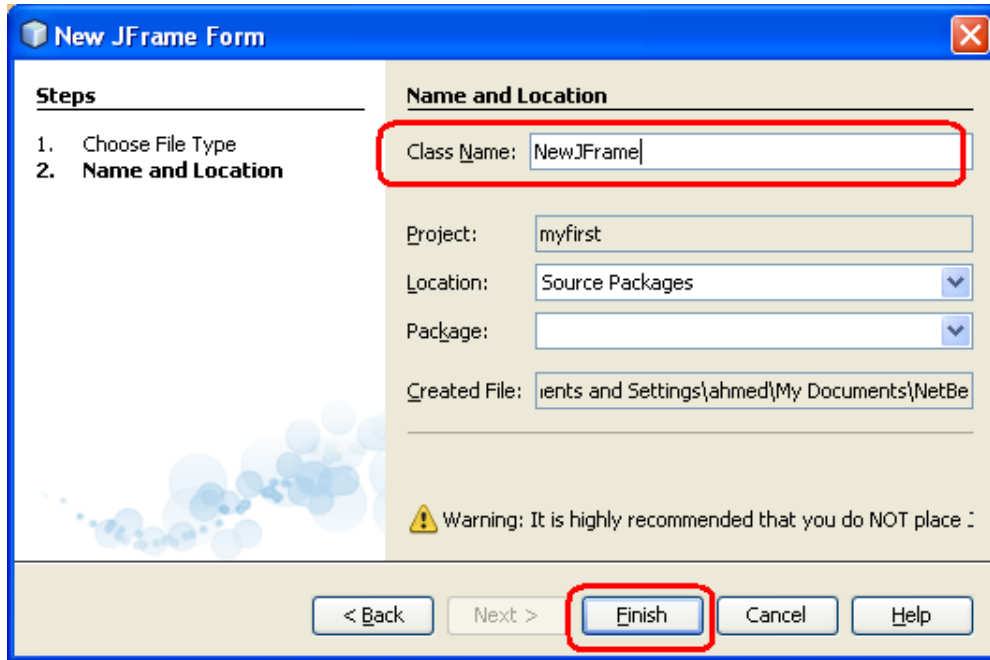
ستجد ان الصورة التالية:



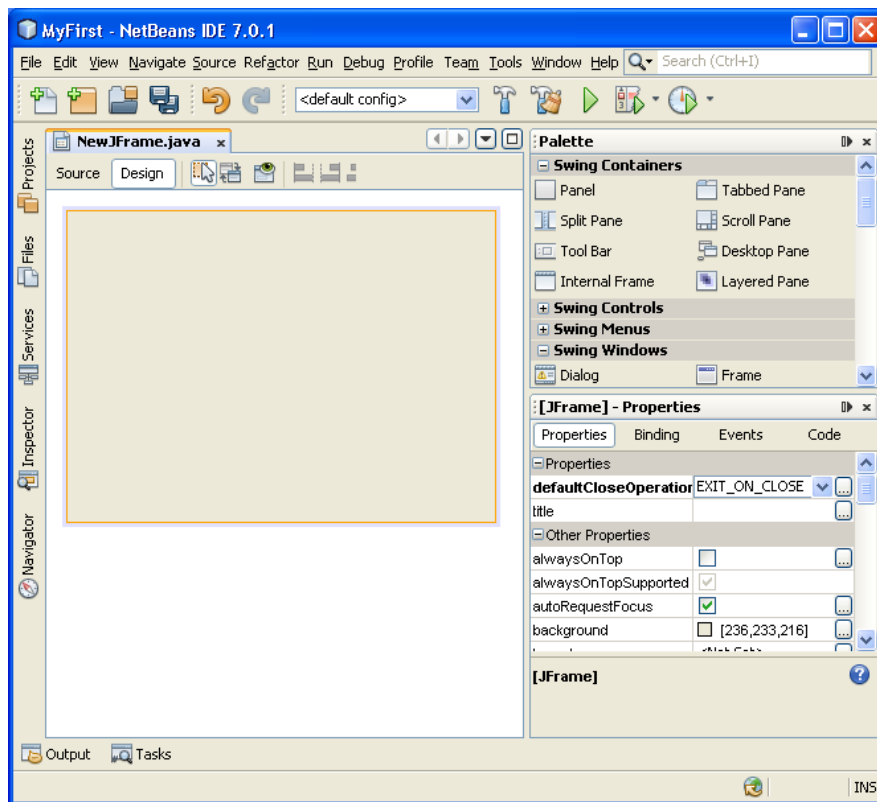
4_ قم بالضغط دبل كليك على ايقون **source packages** ثم قف بالماوس على ايقون **default packages** ثم كليك شمال بالماوس ثم **New** ثم **zfarme** كما فى الشكل التالى:



سوف يظهر لك المربع الحوارى التالى قم بإعطاء اسم للنافذة ثم اضغط إنهاء كما يلى:

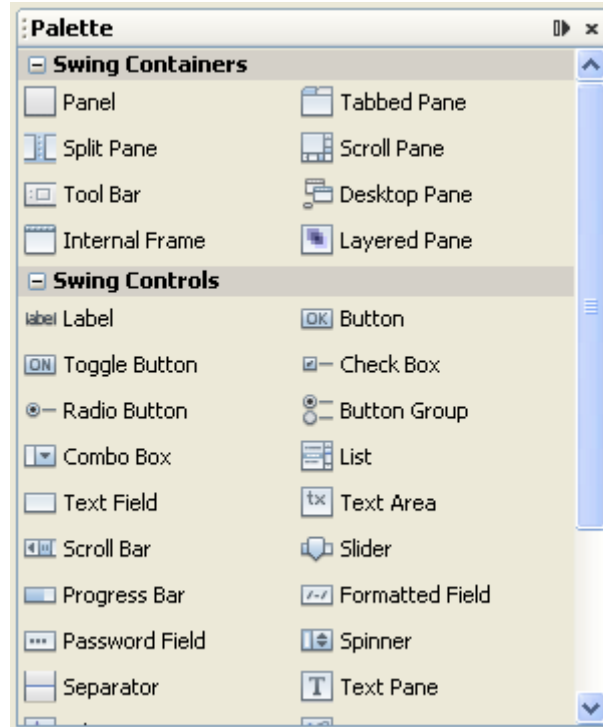


5_ بعد الضغط على إنهاء كما فى الشكل السابق نكون قد قمنا بإنشاء نافذة و ستجدها كما فى الشكل التالى:

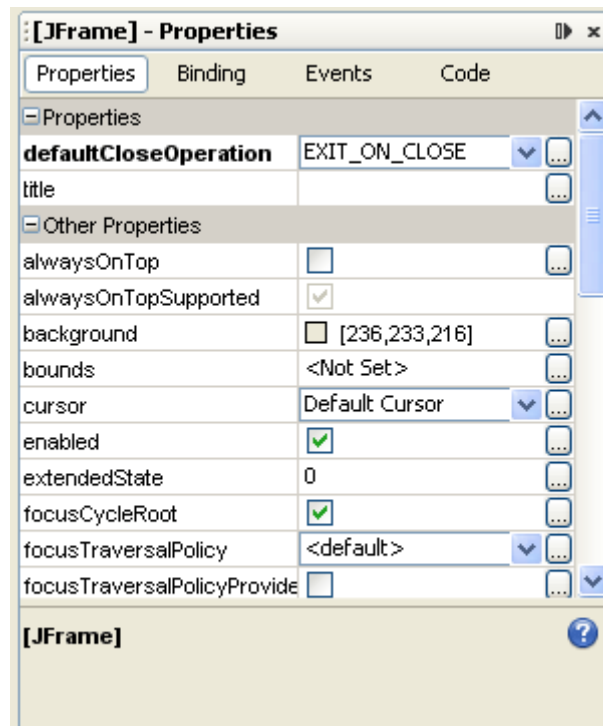


كما هو ظاهر فى الصورة أعلاه قد قمنا بإنشاء نموذج أو نافذة عن طريق الجافا ولا يحتوى على أى كائن اخر فقط مجرد نافذة تحتوى على ازار الاغلاق و التكبير و الاخفاء و لا تظهر إلا عندما يتم تشغيل النافذة.....و

ويوجد على يمين النافذة أو الفورم الذي قمنا بإنشاءه نافذتين.... الأولى تسمى **Palette** و هي تحتوى على جميع العناصر الرسومية الخاصة بالحزمة swing و الحزمة awt مثل الازرار و مربعات النصوص و القوائم وغيرها و سوف يتم شرح تلك العناصر في دورس لاحقة.....



أما النافذة الثانية هي نافذة **الخصائص** الخاصة بالفورم ومن خلالها نتستطيع تغيير بعض خصائص الفورم و هي كما في الشكل التالي.



ثالثا : التعرف على أهم خصائص jfarme

بعد ما قمنا بإنشاء اول نافذة لنا أخوانى الكرام تعالوا الان نتعرف على اهم خصائص هذا الكائن.. jfarme

1_الخاصية : default close operation وهى خاصة بعملية إغلاق النافذة من خلال زر الاخلاق وتأخذ أربع قيم :

- HIDE و تؤدي الى أخفاء الفورم أو النافذة عن الضغط على زر الاغلاق معنى ذلك انها تكون محملة فى ذكرة الجهاز و لكنها مخفية فقط
- EXIT_ON_CLOSE وتؤدي الى إغلاق التطبيق ككل بما فى ذلك باقى نوافذ التطبيق فى حالة الضغط على زر الاغلاق
- DO_NOTHING ولا تؤدي الى فعل اى شىء فى حالة قيام المستخدم بالضغط على زر الاغلاق.
- DISPOSE تؤدي تلك الخاصية الى أخفاء و اغلاق النافذة الحالية فقط دون باقى نوافذ التطبيق.

2_الخاصية. title وهى الخاصية السؤولة عن اعطاء عنوان للنافذة و هى تساوى caption فى الفجوال بيسك

3_الخاصية... alwaysOnTop تأخذ تلك الخاصية قيمة true أو false ووضيقتها جعل النافذة أو الفورم ظاهرا فوق جميع النوافذ فى حالة تحديدها, true و العكس صحيح.

4_الخاصية..... resizable و وظيفة تلك الخاصية انه عند تحديدها يستطيع المستخدم تغيير حجم النافذة أثناء التشغيل و العكس صحيح.

5_الخاصية.... name و تختص بإعطاء أسم برمجى للنافذة و الذى سوف تتعامل معه فى كتابة الاكواد

6_الخاصية.... type و تلك الخاصية تأخذ ٤ قيم منها Normal و تجعل النافذة بالشكل العادى مكون من زر الاغلاق و التكبير و التصغير UTILITY, ... وهى تجعل النافذة لا تحتوى الاعلى زر الاغلاق فقط,....

7_الخاصية..... undecorated و تلك الخاصية تقوم عمل نافذة بدون حواف

8_لتوسيط الفورم أو النافذة على الشاشة عليك النقر على التوبيب (Code) الموجود فى صفحة الخصائص form size policy قم بجعلها تساوى القيمة.... Genarate Resize code ...

كما يوجد العديد من الخصائص الاخرى التى تستطيع ان تكتشفها بنفسك

رابعا : أهم الاحداث الخاصة بانوافذ و الاطارات.

تعلمنا فى لغات برمجة اخرى مثل الفجوال بيسك أن الحدث هو الفعل المراد تنفيذ الكود عند تحقيقه.... فمثلا عند إغلاق النافذة نريد اظهار رسالة للمستخدم و بالتالى فإن إغلاق النافذة حدث وهكذا و يوجد عدة أحداث للنوافذ فى لغة الجافا ومن اجل اختيار حدث لكائن ما و ليكن jfarme الحالى مثلا وكتابه الكود بداخله نتبع ما يلى:

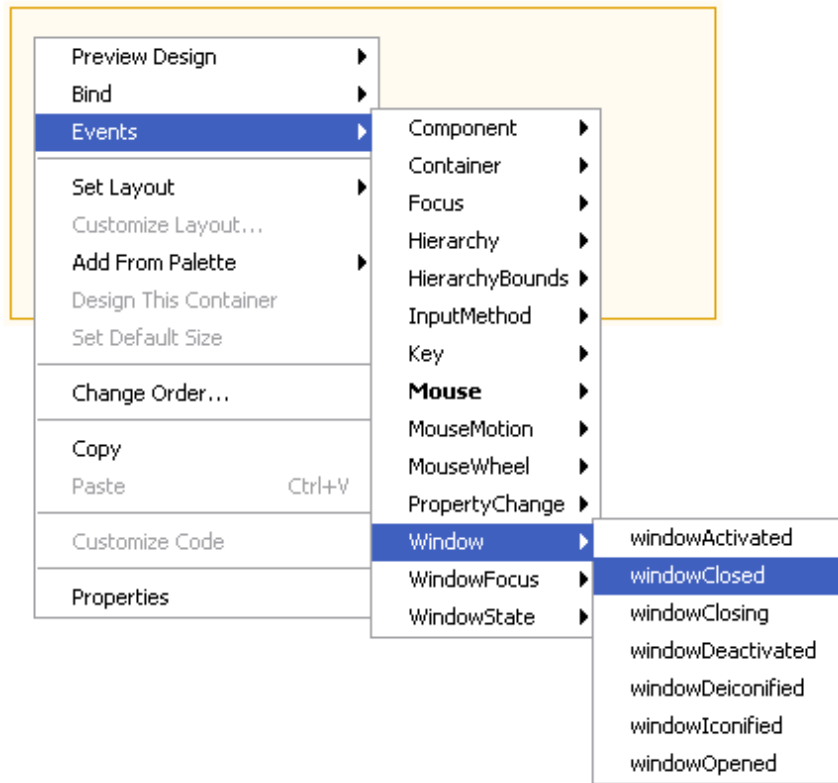
١ -حدد الكائن المراد التعديل او العمل عليه.

٢ -ننقر بزر الماوس الايمن عليه.

٣ -سوف تظهر لنا قائمة كبيره نختار events

٤ -سوف تظهر لنا قائمه اخرى كبيره نختار منها تصنيف الحدث وليكن Window

٥ -سوف تظهر لنا قائمه اخرى كبيره نختار منها الحدث المطلوب وليك WindowClosing كما فى الشكل التالى :



سوف يتم الانتقال بك الى صفحة الكود تحت الحدث المنشىء formWindowClosing لتقوم بكتابة الكود المراد تنفيذة عند هذا الحدث فمثلا نريد اظهار رسالة للمستخدم عند خروجه من البرنامج تفيدة بانه متأكد من الخروج من البرنامج و أم لا سنقوم بكتابة الكود تحت الحدث formWindowClosing كما فى الشكل التالى :

تكبير الصورة معاينة الأبعاد الأصلية.

```

10  @SuppressWarnings("unchecked")
11  Generated Code
37
38  private void formWindowClosing(java.awt.event.WindowEvent evt) {
39  JOptionPane.showMessageDialog(null, "0, "", "هل تريد الخروج من البرنامج ؟");
40  }
41
42

```

والنتيجة انه عندما يتم اغلاق الفورم أو النافذة يتم تنفيذ الكود كما فى الشكل التالى :

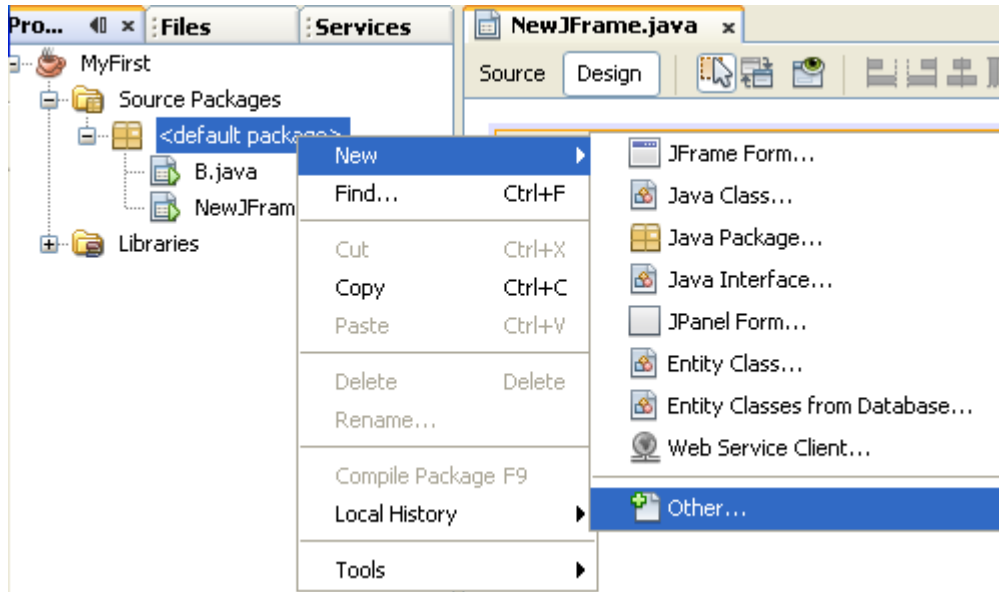


والآن تعالوا نتعرف على أهم أحداث النافذة:

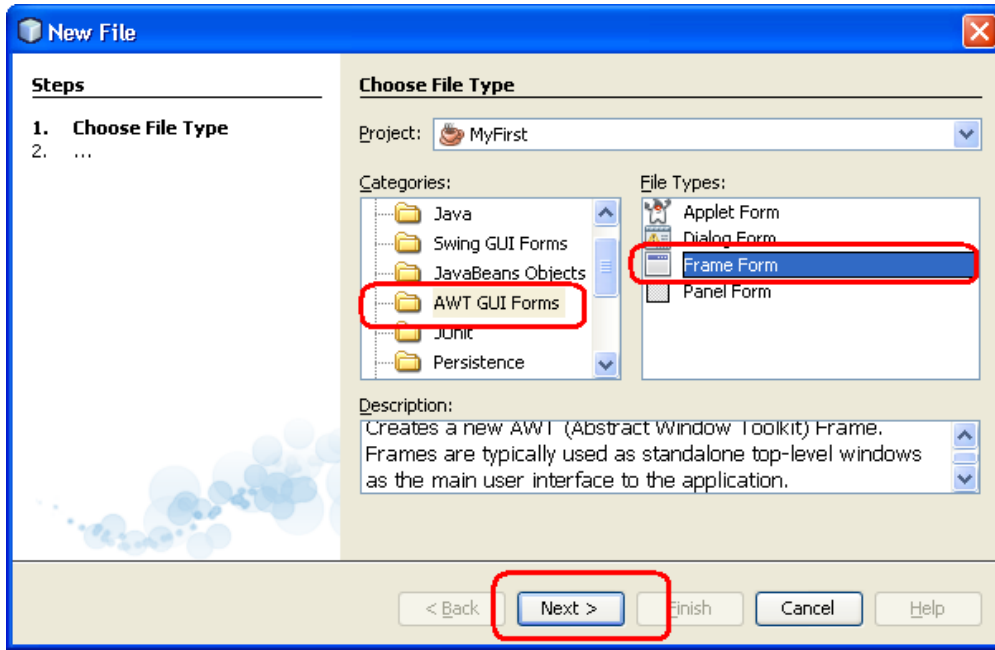
الحدث windowOpened وهو حدث فتح النافذة
الحدث windowClosing وهو حدث يستخدم اثناء بدء عملية إغلاق النافذة...مثل اظهار رسالة قبل عملية إغلاق الفورم أو النافذة
الحدث windowClosed وهو حدث يستخدم بعد عملية إغلاق الفورم وليس اثناء الاغلاق.
الحدث windowActivated وهو حدث يستخدم في حالة كون النافذة نشطة.
الحدث windowGainedFocus و الحدث windowLostFocus وهو حدث يستخدم في حالة كون النافذة التركيز أو فقد التركيز عليها
كما يوجد العديد من الاحداث خاصة بحركة الماوس على النافذة منها MouseClicked و ايضا احداث خاصة بمفاتيح الكيبورد و تستطيع ان تكتشفها بنفسك.

خامسا: إنشاء النوافذ الكلاسيكية fawme الخاصة بالحزمة awt

تحدثنا في بداية الدرس انه يوجد نوعين من النوافذ في لغة الجافا ...نوافذ تتبع الحزمة swing وقد شرحنا كيفية انشائها و تعرفنا على اهم خصائصها ... الان دعونا نتعلم كيفية إنشاء النوافذ التي تتبع الحزمة awt. _قف بالماوس على ايقون default packages ثم كليك شمال بالماوس ثم New ثم Other كما في الشكل التالي:



_سوف يظهر لنا المربع الحوارى التالى.....قم بتحديد AWT GUI Forms ثم أختَر من file type العنصر Frame Form كما فى الشكل التالى :



_سيظهر لك مربع حوارى آخر لوضع اسم للنافذة الجديدة قم بفعل ذلك ثم اضغط على زر إنهاء المعالج (finish) وهكذا تكون قد انشئنا نافذة كلاسكية تابعة للحزمة awt وطبعاً تلك النافذة تكون اخف على النظام من النوافذ التابعة للحزمة swing وتكون بالشكل التالى :



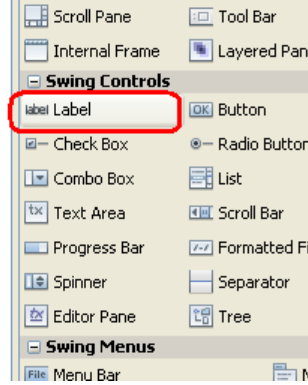
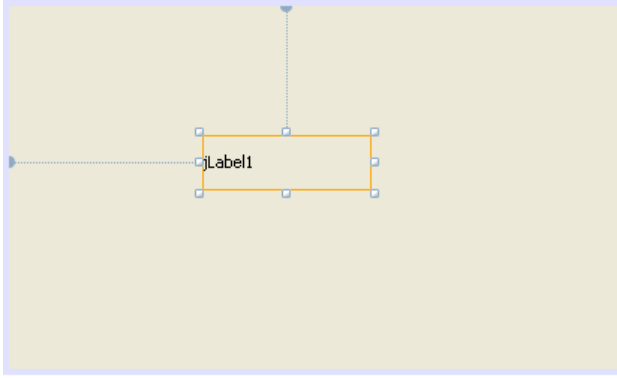
وطبعاً تستطيع تغيير خلفية النافذة من خلال الخاصية bacegraond ثم تقوم بأختيار اللون المناسب.... كما ان معظم الخصائص و الاحداث التى سيق و شرحناها تنطبق على تلك النوافذ....

الفصل الثاني : حقل التسمية JLabel و زر الامر JButton

فى الدرس السابق تعلمنا وتحديثنا عن الاطارات و النوافذ و اليوم ان شاء الله تعالى سنتعلم كيفية التعامل مع حقول التسمية و زر الامر

أولا : العنصر الرسومى JLabel

يوفر العنصر الرسومى حقل التسمية JLabel تعليمات نصية أو معلومات فى فى واجهة المستخدم الرسومية و ذلك عن طريق أظهار سطر نصى أو صورة أو كليهما للقراءة فقط ولا يمكنك التعديل عليه اثناء التشغيل.....ولكى تتمكنك من إضافة هذا العنصر لمشروعك كل ما عليك فعله هو الضغط على ايقونة العنصر فى نافذة الادوات ثم بعد ذلك وضع المؤشر على المنطقة المراد وضعه فى نافذة برنامجك كما فى الصورة التالية:



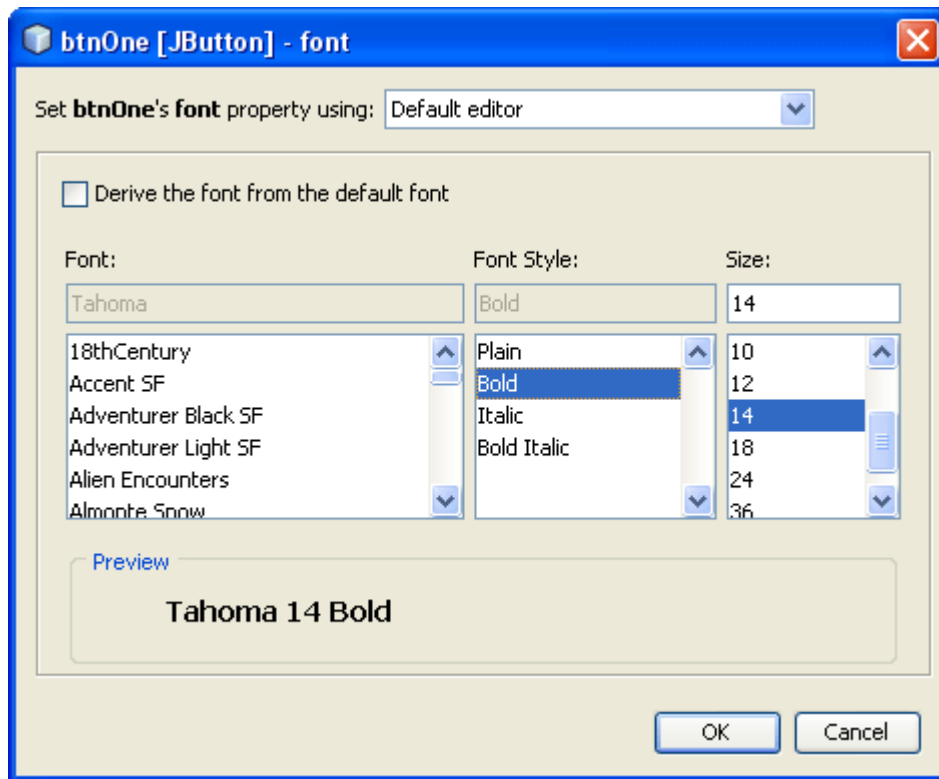
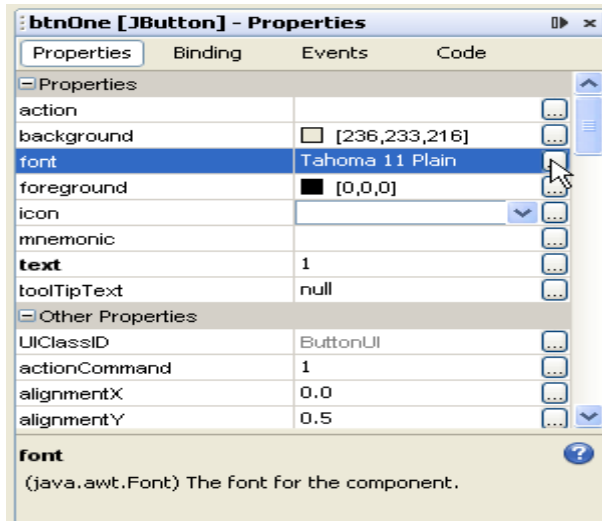
أهم خصائص العنصر الرسومى JLabel

يوجد الكثير من الخصائص لحقل التسمية لكنى سأذكر فقط أهم الخصائص شائعة الاستخدام لهذا العنصر

1_الخاصية variable name والتي نستطيع من خلالها وضع أسم للعنصر للتعامل معه فى صفحة الكود

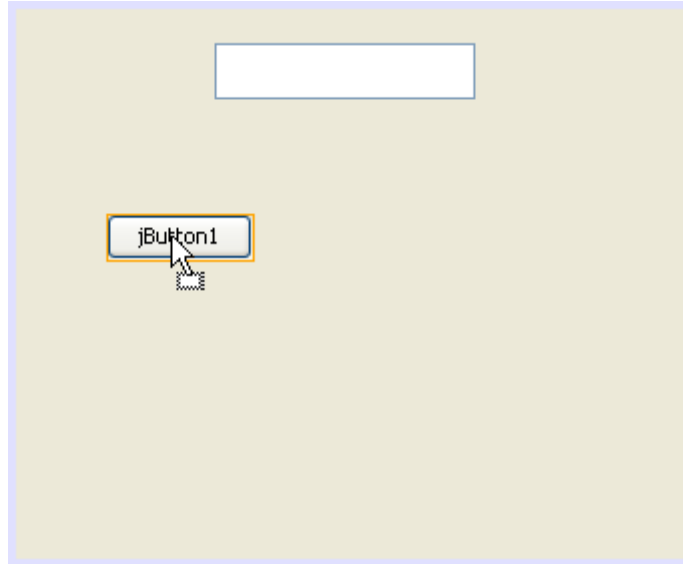
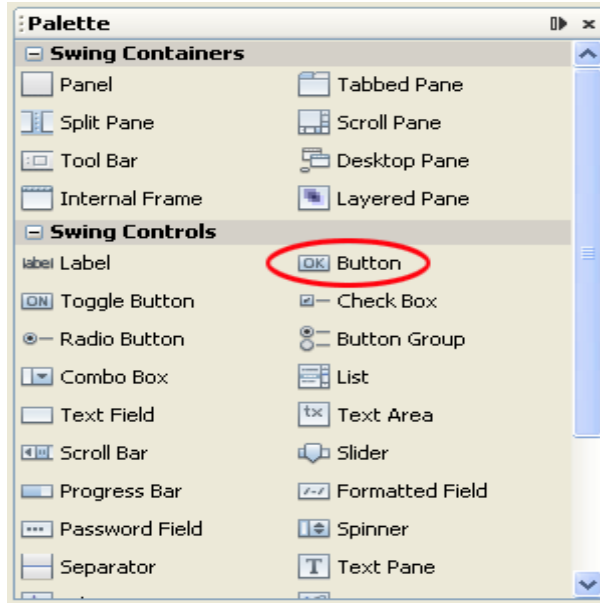
2_الخاصية variable Modifiers و التي من خلالها جعل العنصر عام أو خاص

3_الخاصية font والتي من خلالها نستطيع تحديد الخصائص المميز للخط الحجم و النوع



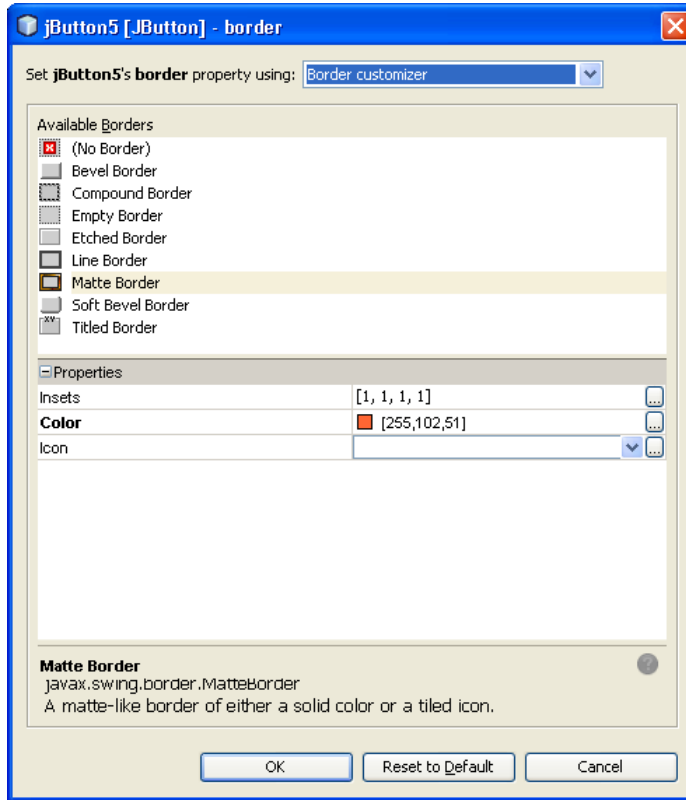
4_ الخاصية foreground وهي المسؤولة لاعطاء لون معين للخط
 5_ الخاصية text و التي تضع اسم للعنصر يكون ظاهرا للمستخدم
 6_ الخاصية icon والتي من خلالها يتم وضع صورة مصغرة للعنصر

ثانيا : العنصر الرسومي JButton زر الامر



زر الامر هو عنصر رسومي يضغطه المستخدم لتنفيذ كود معين....ومن اهم خصائص هذا العنصر الرسومي مايلي:

- 1_الخاصية variable name والتي نستطيع من خلالها وضع اسم للعنصر للتعامل معه في صفحة الكود
- 2_الخاصية border والتي من خلالها نستطيع وضع إطار لجميع للزر كما في الشكل التالي



- 3_ الخاصية rollover icon و التي تعطى صورة عند مرور الماوس على الزر
 4_ الخاصية icon والتي من خلالها يتم وضع صورة مصغرة للعنصر
 5_ الخاصية Horizontal Size و Vertical Size و اللتان تقوم بتحديد حجم الزر عن طريق وضع القيمة بها
 وهناك ايضا الخاصية icon و text و font و foregraoud

اهم أحداث العنصر الرسومي: يوجد الكثير من الاحداث لزر الامر ومنها مثلا...

- أحداث الضغط على زر الامر Keyboard events
- أحداث حركة الفارة mouse move events
- أحداث الضغط على الفارة mouse click events
- أحداث الضغط على الزر Button click events

ويتم الوصول لتلك الاحداث كما في الصورة التالية



وهو ان يتم تحديد زر الامر المراد تنفيذ كود في حدث معين ثم كليك شمال فتظهر لنا قائمة الاحداث كما في الصورة السابقة و ينتقل بينا المؤشر من صفحة التصميم الى صفحة كتابة الشفرة....وفى المثال السابق قمنا بأختيار الحدث actionPerformed وهو يساوى ايضا الحدث.... MouseClicked سندخل صفحة الشفرة و نكتب الكود التالى:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "أهلا وسهلا");
}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}
```

الفصل الثالث : مكونات النصوص Text Components

اليوم إن شاء الله تعالى سنتعلم سويا كيفية التعامل مع مكونات النصوص في الجافا Text Components والتي تشمل

JTextField, JTextArea ,JPasswordField, JFormattedTextField,

1_العنصر الرسومي: JTextField

هذا العنصر الرسومي عبارة عن مربع لعرض و تحرير سطر نصي واحد....ومن اهم الخصائص التي يمتاز بها هذا العنصر الرسومي ما يلي:

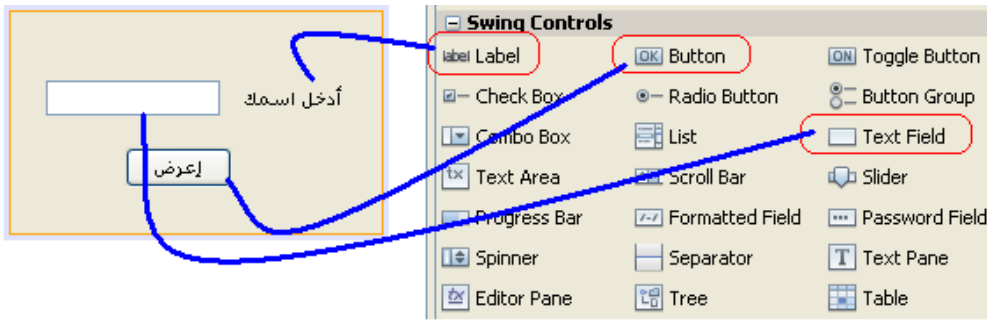
- الخاصية... editable و التي تجعل العنصر الرسومي له القدرة على التحرير اثناء التشغيل من عدمه
- الخاصية... selectioncolor و التي من خلالها تستطيع تحديد اللون المخصص عند تحديد النص
- الخاصية... selecttextcolor و التي من خلالها ايضا وضع لون معين للخط عند تحديد النص.

هذا بالإضافة للخصائص الرئيسية الاخرى والتي تم شرحها سابقا في العناصر الرسومية الاخرى...مثل الخط و حجم الخط و لون الخط ولون الخلفية و غيرها.

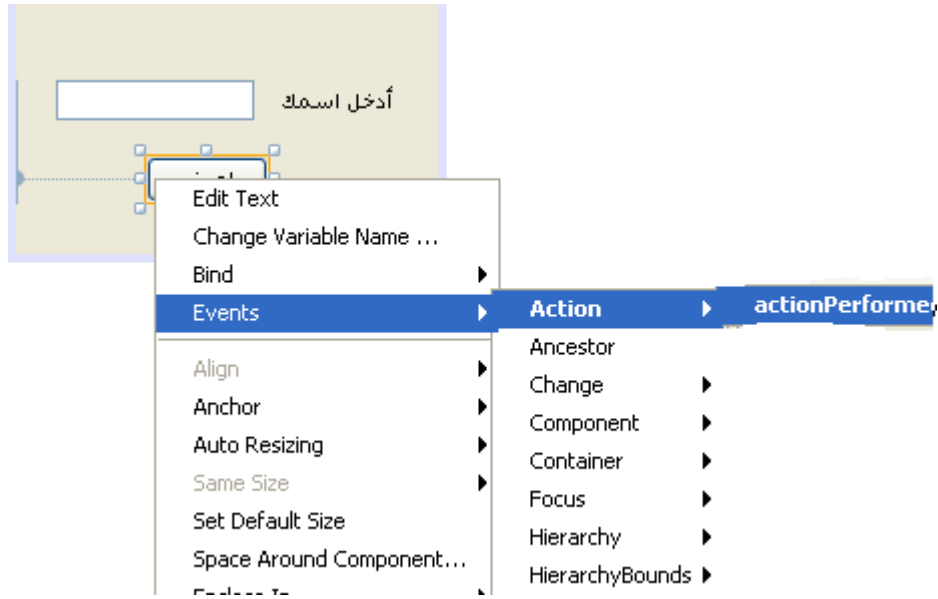
مثال : قم بإنشاء برنامج يقوم المستخدم بكتابة اسمه في مربع النص و عند الضغط على زر (اعرض) يتم إظهار رسالة ترحيب للمستخدم...

الحل:

سنقوم بإنشاء نافذة ونضع عليها عدد واحد حقل تسمية JLabel وعدد واحد زر امر JButton و عدد واحد حقل نصي من نوع JTextField كما في الشكل التالي:



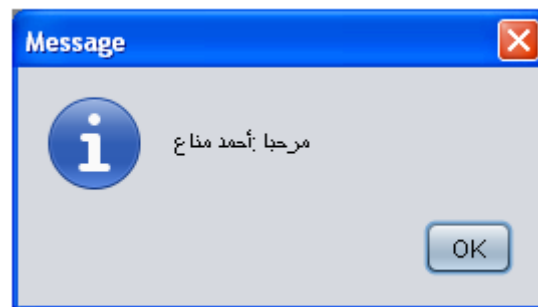
سنقوم بإنشاء الحدث ActionPerformed لزر الامر كما في الشكل التالي:



تحت هذا الحدث نكتب الكود التالي:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String name=jTextField1.getText();
JOptionPane.showMessageDialog(null, ":"+ name);
}
```

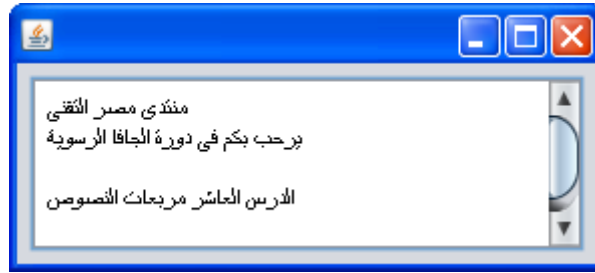
والنتيجة عند تشغيل المشروع تظهر الشاشة التالية و التي تطالبك بوضع اسمك و عند الضغط على زر الامر تظهر رسالة ترحيب بأسمك... كما في الأشكال التالية:



و بالتالى فإن الدالة **getText** هى التى من خلالها الحصول على القيمة الموجودة فى مربع النص
TextField

2 _ العنصر الرسومى JTextArea ...

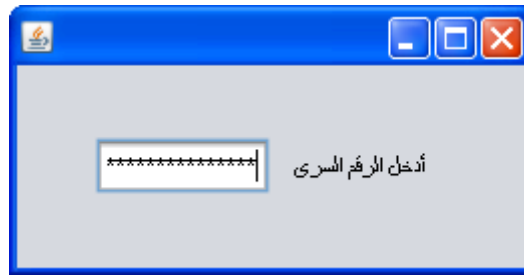
هذا العنصر الرسومى عبارة عن مربع ايضا لعرض وتحرير عدة اسطر و ليس سطر واحد
مثل JTextField لذلك اذا اردت ان يقوم مستخدم برنامج بكتابة عدة اسطر نصية لذلك يجب عليك
استخدام هذا العنصر الرسومى و يتميز هذا العنصر الرسومى بكافة الخصائص السابقة
بالاضافة الى خاصية شريط التمرير الافقى و الراسى .. كما فى الصورة التالية:



3 _ العنصر الرسومى JPasswordField

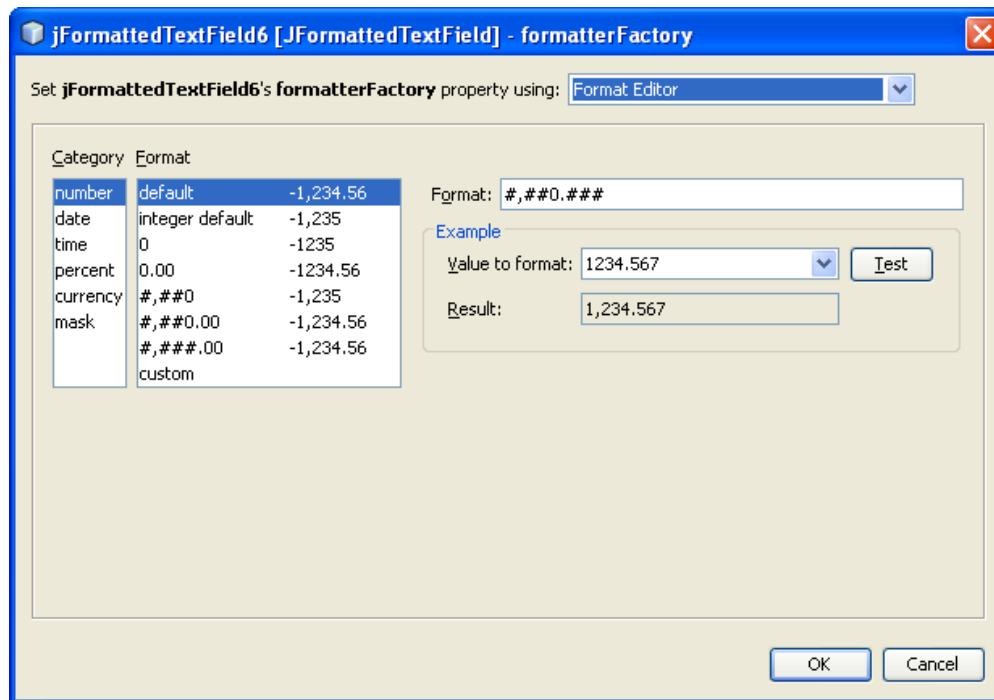
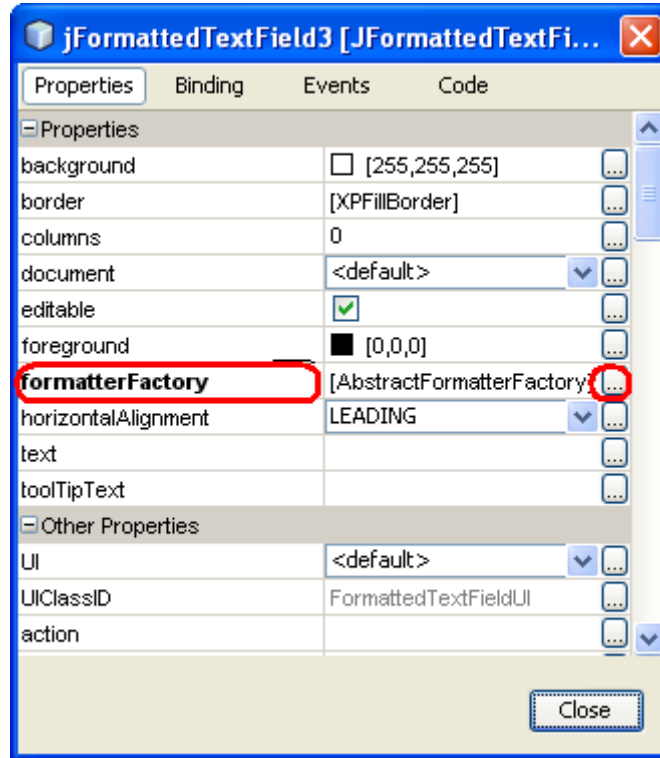
هذا العنصر الرسومى عبارة عن مربع نص لعرض وتحرير المدخلات باشكل و رموز اخرى عندما يقوم
المستخدم بكتابة المدخلات فية و الرمز الافتراضى لتلك البيانات هى علامة النجمة (*). ويستخدم هذا العنصر
الرسومى فى كتابة الارقام السرية لبرنامج مثلا ...

ومن اهم الخصائص المميزة لهذا العنصر هى خاصية EchoChar و التى يمكن من خلالها تغيير الرموز و
الاشكال المراد ظهورها بدلا من الرمز الافتراضى النجمة (*).

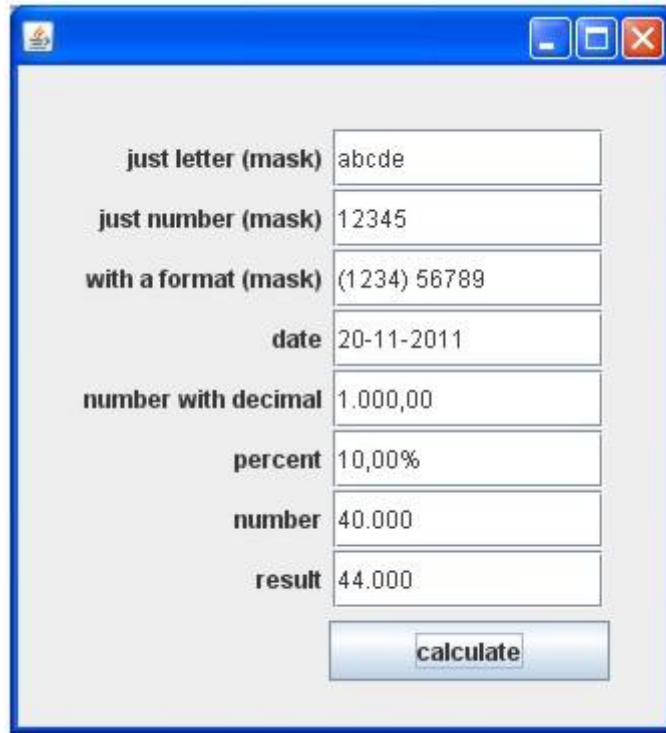


4 _ العنصر الرسومي JFormattedTextField

هذا العنصر الرسومي هو عبارة عن مربع نص لتحرير و عرض البيانات بتنسيقات مختلفة... كأن تجعله يقوم بعرض و تحرير ارقام فقط او تواريخ او اى صيغة اخرى انت تحدها... و يمكن فعل ذلك عن طريق الخاصية formatterFactory و التى تهظر لك مربع حوارى لاستخدام التنسيقات المختلفة كما فى الشكل التالى:



والنتيجة....



The screenshot shows a software window with a blue title bar and standard Windows window controls (minimize, maximize, close). The window contains several input fields with labels on the left and values in the boxes:

just letter (mask)	abcde
just number (mask)	12345
with a format (mask)	(1234) 56789
date	20-11-2011
number with decimal	1.000,00
percent	10,00%
number	40.000
result	44.000

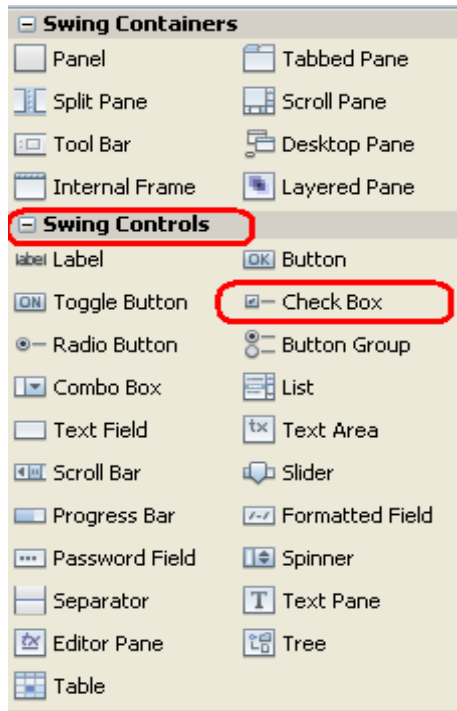
At the bottom of the window is a button labeled "calculate".

الفصل الرابع : مربع الاختيار المتعدد jCheckBox و خانة الاختيار jRadioButton

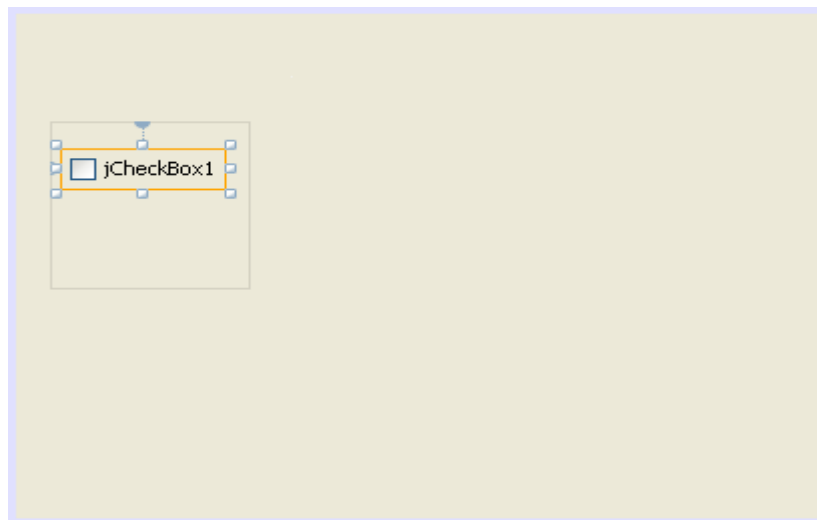
في هذا الدرس إن شاء الله نواصل معرفة المزيد عن واجهة المستخدم الرسومية في الجافا و سنتعلم كيفية التعامل مع عنصرين مهمين هما مربع الاختيار المتعدد jCheckBox و خانة الاختيار jRadioButton

أولا : مربع الاختيار المتعدد jCheckBox

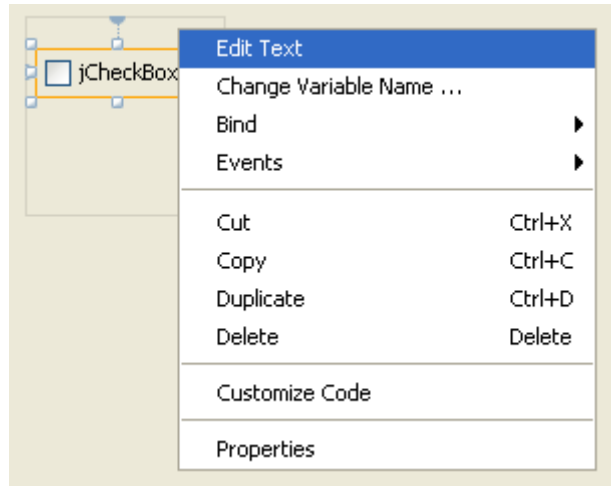
هو عبارة عن عنصر تحكم يسمح للمستخدمين باختيار العناصر عن طريق تحديد أو إلغاء تحديد مربع التجزئة الذي يوفرها في الاختيار لاختيار أكثر من عنصر في وقت واحد... و يمكنك الحصول على مربع الاختيار المتعدد من خلال قائمة Swing Containers كما في الشكل التالي:



الآن قم بسحب العنصر بالماوس ثم اسقطه على نافذة مشروعك كما في الشكل التالي:



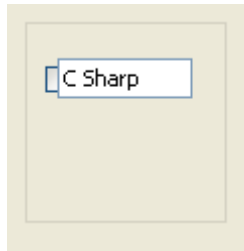
وكما تلاحظون شكل مربع الاختيار المتعدد على نافذة المشروع يأخذ النص jCheckBox1 هو النص الافتراضي للعنصر.و يمكنك تغيير هذا النص إما من خلال نافذة الخصائص ، أو بالنقر بزر الماوس الأيمن على مربع الاختيار. و من القائمة التي تظهر، حدد تحرير النص في الصورة أدناه



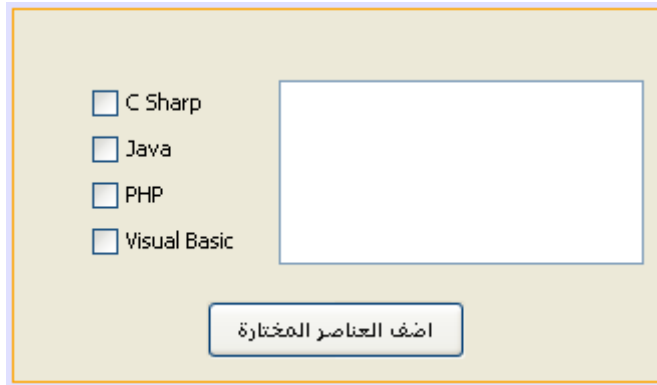
عند النقر على تحرير النص، سيتم تسليط الضوء على النص الافتراضي كما في الشكل التالي:



وهكذا نستطيع تغيير النص للعنصر وليكن C Sharp كما يلي:



سنقوم بعد ذلك بإضافة العنصر JTextArea على النموذج بحيث عندما يقوم المستخدم بتحديد لغة أو لغات وعند الضغط على زر الامر (اصف) سوف يتم سردها في منطقة النص...JTextArea اذن الشكل النهائي للنافذة ستكون كما في الصورة التالية:



نحن الان بحاجة لمعرفة أي خانة الاختيار تم اختيارها من قبل المستخدم حتى نتمكن من إضافتها في العنصر الرسومي.TextArea لوفى مربع الاختيار يحتوي على خاصية تسمى () isSelected التي ترجع حقيقية منطقية لمعرفة إذا تم تحديد خانة الاختيار أم لا ... سنقوم ايضا باستخدام جملة IF و ذلك على النحو التالي:

الرمز:

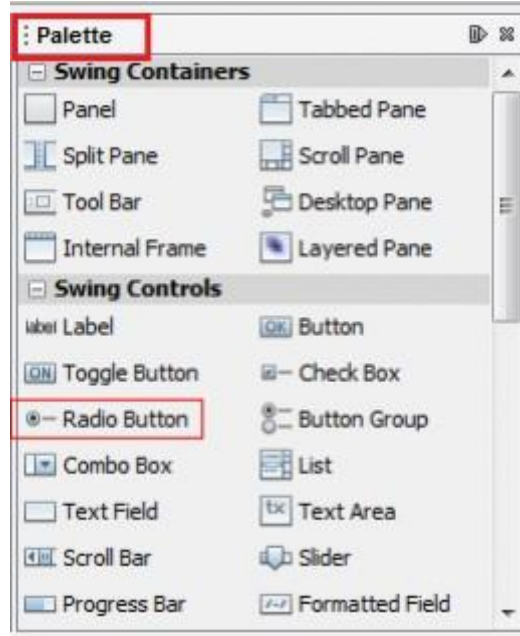
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s="";
if (jCheckBox1.isSelected()){
s=s+""+jCheckBox1.getText()+"\n";
}
if (jCheckBox2.isSelected()){
s=s+""+jCheckBox2.getText()+"\n";
}
if (jCheckBox3.isSelected()){
s=s+""+jCheckBox3.getText()+"\n";
}
if (jCheckBox4.isSelected()){
s=s+""+jCheckBox4.getText()+"\n";
}
jTextArea1.setText(s);
}
```

وكما ترون في الكود السابق قمنا بالاعلان عن المتغير s فإذا تم تحديد خانة الاختيار نحصل على نص التسمية التوضيحية من مربع الاختيار. ثم يتم تخزين هذا النص في المتغير s جنبب الى جنب مع وضع حرف /n و ذلك لوضع المؤشر في سطر جديد ثم استخدمنا الدالة setText للعنصر الرسومي jTextArea1 و ذلك لوضع قيمة المتغير s في العنصر الرسومي قم الان بتشغيل برنامج و ستجد ما يلي:

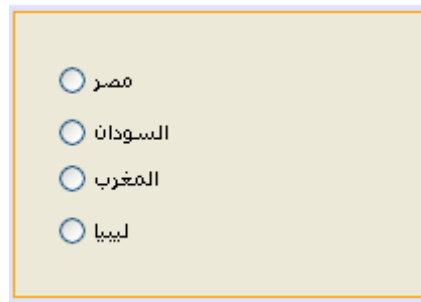


ثانيا : خانة الاختيار jRadioButton

خانة الاختيار هي احد العناصر الرسومية و التي تسمح لمستخدم باختيار عنصر واحد فقط على عكس مربع الاختيار المتعدد الذي تم شرحه اعلاه.....ويمكن الحصول على خانة الاختيار jRadioButton من قائمة Swing Containers كما بالشكل التالي:



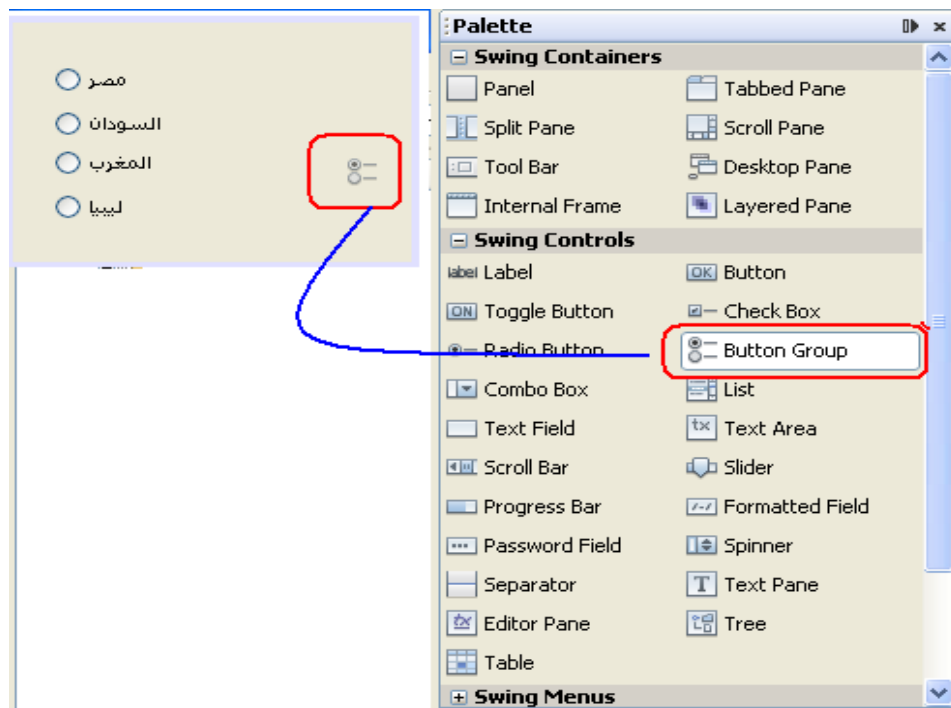
و يمكن تحرير النص الافتراضي لتسمية jRadioButton كما فعلنا مع خانات الاختيار اعلاه....قم باضافة عدد من خانات الاختيار وقم بتغيير التسمية الافتراضية كما في الشكل التالي:



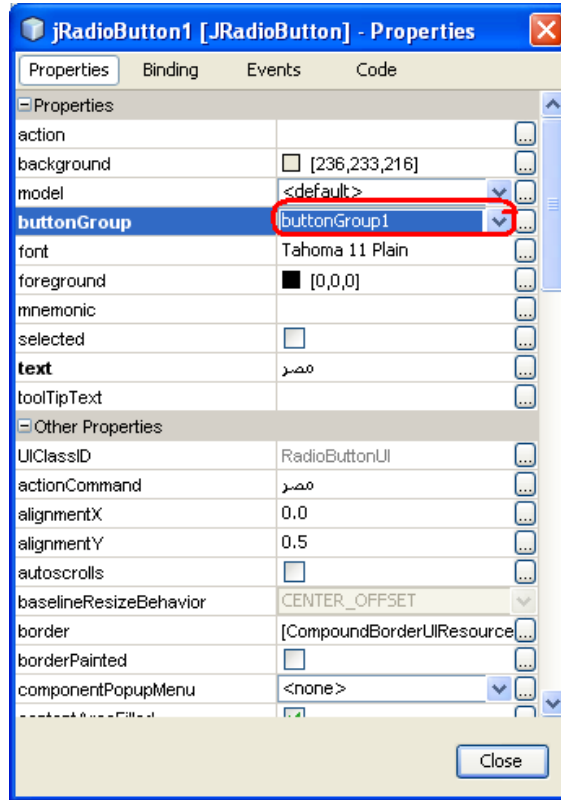
وعند تشغيلك لمشروعك ستجد أن خانة الاختيار jRadioButton لا تعمل بشكل صحيح...أذن انها من المفترض أن تجعل المستخدم غير قادر على اختيار إلا عنصر واحد فقط من تلك العناصر...لكننا لو قمنا بتشغيل مشروعنا الان سنجد ان المستخدم قادر على اختيار اكثر من عنصر كما في الشكل التالي:



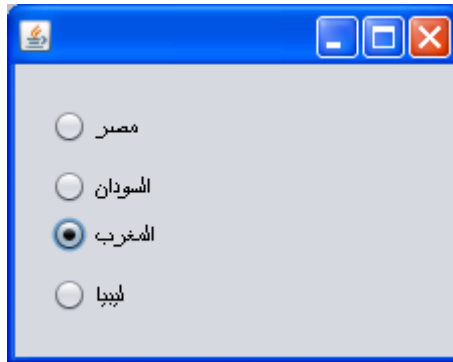
اذن ولحل تلك المشكلة ما عليك سوى الرجوع الى قائمة Swing Containers ستجد المكون ButtonGroup قم بسحبه بالماوس و ضعه على النافذه ...كما في الشكل التالي:



بعد اسقاط العنصر الرسومي ButtonGroup على النافذة ستجد أنه لم يعد ظاهرا عليها فلا تقلق من ذلك الان قم بتحديد خانة الاختيار jRadioButton ومن نافذة الخصائص ستجد الخاصية ButtonGroup ومن خلالها ستحدد المجموعة التي تريد أن ينتمى اليه خانة الاختيار هذه وذلك كما في الشكل التالي:



كرر تلك العملية لجميع خانات الاختيار في مشروعك و بعدها شغل برنامج ستجد أنه يعمل بكفاءة و لن يستطيع المستخدم حينها اختيار اكثر من عنصر كما في الشكل التالي:



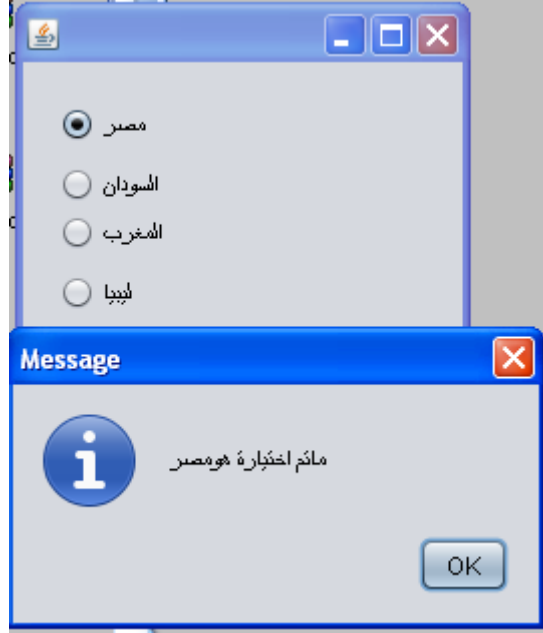
وللحصول على القيمة التي تم اختيارها من خلال خانة الاختيار تلك نستخدم الدالة أو الطريقة `isSelected` كما في الكود التالي:

الرمز:

```
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jRadioButton1.isSelected()){
        String textradio = jRadioButton1.getText();

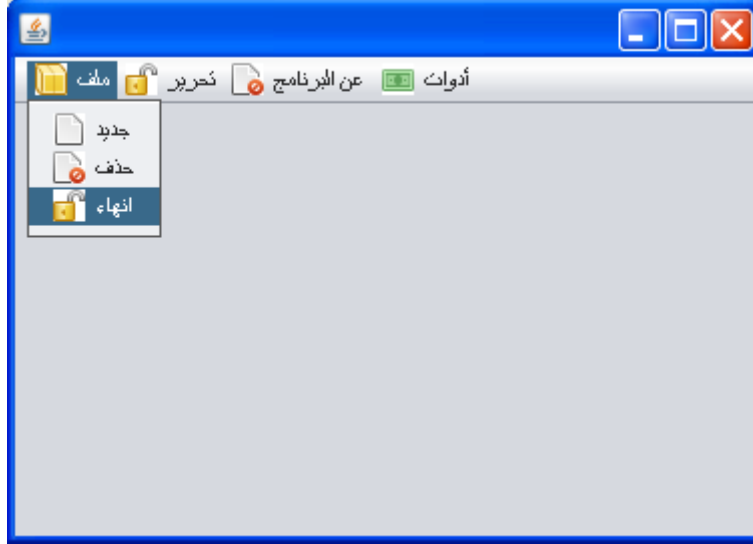
        JOptionPane.showMessageDialog(null, "" + textradio);
    }
}
```


و النتيجة كما في الشكل التالي:



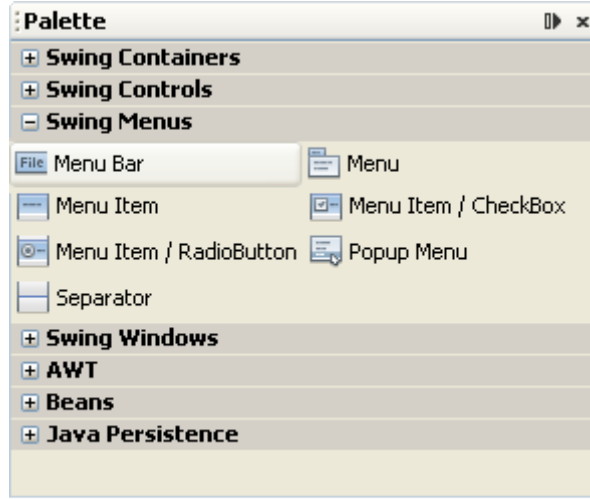
الفصل الخامس : اشرطة القوائم Menu Bar

ومالزنا نتحدث عن واجهة المستخدم الرسومية ..اليوم سنتحدث عن العنصر الرسومي) اشرطة القوائم..(Menu Bar و التي تمكنك من إضافة قوائم على نافذة مشروعك بحيث تسمح للمستخدم باختيار من عدة عناصر وتشمل مثلا قائمة ملف قائمة تحرير ،....الخ غير ذلك كما في الصورة التالية:

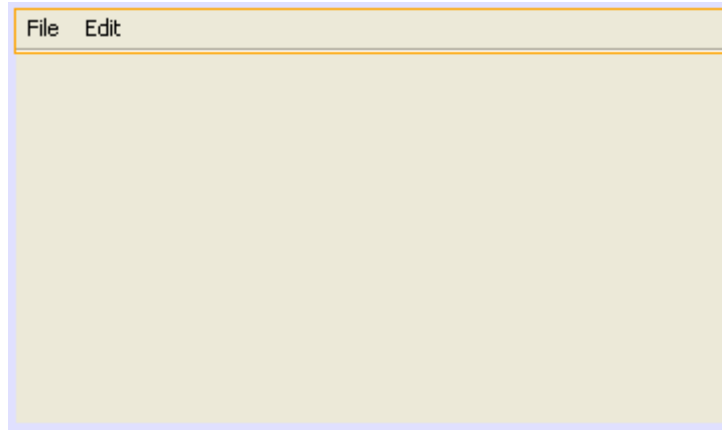


ولكى يتم الحصول على الشكل أعلاه أتبع الخطوات التالية:

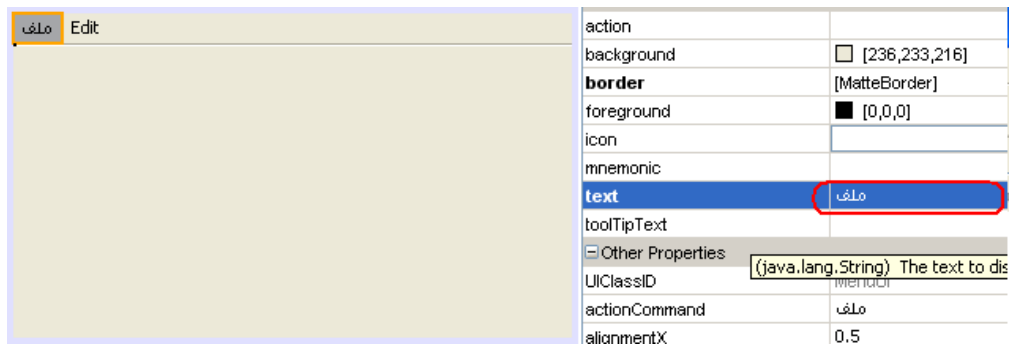
1_قم بإضافة Menu Bar الى نافذة مشروعك و ذلك بسحب العنصر من نافذة palette تحت التصنيف Swing controls ثم اسقاطه على نافذة المشروع كما في الشكل التالي:



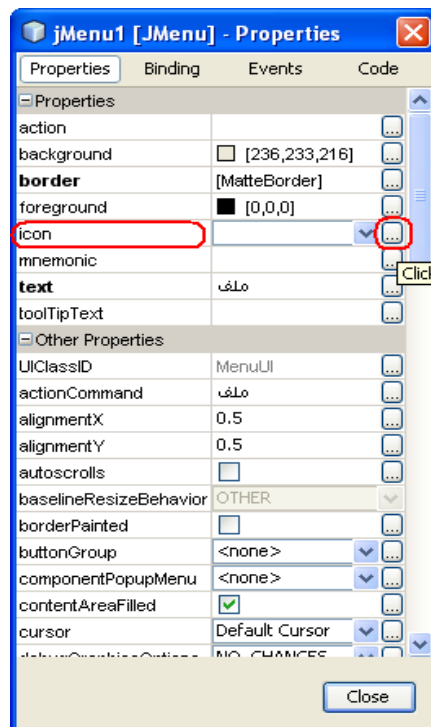
وعند اسقاط العنصر الرسومي Menu Bar على نافذة مشروعك ستجد أن العنصر يحتوي على قائمة ملف و تحرير كما في الشكل التالي:



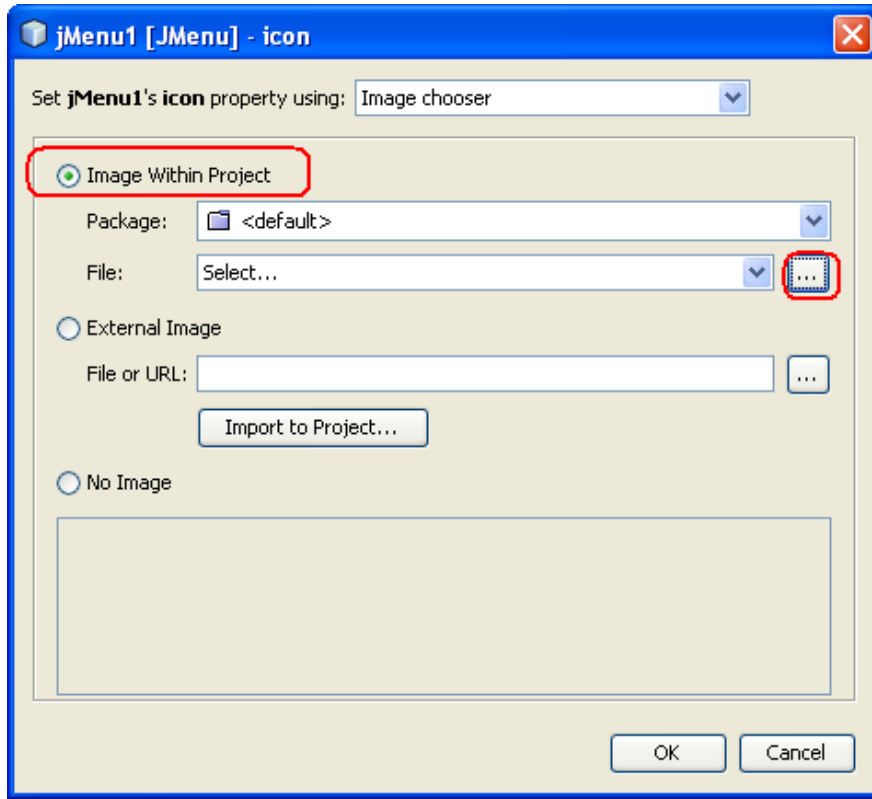
2_ولكى تتمكن من تغيير النصوص الافتراضية لقوائم العنصر الرسومي Menu Bar الى الاسماء التي تريدها قم بتحديد القائمة file ثم من نافذة الخصائص حدد الخاصية text و قم بتغيير مسمى القائمة كما فى الشكل التالى:



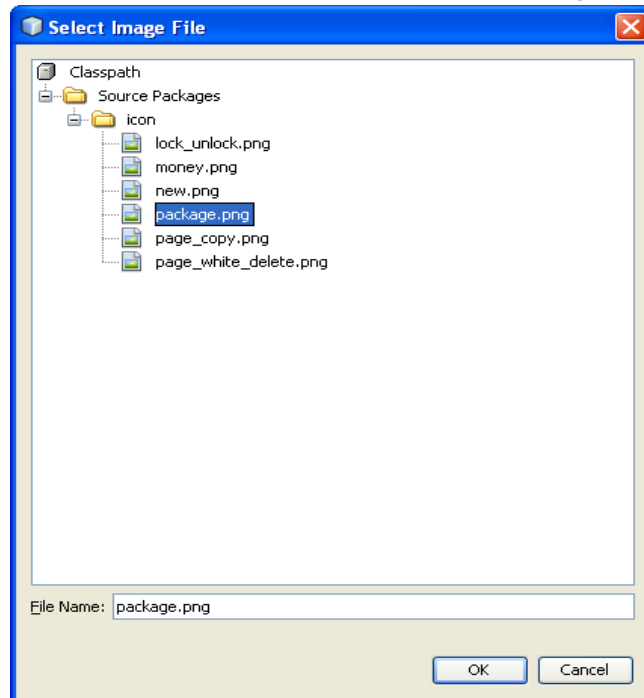
3_يمكنك ايضا إضافة ايقونة للقائمة ملف.... فقط قم بتحديد القائمة ومن نافذة الخصائص حدد الخاصية icons كما فى الصورة التالية:



من الشكل أعلاه قم بالضغط على الزر الصغير الذى بالجانب الايمن من الخاصية يتم اظهار المربع الحوارى التالى:



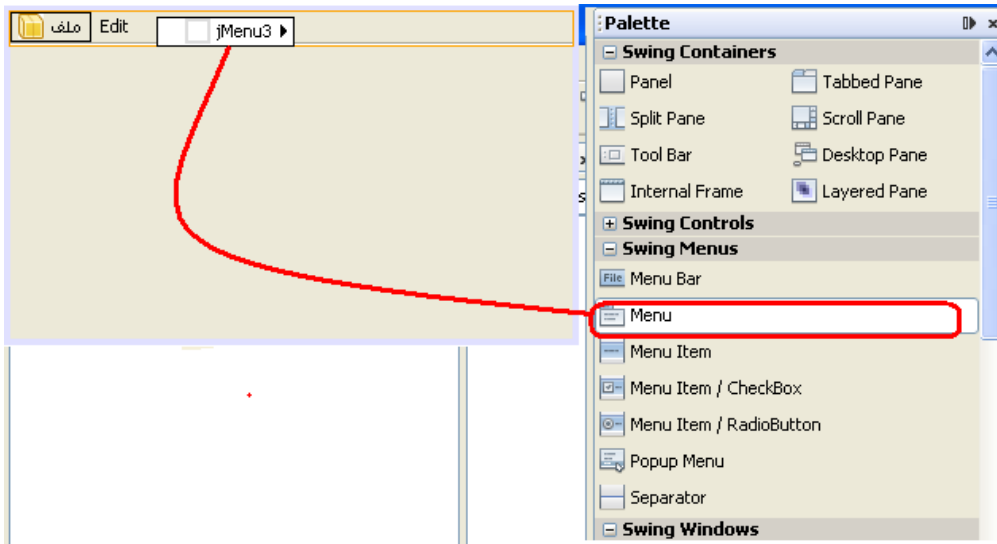
هذا المربع الحوارى يتيح لك كيفية ادراج صورة للعنصر اما من خلال مجلد صور مرفق مع البرنامج او عن طريق مجلد اخر او حتى رابط انترنت لذلك و من الافضل لنا هو ادراج مجلد صور به جميع ايقونات وصور البرنامج و ذلك ليتم ارفاقه مع المشروع اينما ذهب و لكيفية عمل ذلك ما عليك فقط الا ادراج هذا المجلد داخل مجلد البرنامج المسمى.... src فعلى سبيل المثال قمت بعمل مجلد يسمى icon ثم وضعته داخل المجلد src الخاص بمشروعى و بالتالى عند الضغط على زر استعراض كما فى الشكل السابق سأجد مجلد الصور الخاصى بى و سأختار منه الصورة التى اريدها للقائمة كما فى الشكل التالى:



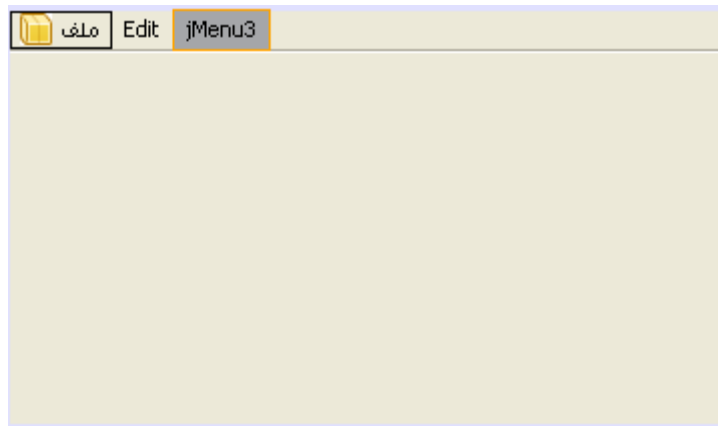
و النتيجة كما يلي:



4_ إضافة قوائم أخرى.... لكي تتمكن من إضافة اي قائمة اخرى لشريط Menu Bar ما عليك اللي التركيز على قائمة الادوات palette ثم تقوم بسحب العنصر الرسومي menu ووضعها على شريط القوائم كما في الشكل التالي:



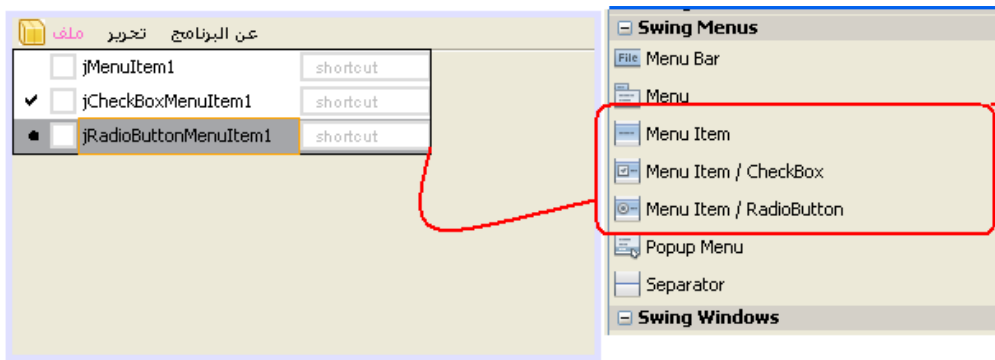
ليصبح بالشكل التالي بعد اسقاطه على نافذة مشروعك و يمكنك الان تغيير اسمه ووضع ايكون له كما فعلنا اعلاه..



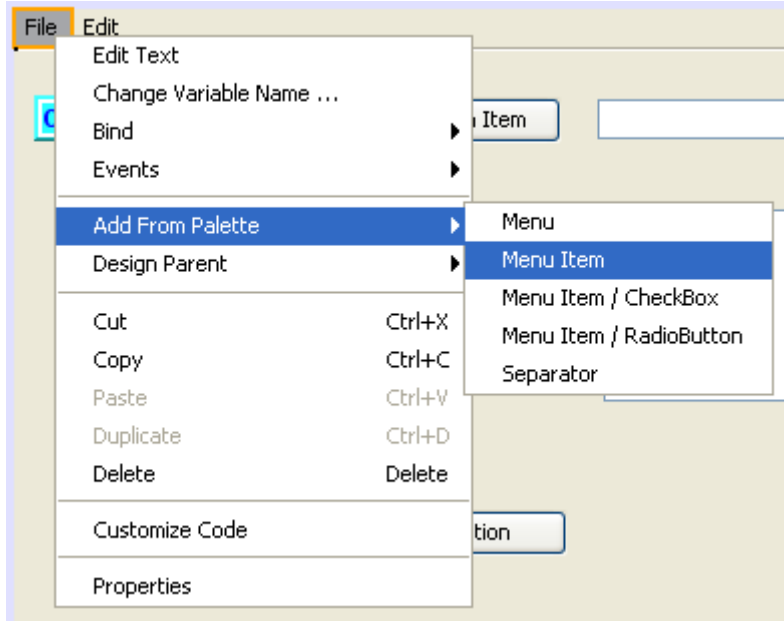
5_ كيفية إضافة عناصر للقوائم.... يوجد ثلاث أنواع من العناصر التي تستطيع إضافتها لاي قائمة وذلك على النحو التالي:

- أ_ العنصر... Menu Item هو عنصر رسومي يحتوى على بيانات عند الضغط عليه القائمة الرئيسية يتم اظهاره
- ب_ العنصر... CheckBoxMenuItem هو نفس العنصر السابق ولكنه يشمل علامة (صح) عند اختيار المستخدم له
- ج_ العنصر... RadioButtonMenuItem وهو يشبه العناصر السابقة ولكنه يشمل علامة (•) عند اختيار المستخدم له

والان تعالوا نتعلم كيفية اضافة تلك العناصر لاي قائمة... يتم ذلك اما من خلال النافذة palette وذلك بسحب تلك العناصر و اسقاطها تحت القائمة المراد ان تظهر بها كما فى الشكل التالي:



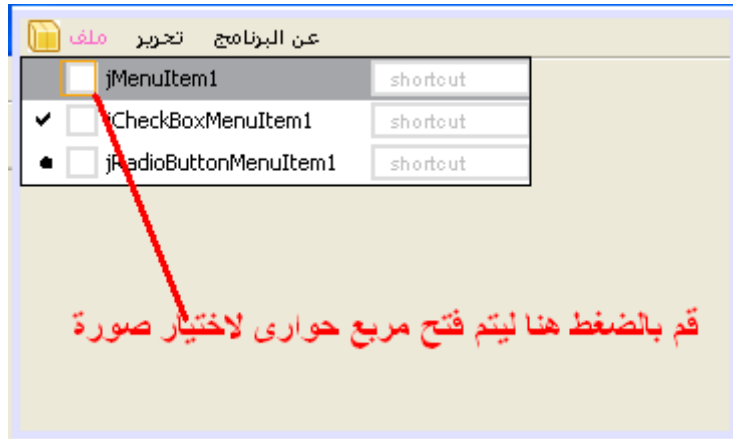
او من خلال تحديد القائمة المعنية ثم كليك شمال ثم نختار الامر Add From Palette ثم نختار العنصر المراد وضعة للقائمة كما فى الشكل التالي:



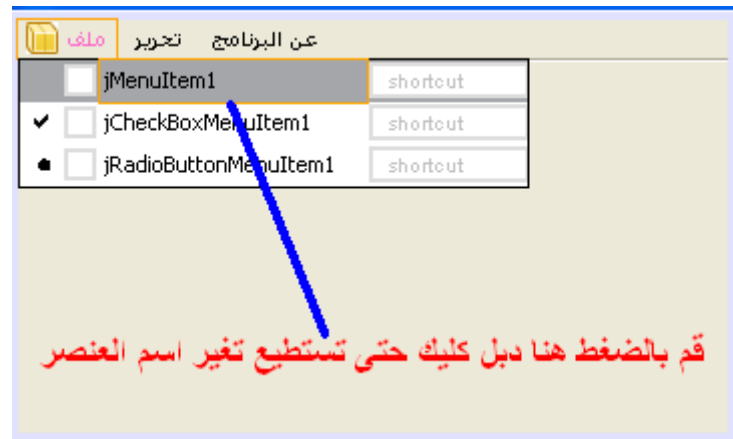
واى عنصر من العناصر السابقة له ثلاث مواضع مهمة....

أ_ الموضع icon لوضع صورة للعنصر الفرعى... قم بتحديد العنصر ثم اضغط على موضع الايقونة دبل كليك ليتم فتح

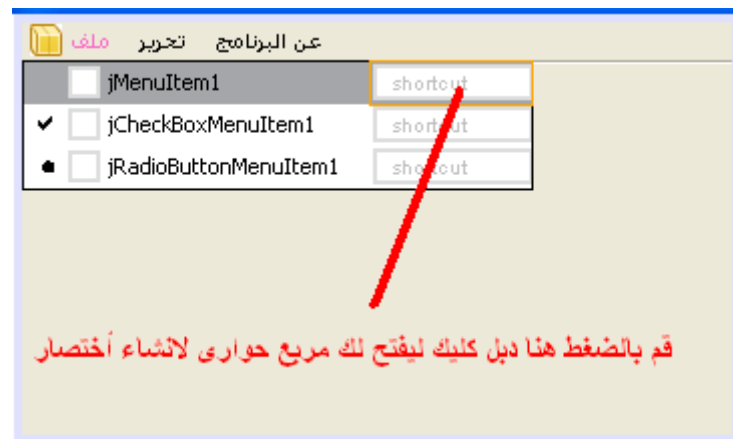
لك المربع الحوارى لاختيار الصورة كما فى الصورة التالية:



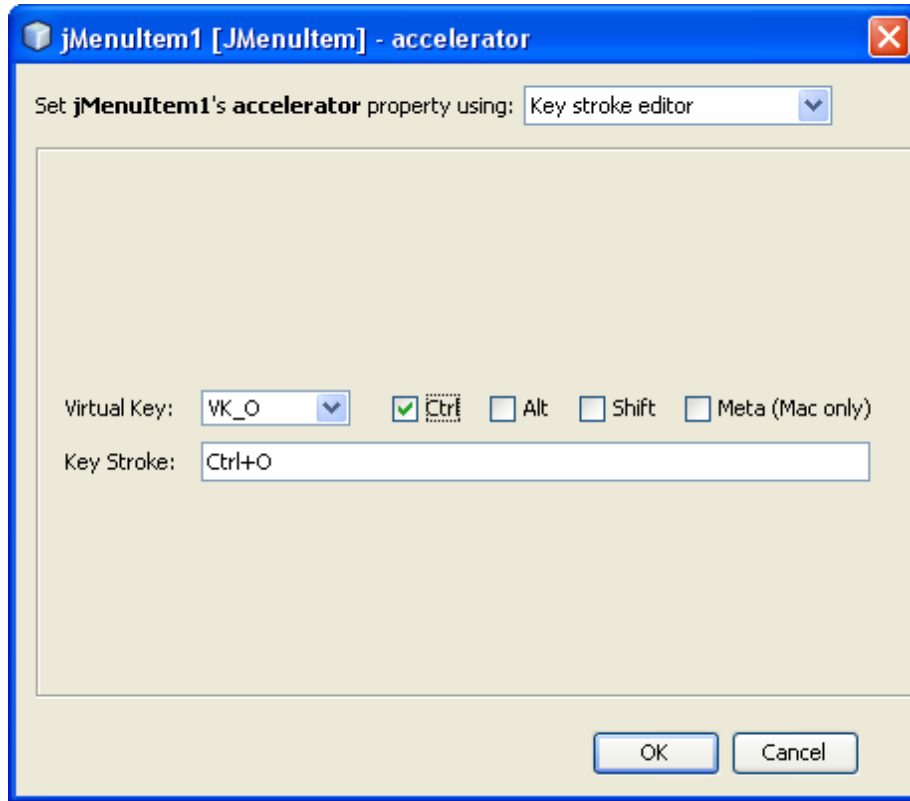
ب_ موضع تغيير اسم العنصر ... قم بالضغط عليه دبل كليك لتستطيع تغيير اسم العنصر كما فى الشكل التالى:



ج_ الموضع shortcut و الذى من خلاله تستطيع انشاء اختصار لهذا العنصر قم بالضغط كما فى الشكل التالى:



ليظهر لك المربع الحوارى التالى:



كما هو ظاهر اعلاه ... يمكنك تعيين مفاتيح الاختصار الذي تريد لعنصر القائمة من هذا المربع الحوار ... فمثلا اردنا ان يكون مفتاح الاختصار لهذا العنصر هو ... CTRL + O فقمنا بتنشيط خانة CTRL ثم اخترنا القيمة vk_o ثم اوكية ... كما هو ظاهر اعلاه.

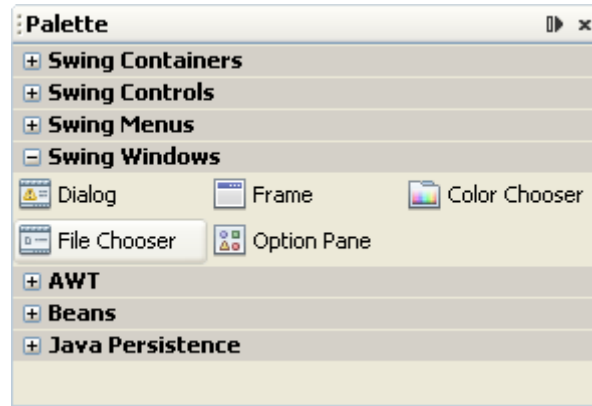
ويمكنك تنفيذ اي امر تحت اي عنصر من خلال Action Performed > Action > Events

الفصل السادس : التعامل مع المربع الحوارى JFileChooser

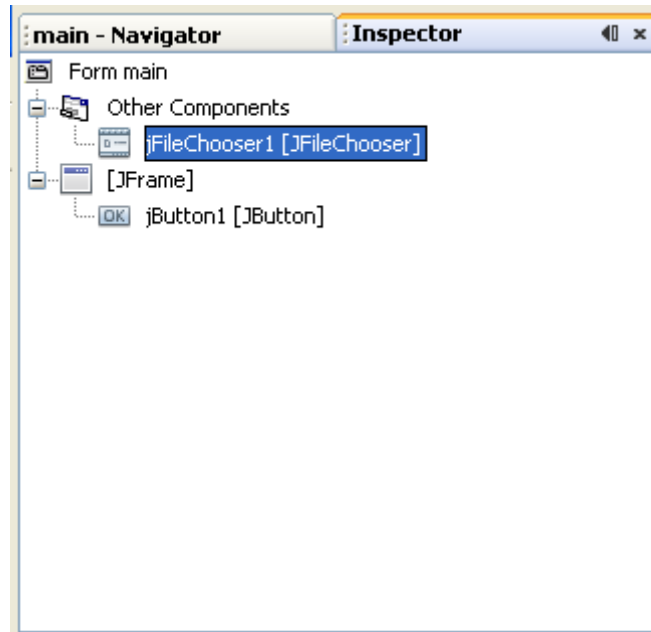
اليوم سنتعلم سويا الدرس الرابع عشر من دورة الجافا الرسومية و هو عن كيفية التعامل مع مربع الحوار File Chooser و الذى يتم استخدامة فى فتح وحفظ الملفات

كيفية استخدام مربع الحوار JFileChooser

المربع الحوارى JFileChooser هو أحد العناصر الرسومية التى توفرها الجافا التى تستخدم لاستعراض الملفات و المجلدات من على جهاز المستخدم ومن ثم فتح أو حفظ ملف من والى الجهاز أو حتى لإنشاء مجلد.....ويمكنك الحصول على هذا العنصر الرسومى من قائمة palette كما فى الشكل التالى:



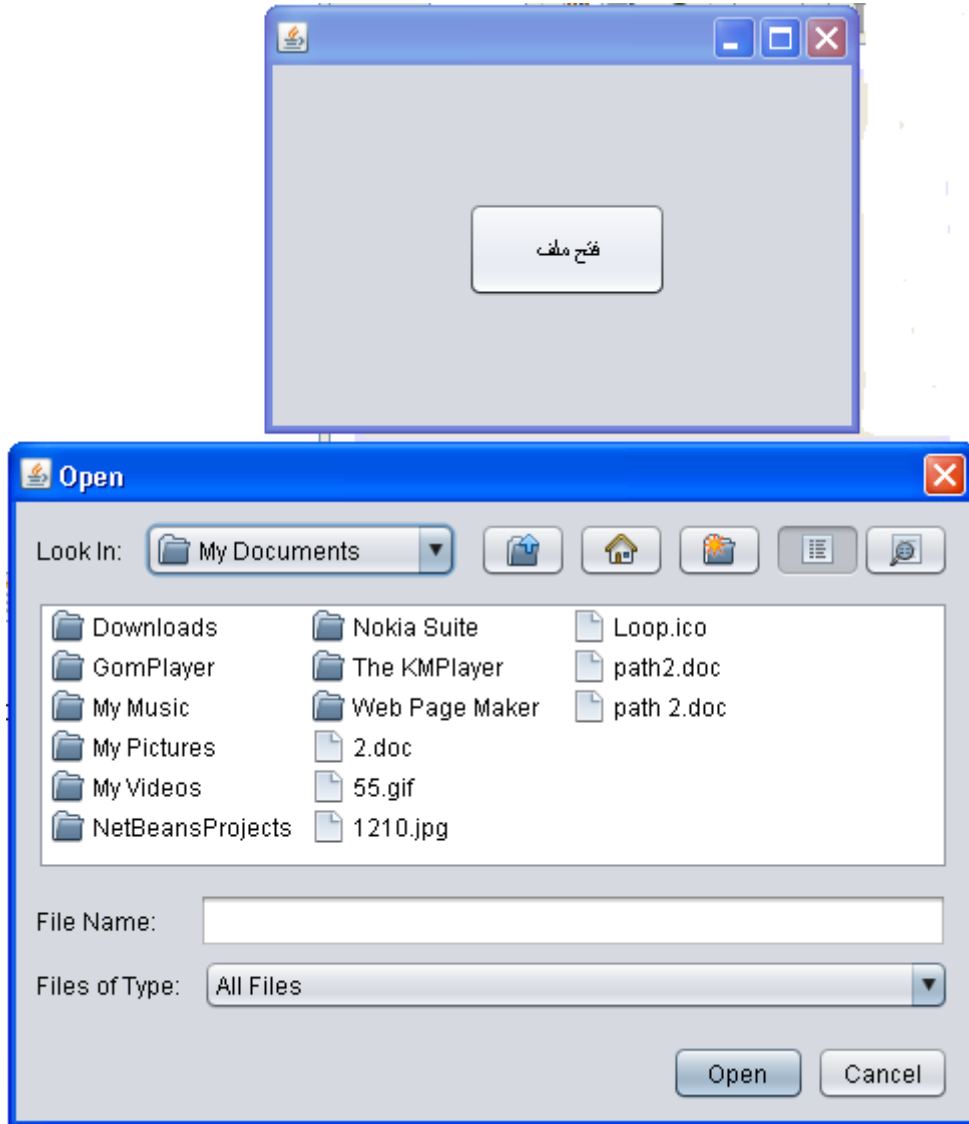
قم إسقاط العنصر الرسومى تحت نافذة مشروعك وليس على النافذة نفسها حتى لا يتم رسم العنصر عليها....أذن كل ما عليك فعله إسقاط العنصر المنقطة البياض اسفل النافذة و ستجده كما فى الشكل التالى...



و لكى تتمكن من اظهار المربع الحوارى نقوم بوض زر أمر على النافذة وفى حدث ActionPerformed نكت الكود التالى:

`jFileChooser1.showOpenDialog(this);`

وهذا هو الكود الخاص باظهار المربع الحوار (فتح) وعند تشغيل مشروعك و الضغط على زر الامر فتح سيظهر لك المربع الحوارى كما فى الشكل التالى:

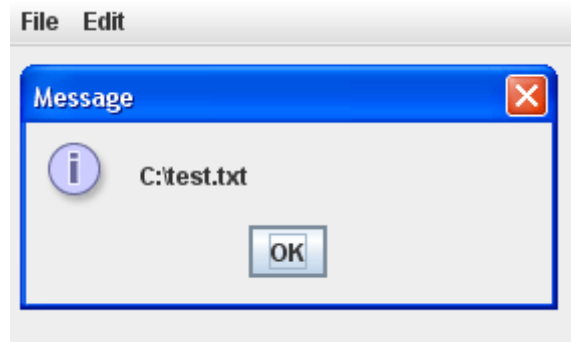


بهذا المربع الحوارى حتى الان لا نستطيع فتح أى ملف و وكل ما نستطيع فعله هو استعراض و التنقل بين الملفات و المجلدات ...اذن لكى نتمكن من فتح ملف معين من خلال هذا المربع الحوارى نستخدم الدالة `getSelectedFile` وذلك بتعديل الكود السابق ليصبح كالتالى:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
int egy =jFileChooser1.showOpenDialog(this);  
if (egy==javax.swing.JFileChooser.APPROVE_OPTION){  
java.io.File file =jFileChooser1.getSelectedFile();  
String file_name = file.toString( );  
JOptionPane.showMessageDialog(null, file_name);  
}  
}
```

- في السطر الثاني قمنا بتعريف متغير من نوع int يسمى egypt ثم جعلناه يحمل مربع الحوار (فتح)
- السطر الثالث : استخدمنا جملة شرطية لمعرفة ما اذا كان المستخدم قام بالضغط على زر (موافق) في المربع الحوارى (فتح) أم لا ... و استخدمنا لذلك APPROVE_OPTION و التى تعنى زر الموافق فى المربع الحوارى
- السطر الرابع ... اذا تحقق من الشرط يقوم بالحصول على اسم الملف المختار ... و استخدمنا لذلك الكائن file من الفئة io وبالتالي فإن الملف الذي اختاره المستخدم في نهاية المطاف يتم تخزينه في المتغير file
- السطر الخامس :قما بالاعلان عن المتغير file_name كمتغير نصى و بعد ذلك جعلناه يقوم بحفظ القيمة المخزنة في الكائن file بعد تحويله الى نص عن طريق الامر toString
- السطر السادس :ظهور رسالة بأسم الملف المختار

الآن قم بتشغيل البرنامج ثم انقر على زر الامر فتح ...لمشاهدة مربع الحوار (فتح). حدد أي ملف على جهاز الكمبيوتر الخاص بك وانقر فوق فتح...ستظهر لك رسالة بأسم الملف المختار كما في الشكل التالي:



تصفية الملفات المختارة

عند استخدامك للمربع الحوارى ستلاحظ أنه يتم تعيين أنواع الملفات إلى "جميع الملفات". ... و يمكنك تصفية الملفات في هذه القائمة، بحيث يمكن للمستخدم فتحها فقط، دون غيرهاكالملفات النصية، أو صور ذات امتدادات معينة (JPEG، GIF، PNG، نصوص، صور، ميديا .. الخ) و يتم ذلك باستخدام الطريقة addChoosableFileFilter بالإضافة الى الكائن FileFilter و الكائن FileNameExtensionFilter و ذلك على النحو التالى:

في البداية قم بإستيراد الكائنات كما يلي:

```
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
```

ولتصفية الملفات نحتاج الى إنشاء كائن FileFilter جديد و ذلك كما يلي:

```
FileFilter ft = new FileNameExtensionFilter("Text Files", "txt");
```

و كما تلاحظون أننا قمنا بوضع نوع الملفات المراد اظهارها و امتدادها بين قوسين ("Text Files", "txt") بعد ذلك نستخدم الدالة addChoosableFileFilter لتصفية الملفات كما يلي:

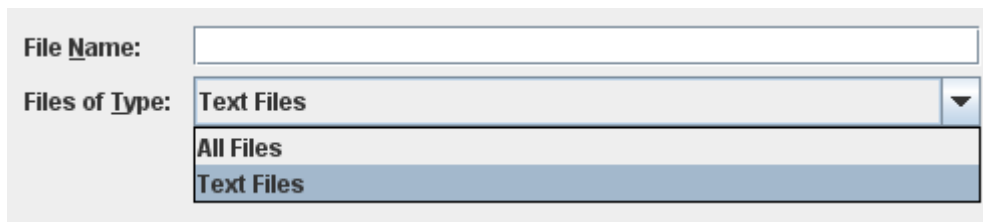
```
jFileChooser1.addChoosableFileFilter(ft);
```

و بالتالى فإن الكود الكامل سيصبح كما يلي:

الرمز:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    FileFilter ft = new FileNameExtensionFilter("Text Files", "txt");  
    jFileChooser1.addChoosableFileFilter(ft);  
  
    int egy =jFileChooser1.showOpenDialog(this);  
    if (egy==javax.swing.JFileChooser.APPROVE_OPTION){  
        java.io.File file =jFileChooser1.getSelectedFile();  
        String file_name = file.toString( );  
        JOptionPane.showMessageDialog(null, file_name);  
    }  
}
```

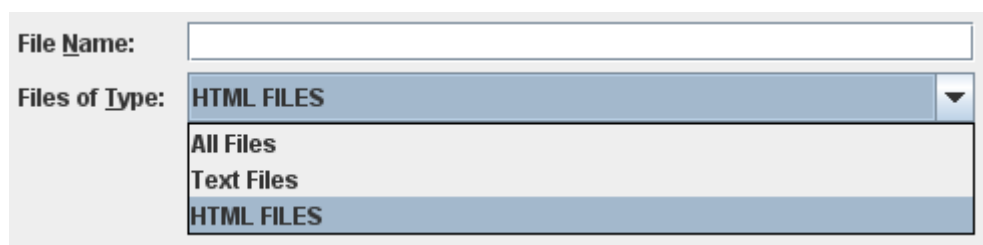
و النتيجة



و إذا كنت تريد إضافة نوع آخر من الملفات في القائمة، (لعرض ملفات HTML ، على سبيل المثال)، يمكنك إعداد كائن
آخر: FileFilter:

```
FileFilter ft = new FileNameExtensionFilter("Text Files",  
"txt");  
FileFilter ft2 = new FileNameExtensionFilter("HTML Files",  
"html");  
  
jFileChooser1.addChoosableFileFilter(ft);  
  
jFileChooser1.addChoosableFileFilter(ft2);
```

و النتيجة ايضا



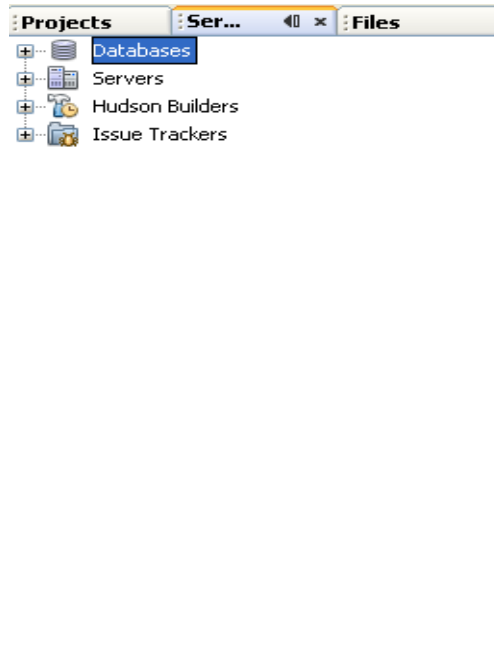
الباب الثالث قواعد البيانات

الفصل الاول : قواعد بيانات Java DB

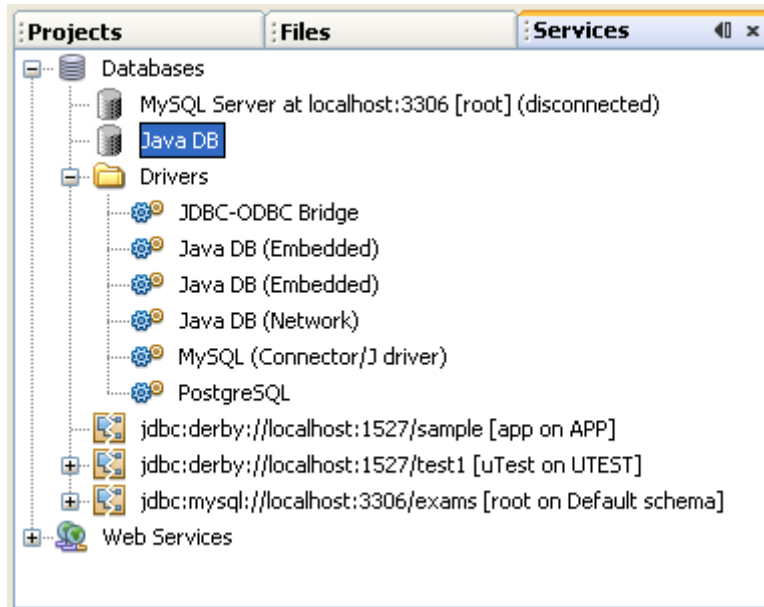
في هذا الدرس ان شاء الله تعالى سنتعلم سويا أولى حلقات التعامل مع قواعد البيانات في الجافا ... و درس اليوم سيكون عن كيفية انشاء قاعدة بيانات Java DB من خلال بنية **NetBeans** وهي قاعدة بيانات علائقية مفتوحة المصدر تم انشائها و ادماجها في الجافا و التي تعتمد على نسخة Apache Derby ليتم تشغيلها على سرفر تخيلي virtual serve و الذي يمكنك تشغيله و ايقافه من خلال NetBeans وكما سنوضح كيفية التعامل معها من إضافة و حذف و تعديل وخلافة وكل ذلك عن طريق المزود JDBC و الذي يتم استخدامه في الاتصال بالكثير من قواعد البيانات مثل (أوراكل و ماي سكو...الخ)...

أولا : إنشاء قاعدة بيانات Java DB من خلال NetBeans

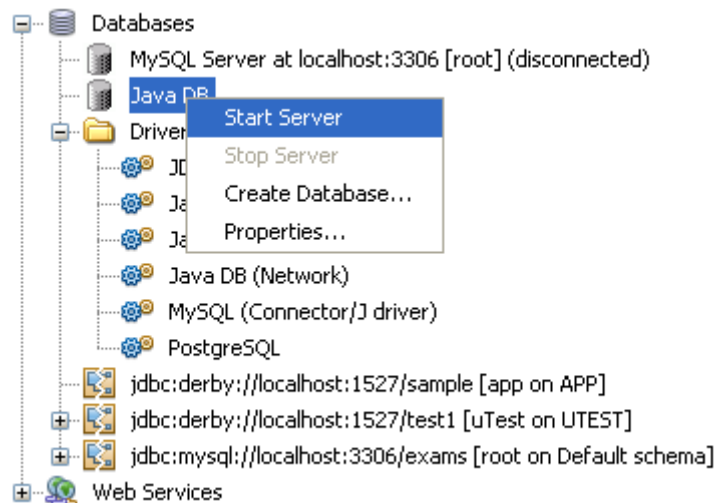
1_ قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services في القوائم الجانبية كما في الشكل التالي:



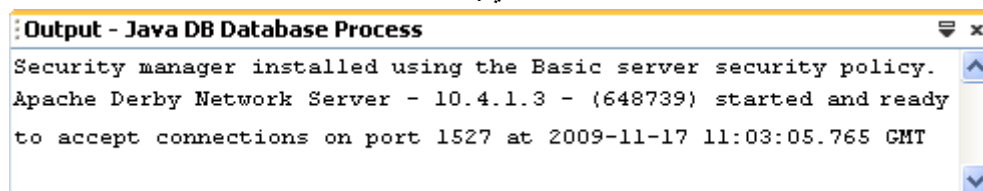
2_ قم بالضغط دبل كليك على ايقونة Databases أو بالضغط على علامة () التي بجوار الايقونة ليتم فتحها بالشكل التالي:



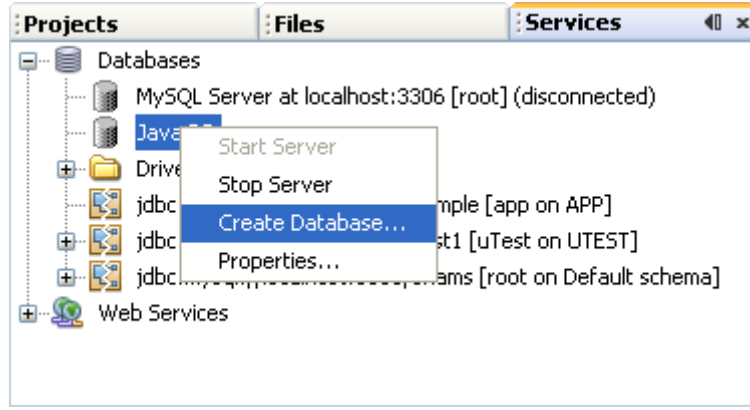
3_ قم بتشغيل الخادم Virtual Server و ذلك بتحديد Java DB ثم كليك يمين ثم أختار الامر Start Server كما في الشكل التالي:



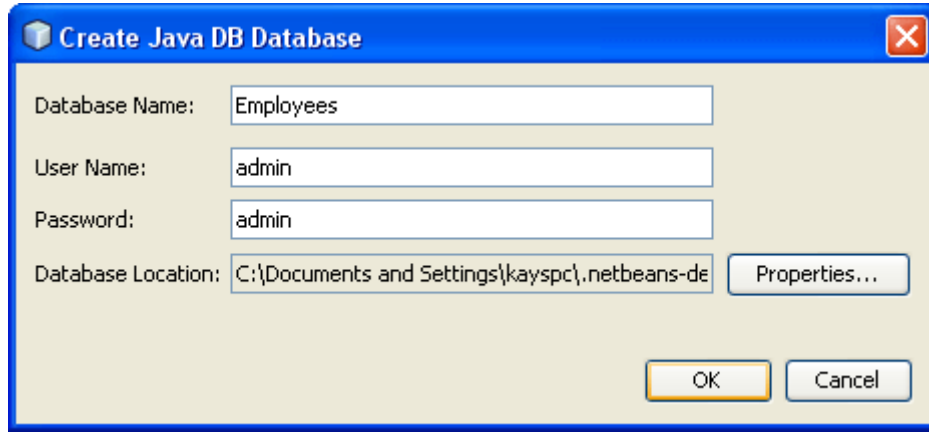
الآن انظر الى نافذة الاخراج في NetBeans و ستجد بعض الرسائل التي تخبرك بأنه تم تشغيل الخادم كما في الصورة التالية:



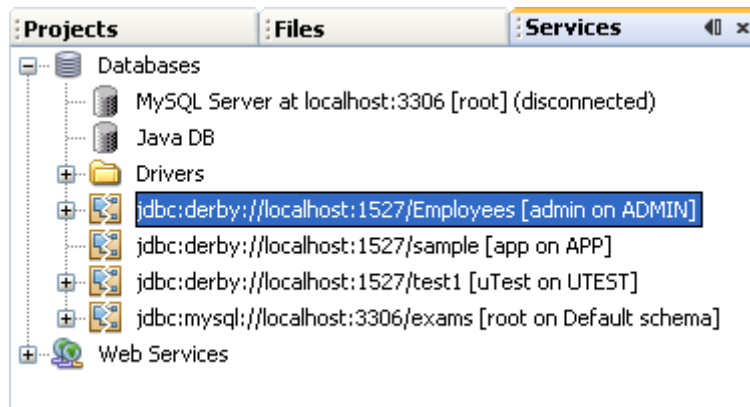
4_ إنشاء قاعدة بيانات Java DB وذلك من خلال تحديد ايقونة Java DB مرة أخرى ثم كليك يمين ثم أختار الامر Create Database كما في الصورة التالية:



5_ عند النقر على الامر Create Database كما هو ظاهر اعلاه سوف يظهر لنا معالج إنشاء قاعدة البيانات و التي يطلب منك بعض البيانات مثل اسم قاعدة البيانات ، اسم المستخدم ، وكلمة المرور ...فعلى سبيل المثال سوف نقوم بإنشاء قاعدة بيانات تسمى "Employees" كما في الصورة التالية:



بعد ما قمنا بإضافة كافة البيانات المطلوبة لقاعدة بيانات Employees نقوم بالضغط على زر (ok) لتظهر قاعدة البيانات في التويب Server كما في الشكل التالي:

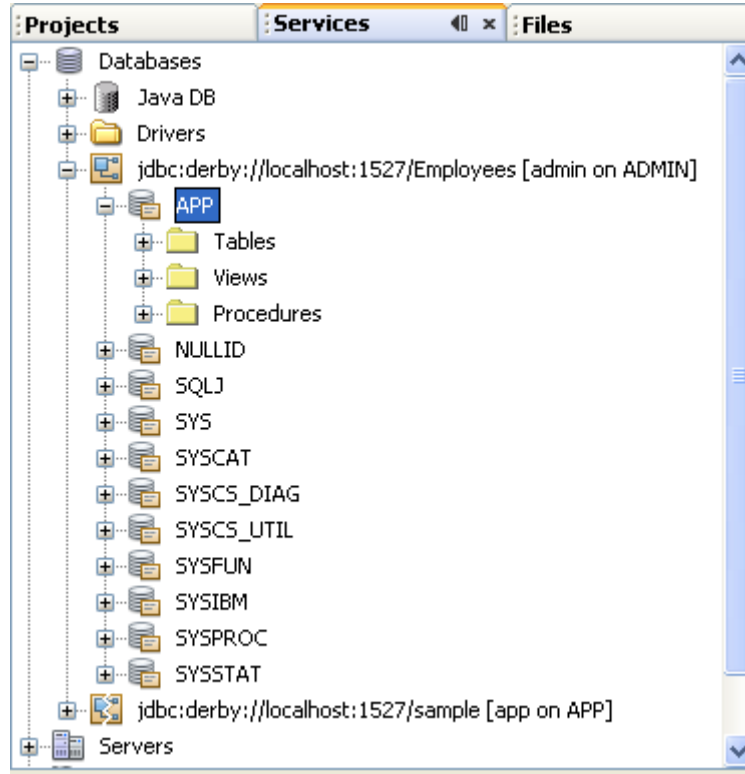


ثانيا : إنشاء جدول داخل قاعدة البيانات:

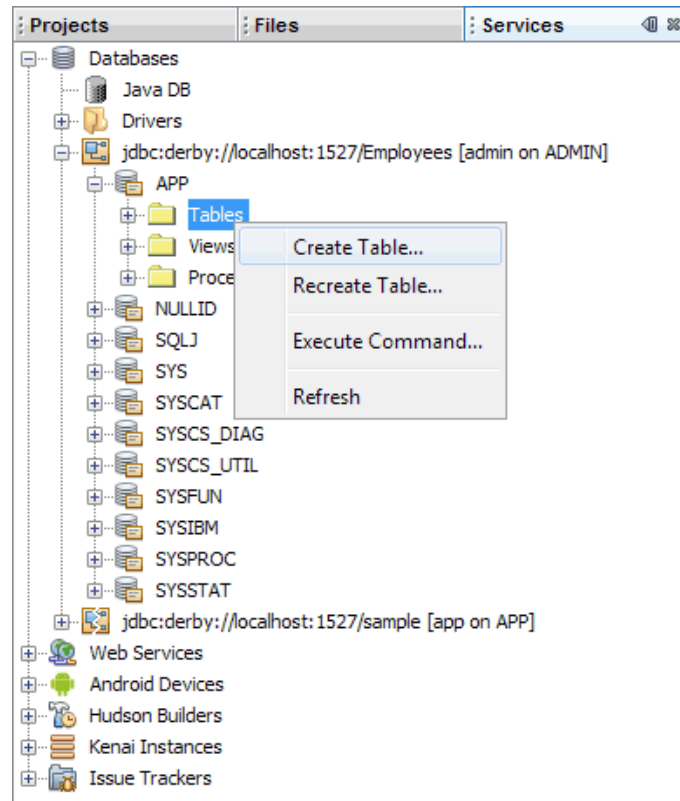
الآن بعد أن تم إنشاء قاعدة البيانات Employees ، تحتاج إلى إنشاء جدول في قاعدة البيانات. للقيام بذلك أتبع الخطوات التالية:

1_ انقر بزر الماوس الأيمن على قاعدة البيانات الخاصة بك. من القائمة التي تظهر حدد الامر Connect كما في الشكل التالي:

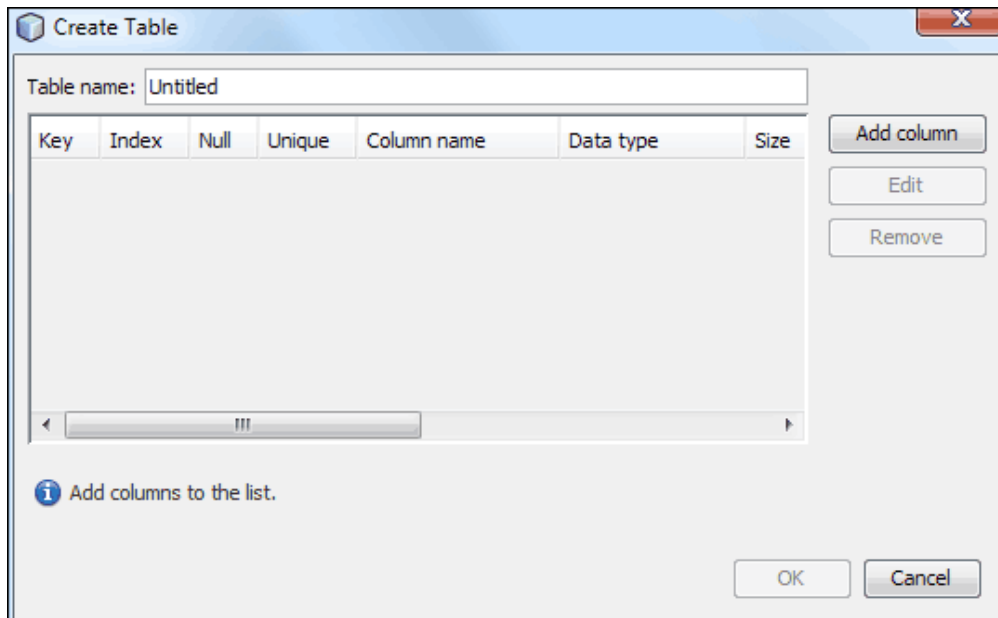
بعد ما يتم عملية الاتصال بالخادم ستجد انه قد تم انشاء ثلاث مجلدات وهي Tables , Views , Procedures و حتى تتمكن من رؤية تلك المجلدات في التويب Server قم بالنقر دبل كليك على قاعدة البيانات التي انشأناها Employees لتفتح لك بعض المجلدات ثم قم بالنقر دبل كليك على المجلد APP لترى المجلدات الثلاث كما في الشكل التالي:



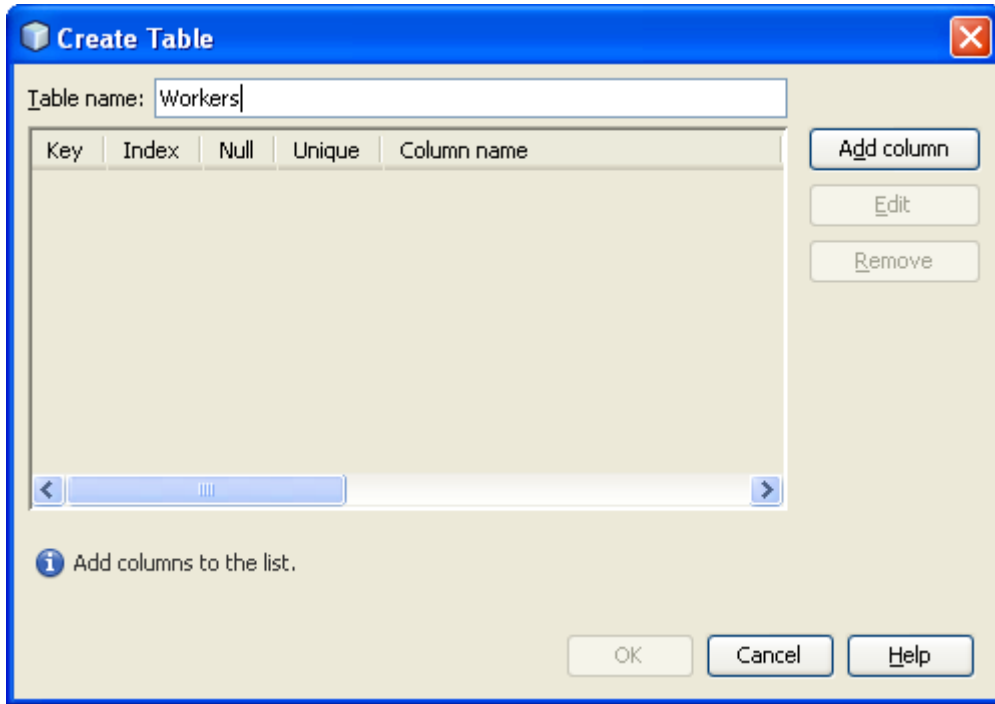
2_ قم بالنقر شمال على المجلد Tables و من القائمة التي تظهر، حدد الامر Create Table كما في الصورة التالية :



3_ عند النقر كما تم شرحه أعلاه يظهر لك المربع الحوارى التالى:

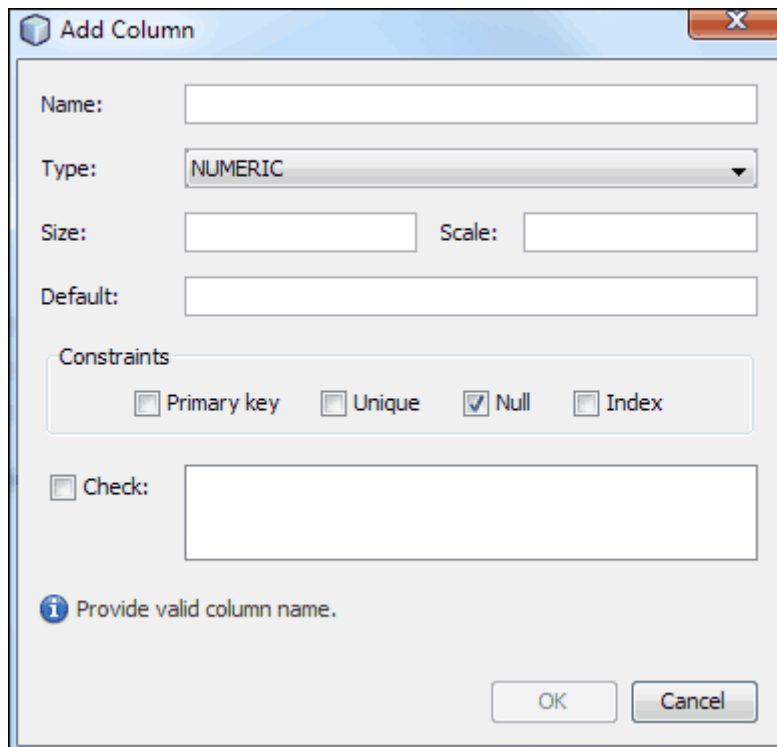


في مربع النص Table Name في الأعلى، أ حذف الاسم الافتراضي..Untitled ثم اكتب اسما جديدا للجدول و ليكن مثلا...Workers و هكذا سيكون لديك جدول يسمى Workers والذي هو في قاعدة البيانات الموظفين كما في الصورة التالية.....

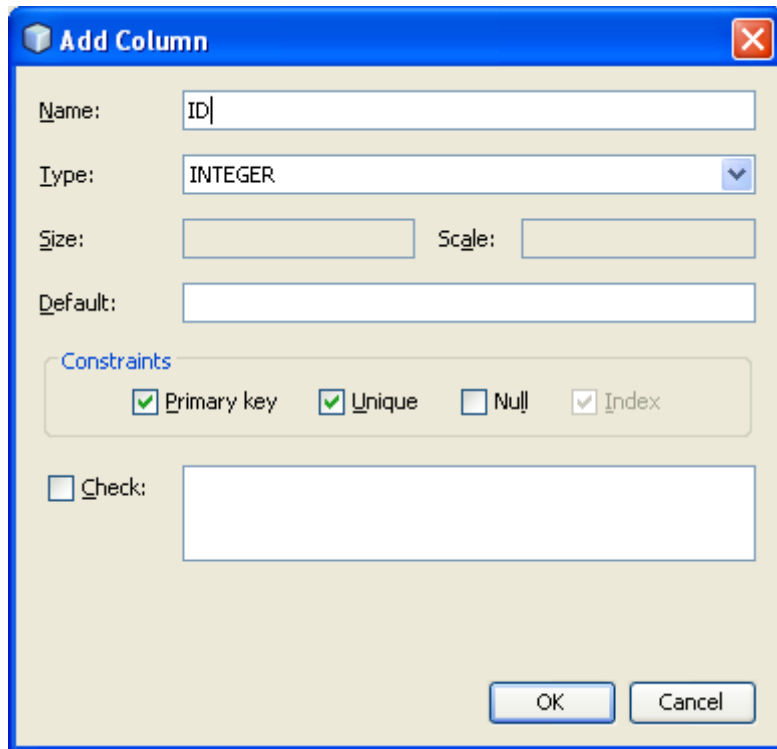


ولكن لا يمكنك انقر فوق موافق حتى الآن لإنشاء هذا الجدول الابعاد إضافة حقول لهذا الجدول..

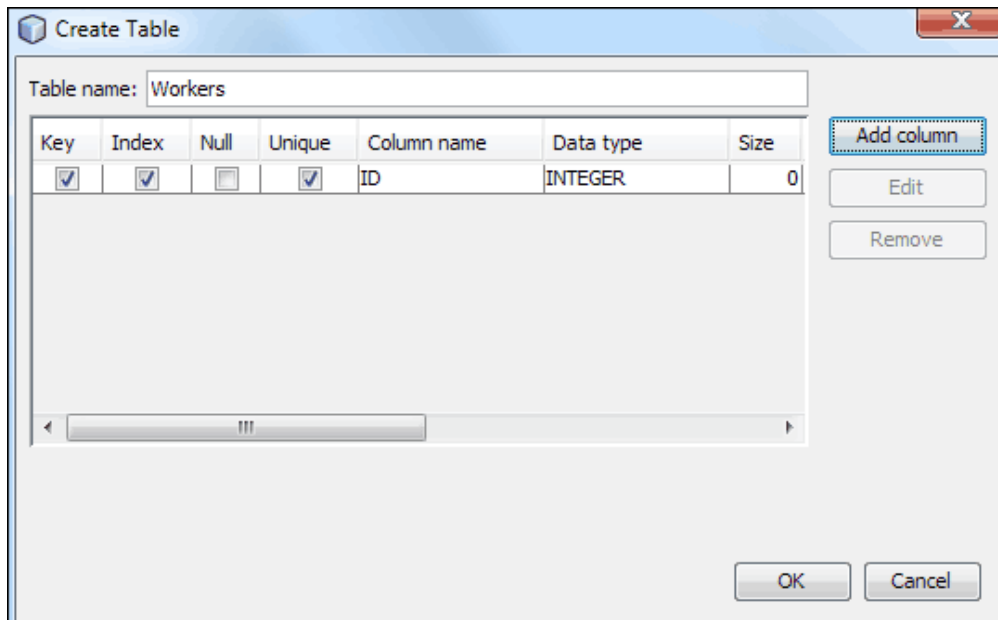
4_إضافة حقول للجدول و يتم ذلك عن طريق النقر فوق زر الامر Add Column فى المربع الحوارى أعلاه ليظهر لك المربع الحوارى التالى:



من المربع الحوارى السابق تستطيع إنشاء الحقول أو الاعمدة التى تريدها فى جدول البيانات... وفى مثالنا هذا نريد إنشاء عدد ٤ حقول او اعمدة لهذا الجدول وهم (رقم العامل / الاسم الاول / الاسم الثانى / الوظيفة...لذلك سنقوم بإنشاء أول حقل وهو رقم الموظف(ID) كما فى الشكل التالى:



ومن الملاحظ في المربع الحوارى إعلاه انه يطلب منك فى مربع النص name أن تدخل أسم مناسب للحقل أو العمود... وفى مربع النص TYPE و التى تحدد نوع البيانات المراد تخزينها فى هذا الحقل... وهذا اهم المتطلبات و الباقي يتم وضع قيم افتراضية... وعند النقر على زر OK سيظهر الحقل كما فى الشكل التالى:



سنكرر العملية لباقي الحقول مع ملاحظة التنسيق التالى:

Column Name: First_Name
Data Type: VARCHAR
Size: 20

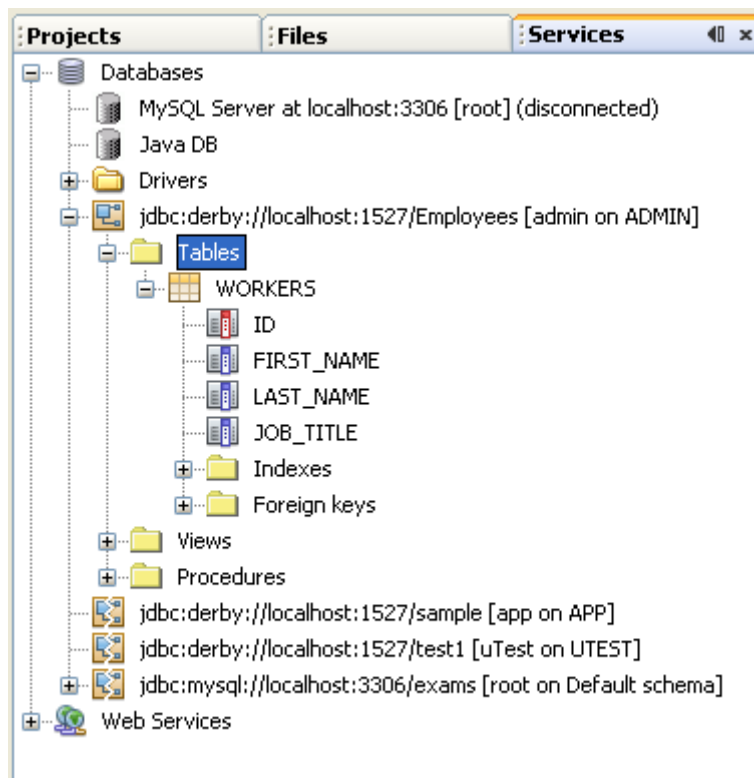
Column Name: Last_Name
Data Type: VARCHAR
Size: 20

Column Name: Job_Title
Data Type: VARCHAR
Size: 40

حتى تتم كما في الشكل التالي:

Key	Index	Null	Unique	Column name	Data type	Size	Scale
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ID	INTEGER	0	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	First_Name	VARCHAR	20	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Last_Name	VARCHAR	20	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Job_Title	VARCHAR	40	0

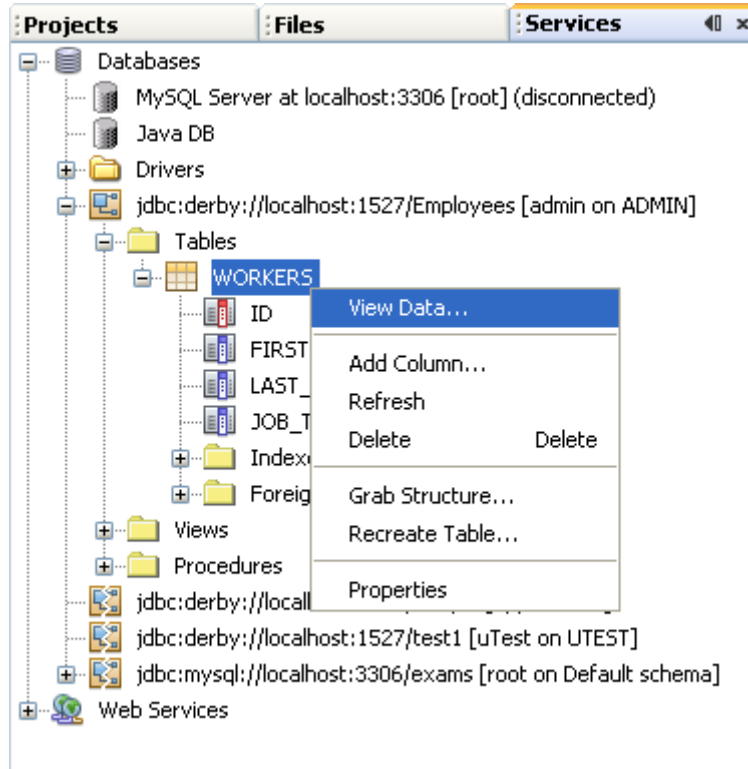
5_ انقر الان فوق موافق بعد ما قد أدخلت كل المعلومات وحقول الجدول المراد إنشاؤه ليظهر كما في الشكل التالي:



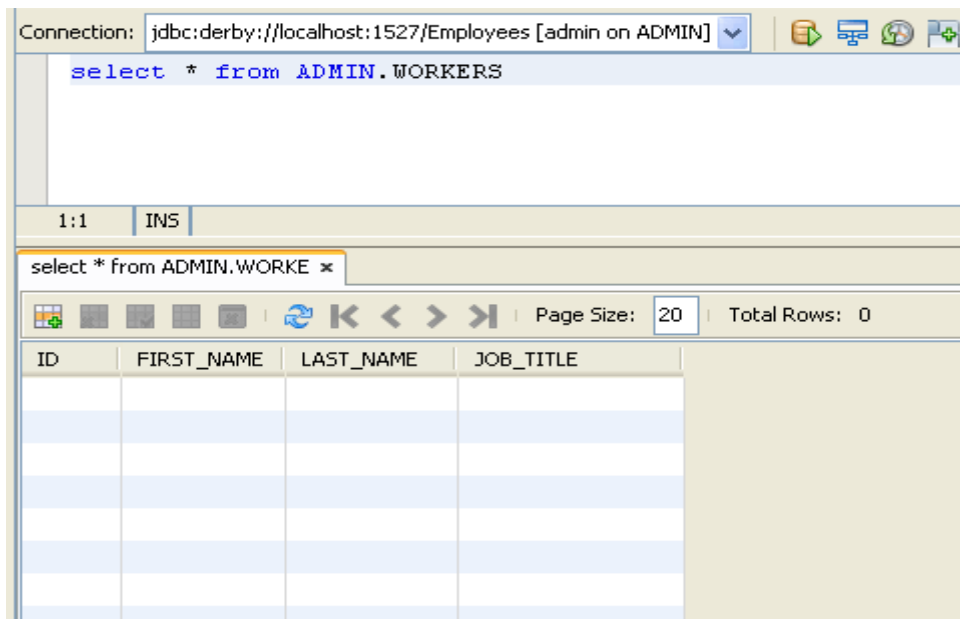
ثالثا : إدخال بيانات الى جدول قاعدة البيانات يدويا:

بعد ما تم إنشاء جدول قاعدة البيانات و حقوله تعالوا نتعلم كيفية ادخال قيم أو بيانات الى هذا الجدول يدويا كأن نفتح قاعدة البيانات و ادراج القيم بها...

1_قم بالنقر بزر الماوس الأيمن على اسم الجدول المراد وضع قيم به و فى مثالنا هذا Workers. و من القائمة التي تظهر، حدد الامر View Data كما فى الشكل التالى:



2_بعد ذلك سيظهر لك نافذة جديدة تظهر في إطار بنية NetBeans الرئيسية كما فى الشكل التالى:



و لإضافة قيم للحقول، انقر فوق رمز علامة () الجمع الخضراء في النصف السفلي من النافذة كما في الشكل التالي

ID	FIRST_NAME	LAST_NAME	JOB_TITLE

ليظهر لك المربع الحوارى التالى و الذى يمكنك إضافة البيانات التى ترغب فيها كما فى الشكل التالى:

Press CTRL+Tab to exit data entry mode from the table. Press CTRL+0 to set NULL value and CTRL+1 to set DEFAULT value for a given column.

#	ID	FIRST_NAME	LAST_NAME	JOB_TITLE
1		أحمد 1	مناع	محاسب
2		خالد 2	كامل	محاسب
3		محمد 3	يوسف	حرامى

إضافة أكثر من حقل

Show SQL Add Row Remove OK Cancel

رابعا : الاتصال بقاعدة البيانات

بعد ما تعلمنا سويا كيفية إنشاء قاعدة بيانات Java DB و التى قد أعطينا لها أسم **Employees** و انشأنا بها جدول **Workers** و أدخلنا به بعض البيانات يدويا ... فى تلك الجزئية سنتعلم أيضا كيفية الاتصال بقاعدة البيانات تلك و جداولها و التعامل عليها بالاضافة و الحذف و التعديل ... الخ...

1_ قم بإنشاء مشروع جديد و ليكن database_console والدالة الرئيسية مثلا: DBConnect كما فى الشكل التالى:

Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

```

package database_console;

public class DBConnect {

    public static void main(String[] args) {

    }

}

```

2_ للاتصال قاعدة بيانات نحتاج إلى كائنات الاتصال وهي `sql.Connection`, `sql.DriverManager`, `sql.SQLException` ويتم أستيراد تلك الكائنات عن طريق الكود التالي:

الرمز:

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

```

3_ نص الاتصال بقاعدة البيانات يكون عن طريق الكود التالي:

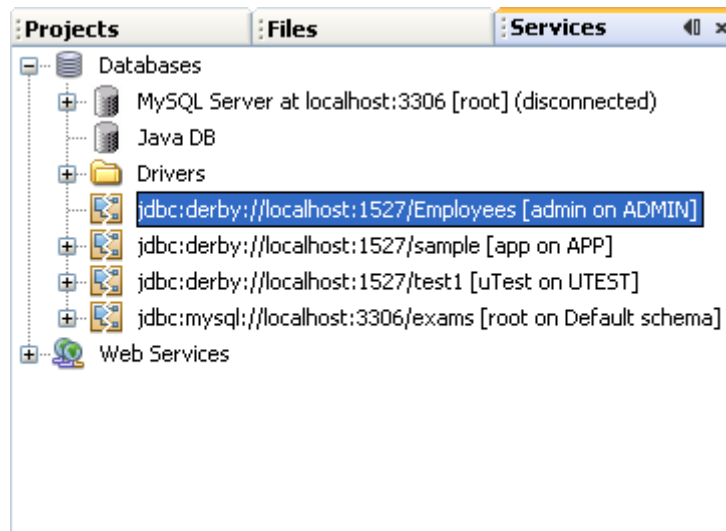
الرمز:

```

Connection con = DriverManager.getConnection( host, username, password );

```

وكما تلاحظون في الكود أعلاه أننا انشأنا اتصال يسمى `con` وذلك عن طريقة `DriverManager` التي به طريقة `getConnection`. وهو يحتاج الى اسم المضيف (الذي هو موقع قاعدة البيانات الخاصة بنا)، واسم المستخدم، وكلمة مرور. حتى يتم الاتصال بنجاح...ويمكنك الحصول عنوان موقع قاعدة البيانات كما في الصورة التالية:



وكما تلاحظون من الصورة فإن عنوان قاعدة البيانات المبينة أعلاه هو:

jdbc:derby://localhost:1527/Employees

لذلك سنقوم بالاعلان عن متغير نصي ليتم تخزين فيه موقع قاعدة البيانات تلك و ذلك بأستخدام الكود التالي:

الرمز:

```
String host = "jdbc:derby://localhost:1527/Employees";
```

كما اننا سنقوم بتعريف متغيرين آخرين الاول ليتم تخزين فيه أسم المستخدم و الثانى ليتم تخزين فيه رقم المرور و ذلك على النحو التالي:

```
String uName = "admin";
String uPass= " admin ";
```

وبالتالى فإن الكود الكامل سيكون على النحو التالى

```
package database_console;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect {

    public static void main(String[] args) {

        String host = "jdbc:derby://localhost:1527/Employees";
        String uName = "admin";
        String uPass = "admin";
        Connection con = DriverManager.getConnection(host, uName, uPass);

    }
}
```


كما ترون في الصورة أعلاه، ستجدون خط أحمر مائل في الكود يدل على وجود خطأ فيه... والسبب في ذلك هو خطأ SQLException والذي ربما يحصل نتيجة عدم وجود المضيف مثلاً لذلك لابد من استخدام التعليمة try ... catch كما في الشكل التالي:

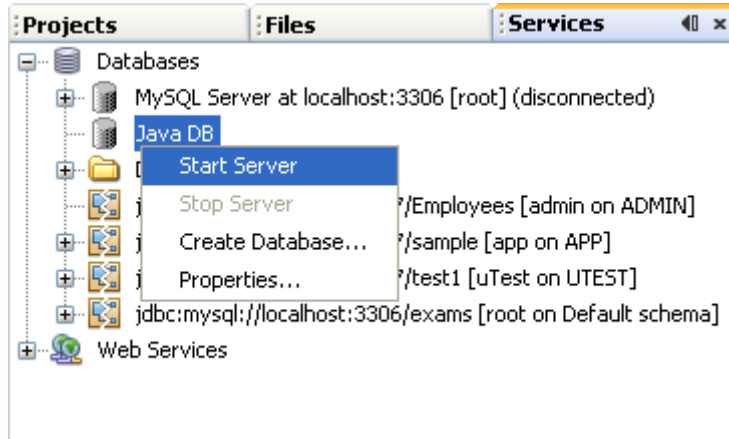
```
public static void main(String[] args) {

    try {
        String host = "jdbc:derby://localhost:1527/Employees";
        String uName = "admin";
        String uPass = "admin";
        Connection con = DriverManager.getConnection(host, uName, uPass);
    }
    catch ( SQLException err ) {
        System.out.println( err.getMessage( ) );
    }
}
```

الآن قم بتشغيل برنامج و لاحظ ماذا سيحدث؟.....ربما تجد رسائل الأخطاء التالية:

"java.net.ConnectException : Error connecting to server localhost on port 1527 with message Connection refused: connect."

و يحدث هذا الخطأ نتيجة أن السرفر متوقف و الحل هو تشغيله كما في الصورة التالية :

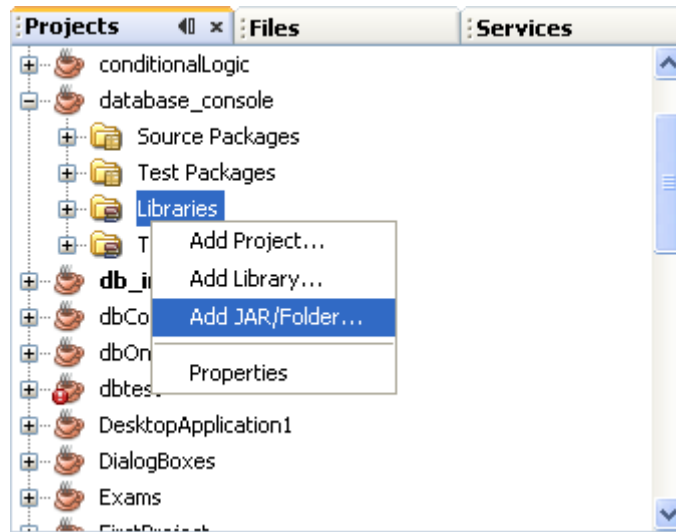


الخطأ الثاني الذي سيحدث هو ما يلي:

"No suitable driver found for jdbc:derby://localhost:1527/Employees"

وهذا الخطأ ناتج بسبب أن DriverManager يحتاج إلى ملف يسمى derbyclient لنجاح الاتصال بقاعدة البيانات.....ولكي تتمكن من إضافة هذا الملف لمشروعك قم بعمل الآتي:

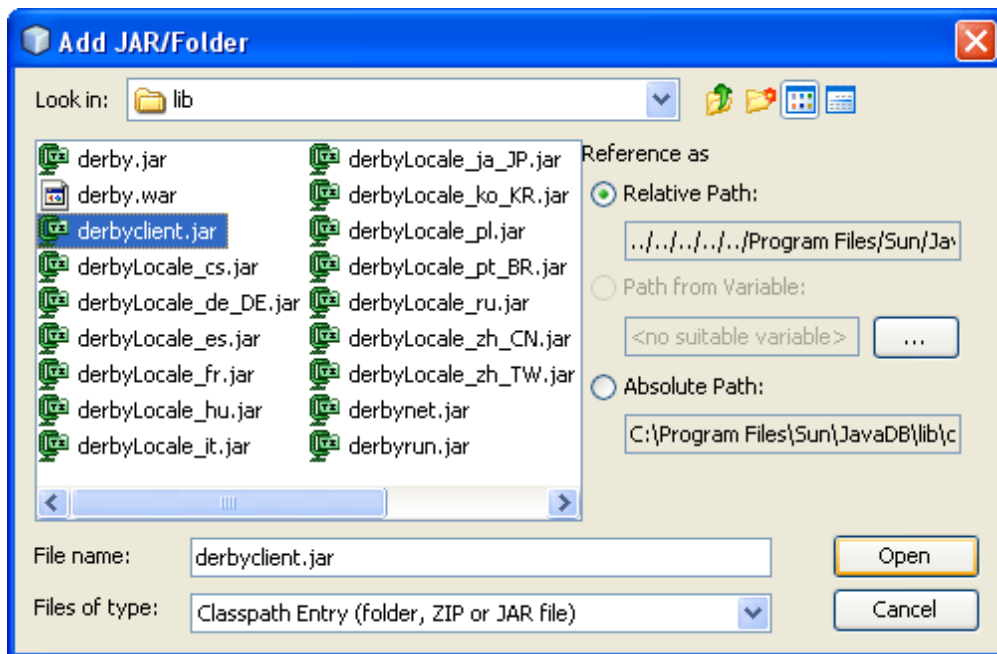
1_ قم بالنقر يمين على المجلد Libraries ومن القائمة التي ستظهر اختر الأمر **Add Jar/Folder** كما في الصورة التالية:



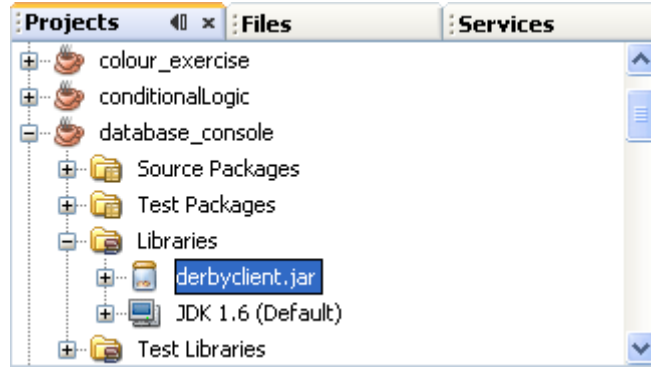
2_ سيظهر لك مربع حوارى لاضافة ملف من نوع jar قم بالبحث عن الملف ... derbyclient.jar بالنسبة لى كان على المسار التالى:

C:\Program Files\Java\jdk1.7.0_01\db\lib

وقمت بإضافة للمشروع كما فى الصورة التالية:



انقر فوق فتح، وسيتم إضافة الملف إلى مكتبة المشروع الخاص بك كما فى الشكل التالى:



الآن يمكنك إعادة تشغيل البرنامج ولن تجد أي رسائل أخطاء أخرى معنى ذلك أنه قد تم الاتصال بقاعدة البيانات بنجاح

خامسا : قراءة محتويات جدول البيانات

بعد ما قمنا بالاتصال بقاعدة البيانات، فإن الخطوة التالية هي للوصول إلى الجدول في قاعدة البيانات وهو في مثالنا هذا Workers... و لكي يتم الاتصال بهذا الجدول نحتاج إلى تنفيذ عبارة SQL ، ومن ثم معالجة كافة الصفوف والأعمدة التي تم إرجاعها....وذلك على النحو التالي:

1_ قم باستيراد الكائن sql.Statement والكائن sql.ResultSet والكائن sql.ResultSet وذلك عن طريق الكود التالي:

```
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
```

2_ في منطقة معالجة الاستثناءات try catch قم بوضع الكود التالي:

```
Statement stmt = con.createStatement( );
```

هنا قمنا بإنشاء الكائن Statement بأسم stmt

3_ تحت الكود السابق قم بإنشاء جملة استعلام للمرور بجدول البيانات Workers وذلك عن طريق الكود التالي:

```
String SQL = "select * from APP.WORKERS";
```

وجملة الاستعلام السابقة تقوم بتحديد كافة السجلات في الجدول Workers

4_ نقوم بإنشاء مؤشر للمرور على قاعدة البيانات و معالجة بياناتها وليكن rs ثم تميز جملة الاستعلام السابقة و ذلك عن طريق الكود التالي:

```
ResultSet rs = stmt.executeQuery( SQL );
```

وكما تلاحظون في الكود أعلاه أننا قد أنشأنا المؤشر rs وهو الوسيلة لتخزين ومعالجة السجلات التي يتم إرجاعها من استعلام SQL و يأخذ ثلاث أنواع هما:

_TYPE_FORWARD_ONLY وهذا النوع يقوم بالمرور على السجلات للامام فقط من البداية الى النهاية.
_TYPE_SCROLL_SENSITIVE وهذا النوع يقوم بالمرور على السجلات للإمام وللخلف مع الكشف عن أي تغييرات يتم إجراؤها على السجلات.

`_TYPE_SCROLL_INSENSITIVE` وهذا النوع يقوم بالمرور على السجلات للإمام و للخلف مع عدم الكشف عن أي تغييرات يتم إجراؤها على السجلات.

ولكى نستطيع ان تحدد نوع المؤشر الخاص بلك اذهب الى الاقواس المستديرة في جملة التالية:

```
Statement stmt = con.createStatement( );
```

و بداخل تلك الاقواس المستديرة يمكنك وضع نوع المؤشر على النحو التالي:

```
Statement stmt = con.createStatement( ResultSet.TYPE_SCROLL_SENSITIVE );
```

وذلك لو تركنا تلك الاقواس فارغة بدون تحديد المؤشر فإن البرنامج سوف يعتمد على المؤشر الافتراضى للسجلات وهو في هذه الحالة `TYPE_FORWARD_ONLY`.... ومع ذلك لم ننتهي بعد من الكود السابق و ذلك لكوننا في حالة استخدامنا النوعين الاخرين من المؤشرات فيجب تحديد ما إذا كان هذا المؤشر للقراءة فقط ام قابل للتحديث و ذلك باستخدام الثوابت التالية

```
ResultSet. CONCUR_READ_ONLY  
ResultSet. CONCUR_UPDATABLE
```

و بالتالى فإن الكود النهائى سيصبح كما يلى:

```
Statement stmt = con.createStatement( ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);
```

لكن فى مثالنا هذا لن نقوم الان بالاضافة و التعديل و خلافة .. و بالتالى سنكتفى بالمؤشر الافتراضى `TYPE_FORWARD_ONLY`

4_إسناد حقول جدول لمتغيرات Workers حتى يستنى لنا طباعتها و ذلك عن طريقة حلقة تكرارية كما فى الكود التالى :

```
while ( rs.next( ) ) {  
int id_col = rs.getInt("ID");  
String first_name = rs.getString("First_Name");  
String last_name = rs.getString("Last_Name");  
String job = rs.getString("Job_Title");  
String p= id_col + " " + first_name + " " + last_name + " " + job ;  
System.out.println(p);  
}
```

ويمكنك ملاحظة فى الكود أعلاه أننا قد استخدمنا حلقة تكرارية للمرور على جميع سجلات الجدول ...واننا قد قمنا بالاعلان عن متغيرات لكل عمود فى الجدول على حسب نوع البيانات لكل عمود و هذا هو الكود بالكامل:

```
package database_console;
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.sql.ResultSet;
```

```

public class DBConnect {

    public static void main(String[] args) {
        try {
            String host = "jdbc:derby://localhost:1527/Employees";
            String uName = "admin";
            String uPass= "admin";

            Connection con = DriverManager.getConnection( host, uName, uPass);
            Statement stmt = con.createStatement( );
            String SQL = "select * from APP.WORKERS";
            ResultSet rs = stmt.executeQuery( SQL );

            while ( rs.next( ) ) {
                int id_col = rs.getInt("ID");
                String first_name = rs.getString("First_Name");
                String last_name = rs.getString("Last_Name");
                String job = rs.getString("Job_Title");
                String p= id_col + " " + first_name + " " + last_name + " " + job ;
                System.out.println(p);
            }
        }
        catch(SQLException e){

            System.out.println(e.getMessage());
        } } }

```

مثال للاتصال بقاعدة البيانات في الواجهة الرسومية

أهلا و سهلا إخواني الكرام في الجزء الاخير من هذا الدرس وهو عن كيفية الاتصال بقاعدة البيانات في الواجهة الرسومية .. تلك الجزئية سهلة بإذن الله تعالى طالما فهمت الاجزاء الاولى لهذا الدرس فهي نفس الاكواد البرمجية مع تعديل بسيط و ذلك على النحو التالي:

1_قم إنشاء نافذة رسومية تحتوي على أزرار وحقول نص كما في الشكل التالي:

The screenshot shows a Java Swing window with a light beige background. It contains four text input fields, each labeled with 'jTextField1' and a corresponding label to its right: 'رقم الموظف', 'الاسم الاول', 'الاسم الاخير', and 'الوظيفة'. Below the text fields, there are two rows of buttons. The first row contains four buttons labeled 'الاول', 'التالى', 'السابق', and 'الاخير'. The second row contains three buttons labeled 'جديد', 'حفظ', and 'حذف'.

2_ قم بحذف النص الافتراضي للحقول النص وترك كل منهم فارغ...وقم ايضا بتغيير أسماء المتغيرات الافتراضية للحقول نص على ما يلي:

```
textID  
textFirstName  
textLastName  
textJobTitle
```

وطبعا تلك الاسماء سوف يتم التعامل عليها في صفحة الكود عند كتابته....

3_ الان سنقوم باستيراد الكائنات اللازمة و التي قد تعرفنا عليها أعلاه للاتصال بقاعدة البيانات Employees و جدولها Workers وذلك عن طريق كود واحد فقط وهو التالي:

```
import java.sql.*;
```

بهذا الكود البسيط قمنا باستيراد جميع الكائنات الموجودة بالفئة sql فبدلا من اتسيراد كل كائن على حده كما فعلنا إعلاه أختصرنا تلك الاكواد بهذا الكود البسيط فالرمز النجمة (*) تعنى جميع كائنات الفئة..

4_ نقوم باستيراد الكائن JOptionPane و ذلك لعرض رسائل الاخطاء التي قد تحدث بالشكل الرسومي المتعارف عليه بدلا من System.out.println و يتم الاستيراد باستخدام الكود التالي:

```
import javax.swing.JOptionPane;
```

5_ نقوم بتعريف متغيرات الاتصال بقاعدة البيانات التالية:

```
Connection con;  
Statement stmt;  
ResultSet rs;
```

ويمكنك ملاحظة أننا قمنا بتعريف تلك المتغيرات على مستوى الفئة حتى نتمكن من استخدامها في أى حدث من الاحداث داخل الفئة

6_ نقوم بإنشاء طريقة و ليكن مثلا DoConnect و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجدولها.... و يتم إنشاء الطريقة باستخدام الكود التالي:

```
public void DoConnect() {  
  
  
  
}
```

6_ بداخل الطريقة سنقوم بوضع الاكواد اللازمة للاتصال بقاعدة البيانات و جدولها كما يلي:

_كود الاتصال بقاعدة البيانات:

```
String host = "jdbc:derby://localhost:1527/Employees";  
String uName = "admin";  
String uPass= "admin";  
  
con = DriverManager.getConnection( host, uName, uPass);
```

-كود الاتصال بجدول البيانات:

```

stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE );
String SQL = "select * from APP.WORKERS";
rs = stmt.executeQuery( SQL );

```

ستلاحظ أننا قد إضافنا بعد التغيرات في الكود السابق للاتصال بجدول البيانات عن الكود إعلاه في الجزئية السابقة... فقد حدد نوع المؤشر و أنه لتحديث البيانات و ذلك لإننا سوف نتعامل مع قاعدة البيانات بالاضافة و الحذف و التعديل على بياناتها

_كود إسناد حقول الجدول الى مربعات النصوص في النافذة:

```

rs.next( );
int id_col = rs.getInt("ID");
String id =Integer.toString(id_col);
String first_name = rs.getString("First_Name");
String last_name = rs.getString("Last_Name");
String job = rs.getString("Job_Title");

textID.setText(id);
textFirstName.setText(first_name);
textLastName.setText(last_name);
textJobTitle.setText(job);

```

و بالنظرة السريعة على الكود سنجد أننا قمنا بالاعلان عن المتغير الرقمي الصحيح id_col و ذلك لتخزين قيمة الحقل (ID) الموجود في جدول البيانات الذي نوعه ايضا رقمي....(integer) لكن المشكلة ان حقول النص في الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصي (id) لتخزين فيه قيمة المتغير id_col بعد تحويله الى متغير نصي باستخدام الامر Integer.toString

_وضع كود معالجة الاستثناءات try.... catch و بالتالى فإن الكو الكامل سيصبح كما يلي:

```

public void DoConnect( ) {
try {
String host = "jdbc:derby://localhost:1527/Employees";
String uName = "admin";
String uPass= "admin";
Connection con = DriverManager.getConnection( host, uName, uPass);
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE );
String SQL = "select * from APP.WORKERS";
ResultSet rs = stmt.executeQuery( SQL );

rs.next( );
int id_col = rs.getInt("ID");
String id =Integer.toString(id_col);
String first_name = rs.getString("First_Name");
String last_name = rs.getString("Last_Name");
String job = rs.getString("Job_Title");

textID.setText(id);
textFirstName.setText(first_name);
textLastName.setText(last_name);
textJobTitle.setText(job);

catch (SQLException err) {
OptionPane.showMessageDialog(main.this, err.getMessage());
}
}

```

7_ الان قم بوضع (DoConnect) في الدالة الرئيسية للبرنامج كما في الشكل التالي:

8_ قم بتشغيل برنامج و ستجد قيم جدول البيانات قد ظهرت في مربعات النصوص كما في الشكل التالي:



9_ برمجة ازرار التنقل بين السجلات التالي / السابق / الاول / الاخير..و ذلك على النحو التالي:

برمجة زر التالي

المطلوب من هذا الزر هو عند النقر عليه يجب أولا التحقق ما إذا كان هناك سجل التالي للانتقال إليه أم لا وثانيا، إذا كان هناك السجل التالي، يقوم بعرضه في الحقول النصية.... ويتم ذلك باستخدام الكود التالي _ قم بالنقر على هذا الزر ليتم فتح صفحة الكود ثم اكتب الكود التالي:

```
try {  
  
if ( rs.next( ) ) {  
int id_col = rs.getInt("ID");  
String id =Integer.toString(id_col);  
String first_name = rs.getString("First_Name");  
String last_name = rs.getString("Last_Name");  
String job = rs.getString("Job_Title");  
  
textID.setText(id);  
textFirstName.setText(first_name);  
textLastName.setText(last_name);  
textJobTitle.setText(job);  
}  
else {  
rs.previous( );  
JOptionPane.showMessageDialog(main.this, "End of File");  
}  
}  
catch (SQLException err) {  
JOptionPane.showMessageDialog(main.this, err.getMessage());  
}  
}
```


_ في السطر الاول من الكود استخدمنا تعليمة try...catch و ذلك لمعالجة الاخطاء التي ستظهر في حالة عدم وجود سجل للانتقال إلى ... ثم مررنا الرسالة (End of File) اليها حتى يتم اظهارها للمستخدم لاختبارة بذلك

_ في السطر الثاني استخدمنا الجملة الشرطية else.....if وذلك للتحق من وجود سجل للانتقال الية في حالة وجوده يتم الانتقال الية و إسناد حقول الجدول الى مربعات النصوص...

برمجة زر السابق

الكود الخاص بهذا يشبه الى حد كبير الكود الخاص بزر التالي إعلاء مع فرق بسيط ..فبدلا من استخدام **rs.Next** بعد كملة **if** سوف نستخدم ... **rs.previous** و ايضا بعد كلمة **else** بدلا من استخدام **rs.previous** سوف نستخدم **rs.next** بالتالي فإن الكود لهذا الزر كما يلي:

```
try {  
  
if ( rs.previous() ) {  
int id_col = rs.getInt("ID");  
String id =Integer.toString(id_col);  
String first_name = rs.getString("First_Name");  
String last_name = rs.getString("Last_Name");  
String job = rs.getString("Job_Title");  
  
textID.setText(id);  
textFirstName.setText(first_name);  
textLastName.setText(last_name);  
textJobTitle.setText(job);  
}  
else {  
rs.next();  
JOptionPane.showMessageDialog(main.this, "End of File");  
}  
}  
catch (SQLException err) {  
JOptionPane.showMessageDialog(main.this, err.getMessage());  
}  
}
```

برمجة زر الامر الاول

المطلوب من هذا الزر هو عند النقر عليه الانتقال الى اول حقل في جدول قاعدة البيانات ...و بالتالي فالكود سهل جدا فقط استخدم الامر **rs.first** ثم إسناد الحقول الى مربعات النصوصو بالتالي فالكود سيكون على النحو التالي:

```
try {  
  
rs.first() ;  
int id_col = rs.getInt("ID");  
String id =Integer.toString(id_col);  
String first_name = rs.getString("First_Name");  
String last_name = rs.getString("Last_Name");  
String job = rs.getString("Job_Title");  
  
textID.setText(id);  
  
}
```

```

textFirstName.setText(first_name);
textLastName.setText(last_name);
textJobTitle.setText(job);
}
catch (SQLException err) {
JOptionPane.showMessageDialog(main.this, err.getMessage());
}

```

برمجة زر (الايخبر) المطلوب من هذا الزر هو الانتقال الى اخر سجل في جدول قاعدة البياناتو هو نفس الكود أعلاه مع تغيير فقط rs.first الى rs.last

10_ برمجة ازرار التعامل مع البيانات

برمجة زر جديد

المطلوب من هذا الزر عند النقر عليه مسح القيم الموجودة في مربعات النصوص و ذلك لتهيئة ادخال القيم الجديدة و ذلك عن طريق الكود التالي:

```

try {

textFirstName.setText("");
textLastName.setText("");
textJobTitle.setText("");
textID.setText("");

}
catch (SQLException err) {
System.out.println(err.getMessage());
}

```

برمجة زر الامر حفظ

المطلوب من هذا الزر عند النقر عليه فتح سجل في جدول قاعدة البيانات وحفظ القيم الموجودة في مربعات النصوص في هذا السجل...ولكى تتمكن من ذلك يجب تحريك المؤشر إلى ما يسمى **Insert Row** وهذا يخلق سجل فارغ في resultset. ثم يتم إسناد القيم في مربعات النصوص إلى ResultSe و ذلك على النحو التالي:

بداية نقوم بالاعلان عن المتغيرات و إسناد قيم مربعات النصوص اليها على حسب نوع كل حقل و ذلك على النحو التالي :

```

String id1 =textID.getText();
int newid=Integer.parseInt(id1);

String first1 =textFirstName.getText();
String last1 =textLastName.getText();
String job1 = textJobTitle.getText();

```

ثانيا :بعد ذلك نقوم باسناد القيم المخزنة في المتغيرات أعلاه الى حقول الجدول المراد إضافة البيانات اليه ثم نقوم بعملية إدراج لتلك الحقول و ذلك عن طريق الكود التالي:

```

try{
    rs.moveToInsertRow( );

    rs.updateInt("ID", newid);
    rs.updateString("First_Name", first1);
    rs.updateString("Last_Name", last1);
    rs.updateString("Job_Title", job1);

    rs.insertRow( );

```

ثالثا نقوم بإغلاق الكائن stmt و الكائن rs و إظهار رسالة للمستخدم تخيرة باتمام عملية الحفظ و الاضافة و ذلك عن طريق الكود التالي:

```

stmt.close();
rs.close();
JOptionPane.showMessageDialog(null, "");

```

و الكود النهائي لزر الامر حفظ سيكون على النحو التالي:

```

public void DoConnect( ) {
    try {
        String host = "jdbc:derby://localhost:1527/Employees";
        String uName = "admin";
        String uPass= "admin";
        con = DriverManager.getConnection( host, uName, uPass);
        stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE );
        String SQL = "select * from APP.WORKERS";
        rs = stmt.executeQuery( SQL );

        rs.next( );
        int id_col = rs.getInt("ID");
        String id =Integer.toString(id_col);
        String first_name = rs.getString("First_Name");
        String last_name = rs.getString("Last_Name");
        String job = rs.getString("Job_Title");

        textID.setText(id);
        textFirstName.setText(first_name);
        textLastName.setText(last_name);
        textJobTitle.setText(job);
    }
    catch (SQLException err) {
        JOptionPane.showMessageDialog(main.this, err.getMessage());
    }
}

```

برمجة زر الامر حذف عن طريق الكود التالي:

```

try{
rs.deleteRow();

stmt.close();
rs.close();
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(main.this, err.getMessage());
}

```

الفصل الثاني : قواعد بيانات MySQL

في الفصل السابق تعلمنا كيفية إنشاء قاعدة بيانات Java DB من خلال NetBeans و كيفية التعامل معها أما في هذا إن شاء الله تعالى سنتعلم كيفية إنشاء قاعدة بيانات MySQL و الاتصال بها سواء كانت تلك القاعدة على القرص الصلب أو على شبكة الانترنت...

أولا التعريف بقواعد بيانات MySQL:

قاعدة بيانات MySQL هي عبارة عن قاعدة بيانات علاقية من اشهر قواعد البيانات مفتوحة المصدر التي يمكنك استخدامها مجانا و جميع تطبيقات الويب تستخدم تلك القاعدة لتخزين البيانات و يمكنك استخدامها ايضا في تطبيقات سطح المكتب..

ثانيا إنشاء قاعدة بيانات MySQL من خلال NetBeans على السرفر المحلي (القرص الصلب):

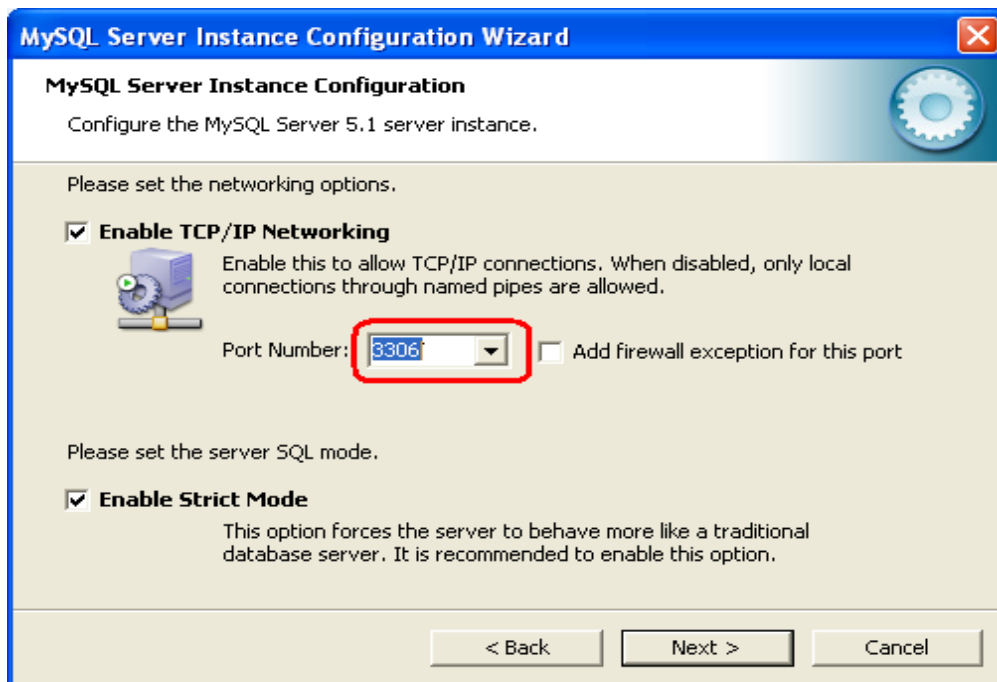
توجد طرق عديدة لإنشاء قاعدة بيانات MySQL منها استخدام موجه الاوامر Dos و ايضا بعض البرنامج الاخرى التي تقوم بذلك.... لكن دراستنا ستكون عن كيفية إنشائها من خلال بنية NetBeans وذلك على النحو التالي:

1_ سنحتاج الى تثبيت حزمة mysql-server و حزمة mysql-client و تلك الحزمتين ستجدها في الملف mysql-essential-5.1.5 و يمكنك تحميله من الرابط التالي:

<http://downloads.skysql.com/archive/signature/p/mysql/f/mysql-essential-5.1.52-win32.msi/v/5.1.52>

<http://dev.mysql.com/downloads/mysql/5.1.html>

2_ بعد الانتهاء من تحميل الملف السابق قم بتثبيته و يكفيك لذلك الضغط على زر..(next) و لكن كل ما اریده منك هو عند ظهور الشاشة التالية:

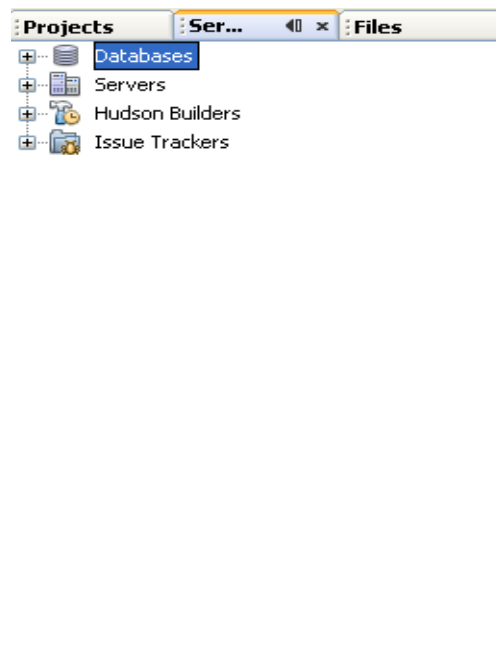


اريد منك أن تعلم و تحفظ رقم البور أو المنفذ الخاص بك وهو نفسه الموجود الصورة أعلاهو ايضا عند ظهور الشاشة التى تطالبك بوضع كلمة المرور للسرفر أن حفظها كما فى الشكل التالى:

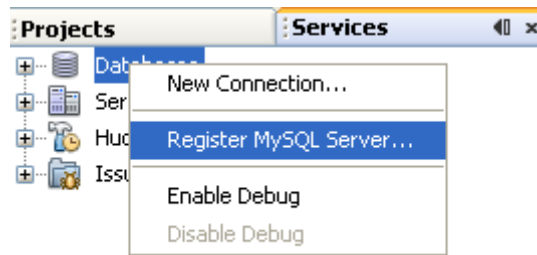


رقم المرور الذى سوف تقوم بادخاله ستخدمة عند انشاء قاعدة البيانات بواسطة NetBeans

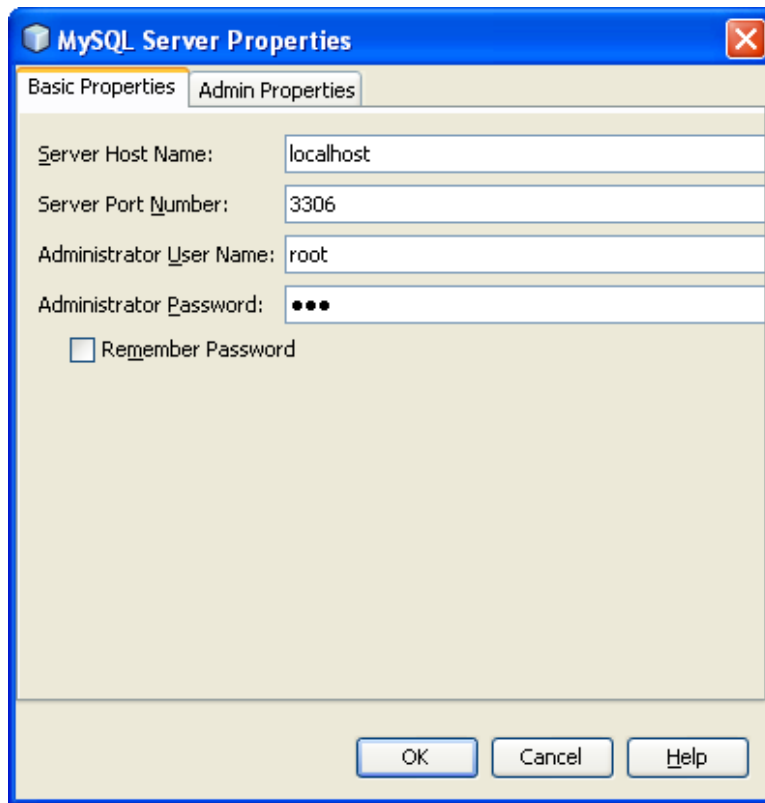
3_قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services فى القوائم الجانبية كما فى الشكل التالى :



4_ قم بتحديد ايقونة Databases ثم كليك يمين ومن القائمة التي سوف تظهر اختر الامر register mysql service كما في الشكل التالي:

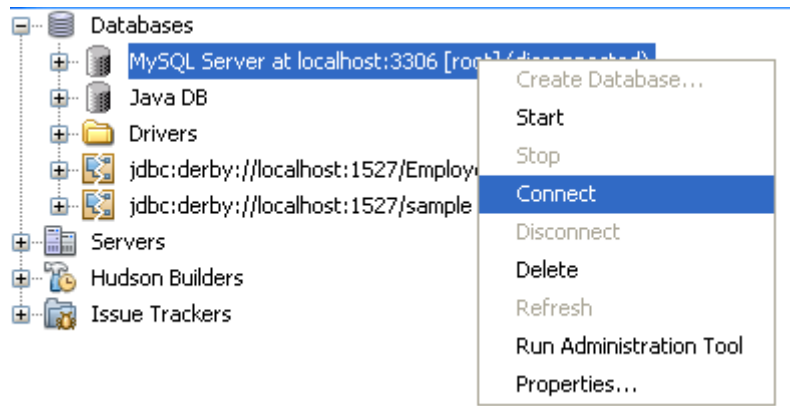


5_ عند النقر على الامر register mysql service كما هو ظاهر أعلاه سيظهر لك المربع الحوارى التالي:

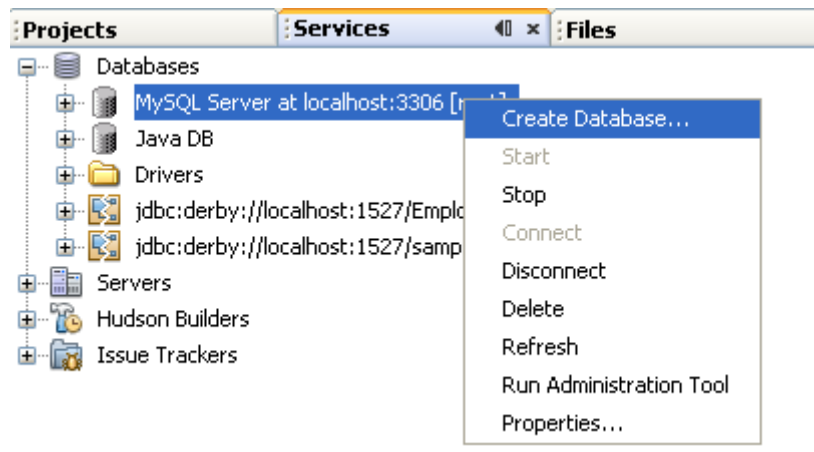


يطالبك هذا المربع service host name وهو فى مثالنا هذا localhost وايضا رقم المنفذ service port وطبعاً هو نفس رقم المنفذ الذى حددناه اثناء تنصيب... service اما اسم المستخدم فهو root وكلمة المرور هى نفس كلمة المرور التى حددناها اثناء تنصيب السرفر.... بعد اتمام ادخال البيانات انقر على زر الامر (موافق)

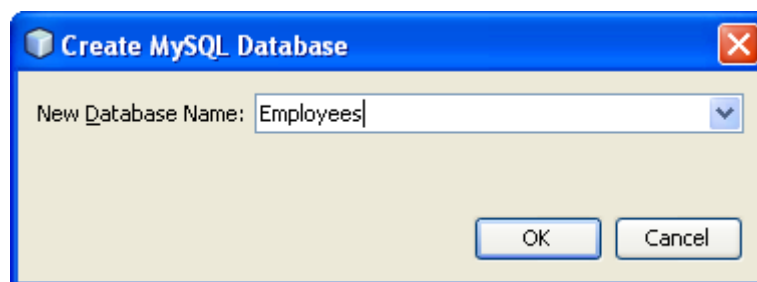
6_ بعد النقر على (ok) ستجد فى التبويب Services ايقونة mysql Services at localhost:3306[root قم بتحديد تلك الايقونة ثم كليك يمين و من القائمة التى ستظهر اختر الامر connet كما فى الشكل التالي:



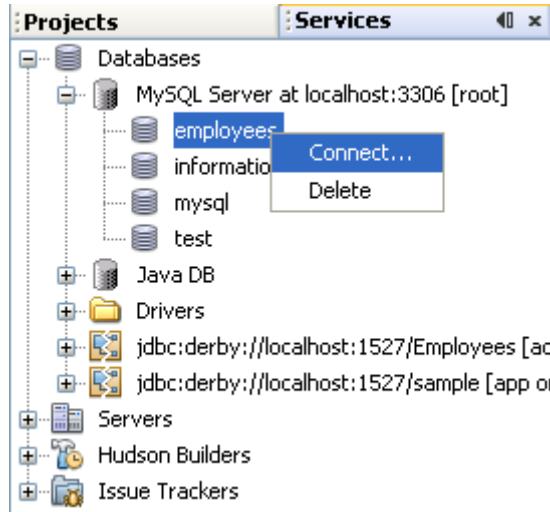
7_ بعد تشغيل السرفر عاود الكرة مرة أخرى بتحديد الايقونة ثم كليك يمين و من القائمة التي ستظهر اختر الامر Create Database كما في الصورة التالية:



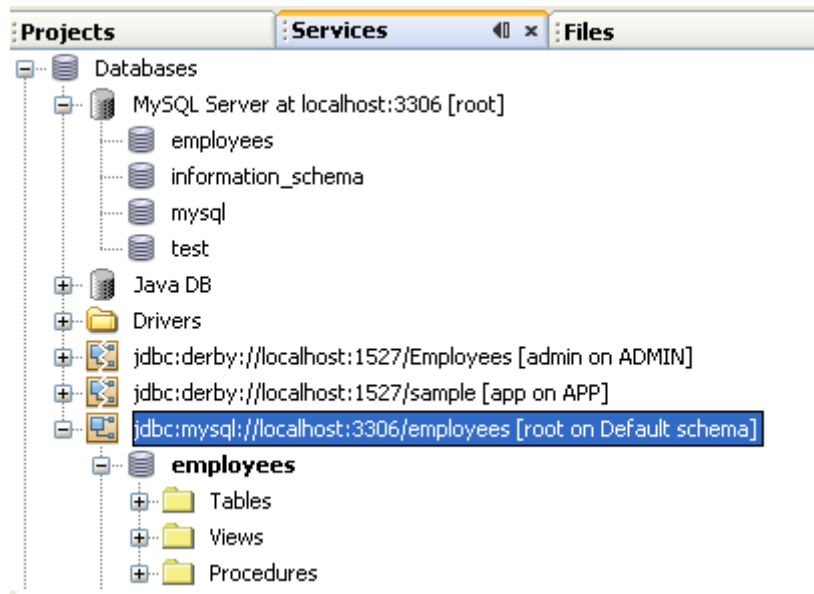
8_ سيظهر لك المربع الحوارى التالى يطالبك بإدخال اسم لقاعدة البيانات المراد انشائها و ليكن مثلا Employees كما في الصورة التالية:



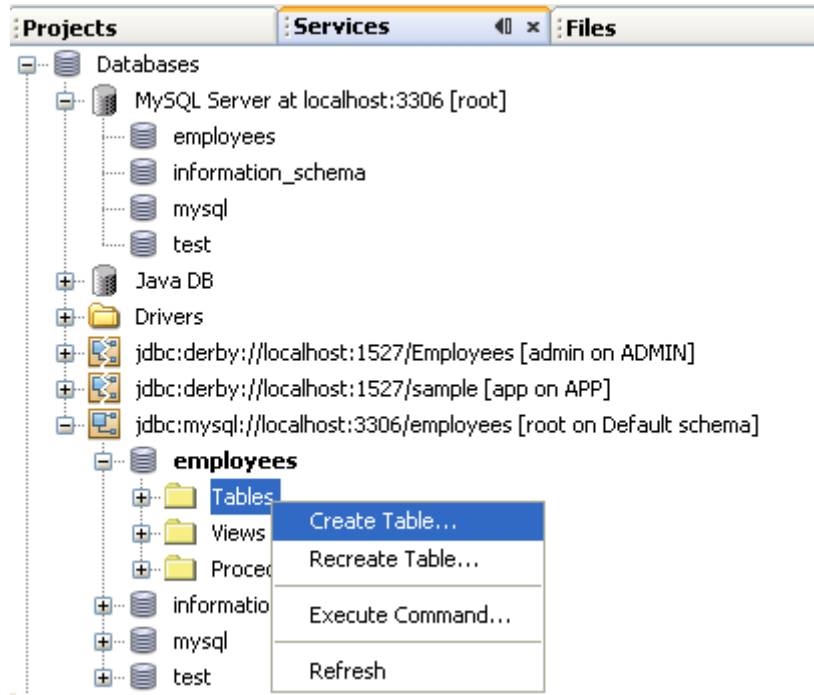
9_ قم الان بالنقر دبل كليك على ايقونة mysql Services at localhost:3306[root@localhost:3306] ستجد تحتها قاعدة البيانات التي قمنا بإنشائها... قم بتحديثها ثم كليك يمين و من القائمة التي ستظهر اختر الامر connet كما في الشكل التالى:



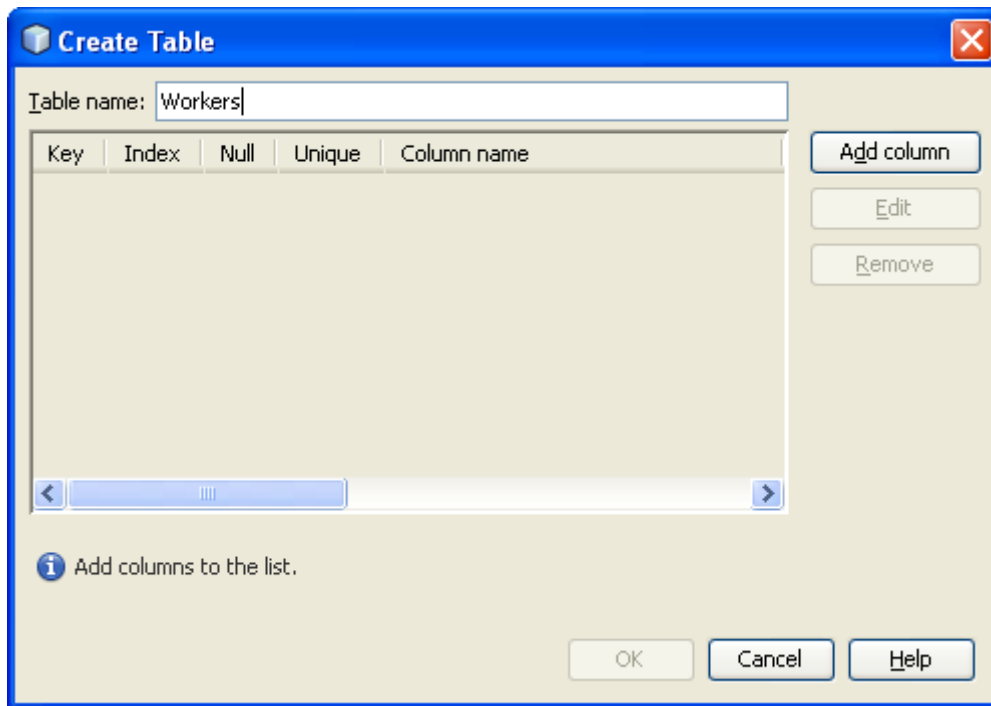
10_ من المربع الحوارى الذى سيظهر أدخل كلمة المرور ثم أنقر فوق (ok) بعد ما يتم عملية الاتصال بالخادم ستجد انه قد تم انشاء ثلاث مجلدات وهى Tables , Views, Procedures و حتى تتمكن من رؤية تلك المجلدات فى التويب Server قم بالنقر دبل كليك على قاعدة البيانات التى انشأناها Employees لتفتح لك بعض المجلدات و لترى المجلدات الثلاث كما فى الشكل التالى:



11_ قم بالنقر شمال على المجلد Tables و من القائمة التى تظهر، حدد الامر Create Table كما فى الصورة التالية:



12_ عند النقر كما تم شرحه أعلاه يظهر لك المربع الحوارى التالى:



في مربع النص Table Name في الأعلى، أ حذف الاسم الافتراضى..Untitled ثم اكتب اسما جديدا للجدول و ليكن مثلا Workers.... وهكذا سيكون لديك جدول يسمى Workers

13_ إضافة حقول للجدول و يتم ذلك عن طريق النقر فوق زر الامر Add Column فى المربع الحوارى أعلاه ليظهر لك المربع الحوارى التالى:

وهكذا يمكنك إنشاء قاعدة بيانات mysql و جداولها من خلال النت بينز على القرص الصلب

ثالثا : كيفية إنشاء قاعدة بيانات mysql على مقلم بعيد (شبكة الإنترنت):

بداية يجب أن تمتلك سرفر و يدعم هذا السرفر قواعد البيانات mysql لذلك سنقوم بالتجربة على سرفر مجاني على الرابط التالي:

http://www.db4free.net/d4f_apply.php

قم بالدخول على الرابط أعلاه ستجد أنه يطالبك ببعض البيانات لكي تستطيع إنشاء قاعدة بيانات mysql على هذا السرفر كما في الصورة التالية:

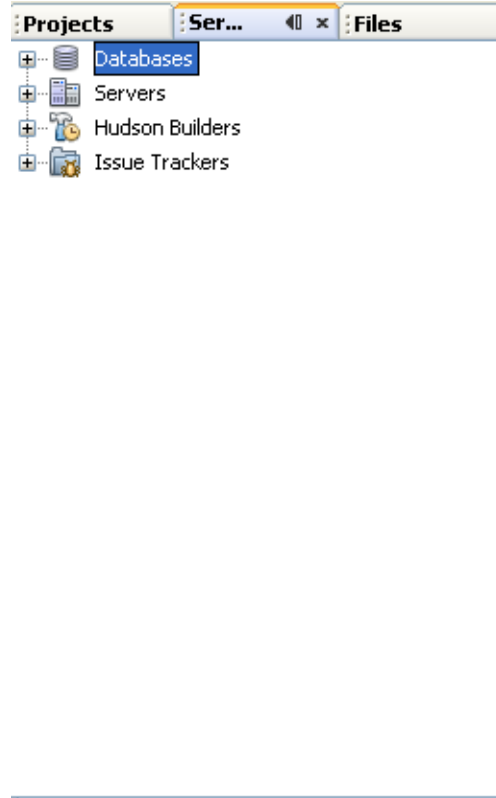
MySQL database name:	<input type="text" value="mysql20"/>
MySQL username:	<input type="text" value="egytech"/>
MySQL user password:	<input type="password" value="••••••"/>
MySQL user password verification:	<input type="password" value="••••••"/>
Allow access from following IP **: :	<input style="width: 100px;" type="text" value="%"/>
Default collation ***:	<input style="width: 100px;" type="text" value="utf8_general_ci"/>
Email address:	<input type="text" value="zakyshny@gmail.com"/>

I have read the [conditions of use](#) and I agree with them.

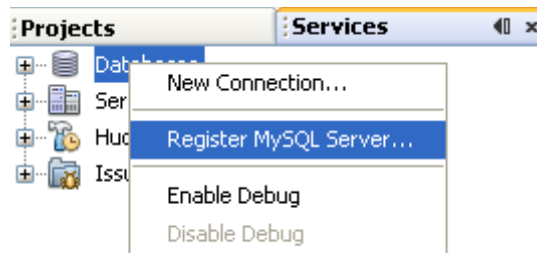
قم بادخال اسم قاعدة البيانات mysql وفي مثالنا هذا كتبنا ...mysql20 بعد ذلك أدخلنا اسم المستخدم... egyptech ثم أدخلنا كلمة المرور ولا تنسى أن تضع البريد الإلكتروني الخاص بك لأنه سوف يتم ارسال كود التنشيط عليه ادخل على الميل هنالقي رسالة تأكيد الاشتراك

قم بتأكيد الاشتراك ثم اتبع الخطوات التالية:

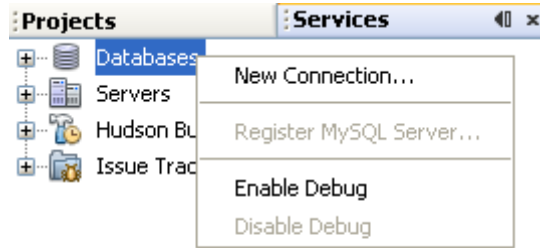
1_ قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services فى القوائم الجانبية كما فى الشكل التالى:



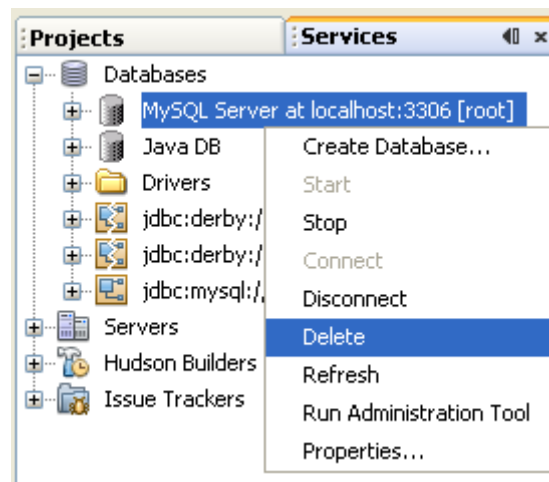
2_ قم بتحديد ايقونة Databases ثم كليك يمين ومن القائمة التى سوف تظهر أختَر الامر register mysql service كما فى الشكل التالى:



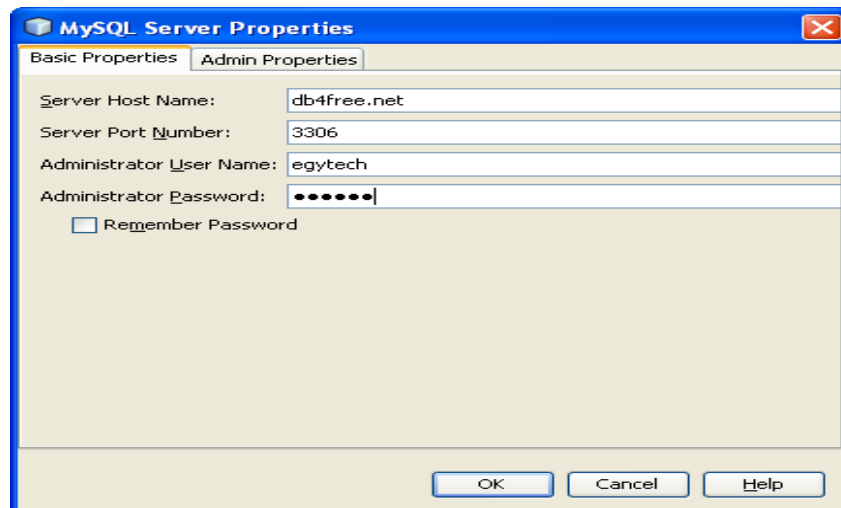
لكن ربما تجد أن الامر register mysql service غير نشط كما فى الصورة التالية:



وهذا معناه أنك قد قمت من قبل بتسجيل معلومات السرفر لذلك قم بحذف معلومات السرفر القديم كما فى الشكل التالى:

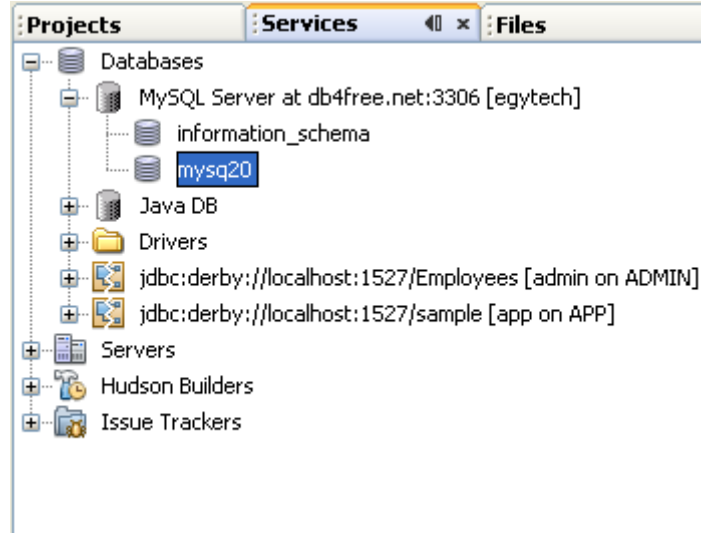


3_ عند النقر على الامر register mysql service كما هو ظاهر أعلاه سيظهر لك المربع الحوارى التالى:

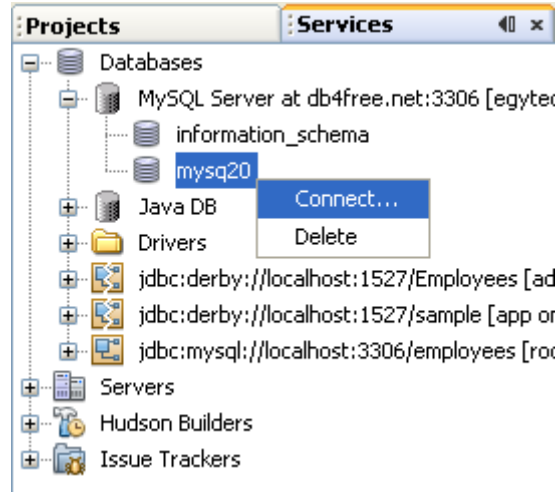


يطالبك هذا المربع service name وهو في مثالنا هذا db4free.net وايضا رقم المنفذ service port وطبعاً هو نفس رقم المنفذ الذي حددناه اثناء تنصيب... service اما اسم المستخدم فهو egytech وهي نفس اسم المستخدم وكلمة المرور التي حددناها اثناء التسجيل في السرفر.... بعد اتمام ادخال البيانات انقر على زر الامر (موافق).

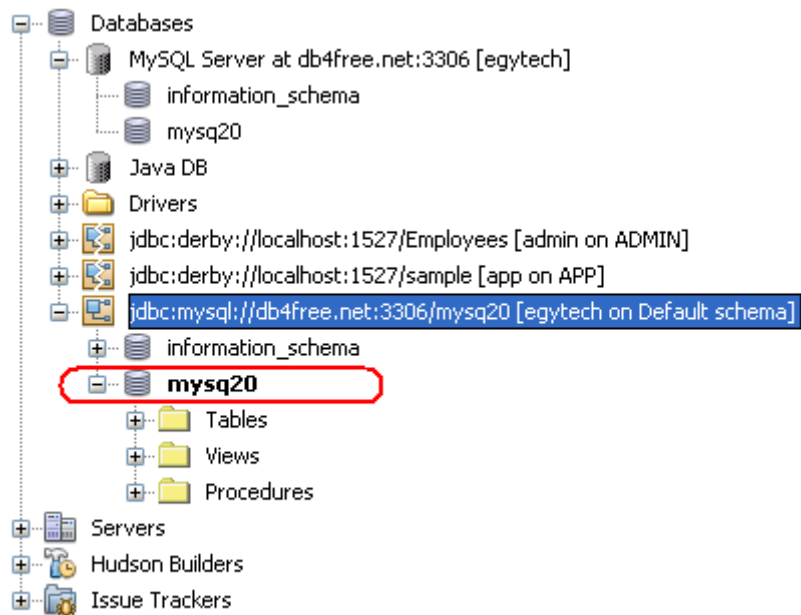
4_ بعد ان يتم الاتصال بالسرفر db4free.net ستجد انه قد جلب جميع قواعد البيانات الموجودة عليه كما في الشكل التالي:



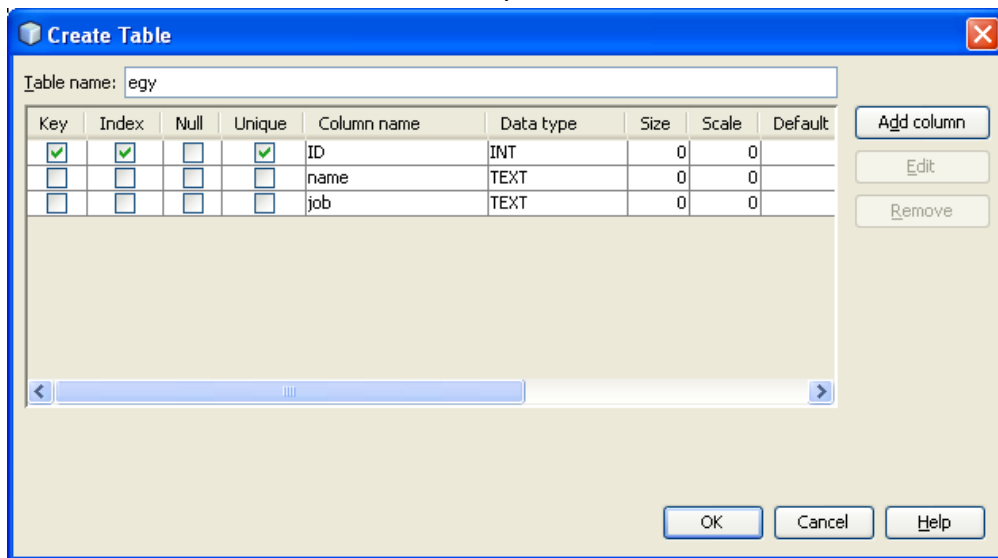
من الصورة أعلاه حدد قاعدة البيانات التي قد قمنا بإنشائها اثناء التسجيل في السرفر وهي mysq20 ومن القائمة التي ستظهر أختار الامر connect كما في الشكل التالي:



5_ بعد عملية الاتصال بقاعدة البيانات mysq20 ستجد ايقونة بأسم jdbc:mysql://db4free.net:3306/mysq20 [egytech on Default schema] وعند النقر عليها دبل كليك ستجد قاعدة البيانات التي انشأناها كما في الشكل التالي :



6_ الات قم بالنقر يمين على المجلد Tables و من القائمة التي تظهر، حدد الامر Create Table كما في الصورة التالية :



ومن هنا نستطيع انشاء الجدوال الخاصة بنا في قاعدة بيانات mysql20 على سرفر db4free.net وكما ترون في الصورة أعلاه قمنا بانشاء الجدول EGY وبه الحقول التالية ID , NAME . JOB و لو فتحت السرفر الخاص بك ستجد الجدوال به كما في الصورة التالية:

The screenshot shows the phpMyAdmin interface for a MySQL database named 'mysq20'. The top navigation bar includes 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', and 'Routines'. A table named 'egy' is selected, and its structure is displayed with columns: ID (int(11)), NAME (text), and JOB (text). Below the structure, there are options to 'Check All / Uncheck All' and 'With selected:'. A red circle highlights the 'JOB' column in the structure table.

Below the structure, there is a 'Create table' section with a 'Name' field and a 'Number of columns' field. A red arrow points to the 'JOB' column in the structure table, and another red arrow points to the 'JOB' column in the structure table below the 'Create table' section.

The bottom part of the screenshot shows a 'Print view Relation view Propose table structure @ Track table' section with a 'Add' field and radio buttons for 'At End of Table', 'At Beginning of Table', and 'After ID'. There is also a '+ Indexes' link.

Handwritten Arabic text in red: 'جدول البيانات' (Table of data) is written over the top part of the screenshot. Another handwritten Arabic text in black: 'حقول جدول البيانات التي انشأناها بالنت بينز' (Fields of the table we created with NetBeans) is written over the bottom part of the screenshot.

...في هذا المثال التجريبي يجب ملاحظة أننا قد أنشأنا قاعدة بيانات mysq20 عن طريق السرفر db4free.net عند التسجيل فيه و ليس عن طريق NetBeans و كل ما أنشأناه بالنت بينز هو جداول تلك القاعدة ... لكن تستطيع أن تشيء قاعدة بيانات باستخدام NetBeans فقط قم بتحديد السرفر الخاص بك ثم كليك يمين ومن القائمة التي ستظهر اختر الامر Create Database كما في الصورة التالية:

The screenshot shows the NetBeans IDE interface with the 'Services' tab selected. A context menu is open over the 'MySQL Server at db4free.net:3306 [readonly]' service. The menu items are: 'Create Database...', 'Start', 'Stop', 'Connect', 'Disconnect', 'Delete', 'Refresh', 'Run Administration Tool', and 'Properties...'. The 'Create Database...' option is highlighted.

لكن للأسف لن تستطيع ان تجرب ذلك على هذا السرفر المجانى db4free.net و ذلك لكون لا يتيح الا بإنشاء قاعدة بيانات واحد فقط و هى التى أنشأناها عند التسجيل.... لذلك جرب على سرفر آخر

رابعا : الاتصال بقواعد بيانات mysql

تعلمنا كيفية إنشاء قاعدة بيانات MySQL سواء كانت على القرص الصلب أو على شبكة الانترنت....اليوم بإذن الله تعالى سنتعلم سويا كيفية الاتصال بتلك القاعدة من خلال JAVA وذلك على النحو التالى :

1_قم بإنشاء قاعدة بيانات MySQL و ليكن Employees بها جدول وليكن egy وبه الحقول التالية

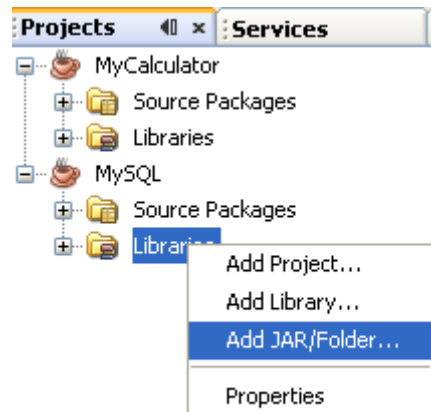
Column Name: ID
Data Type: int

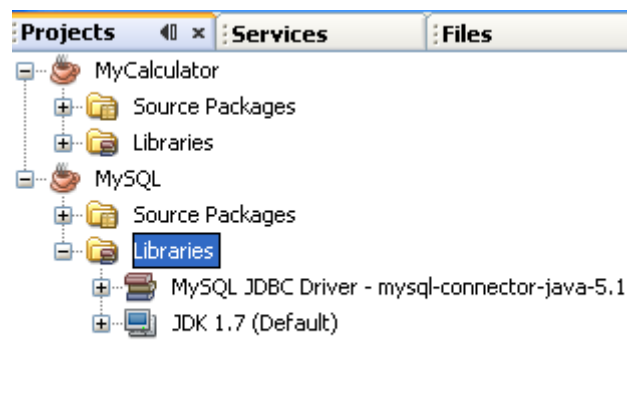
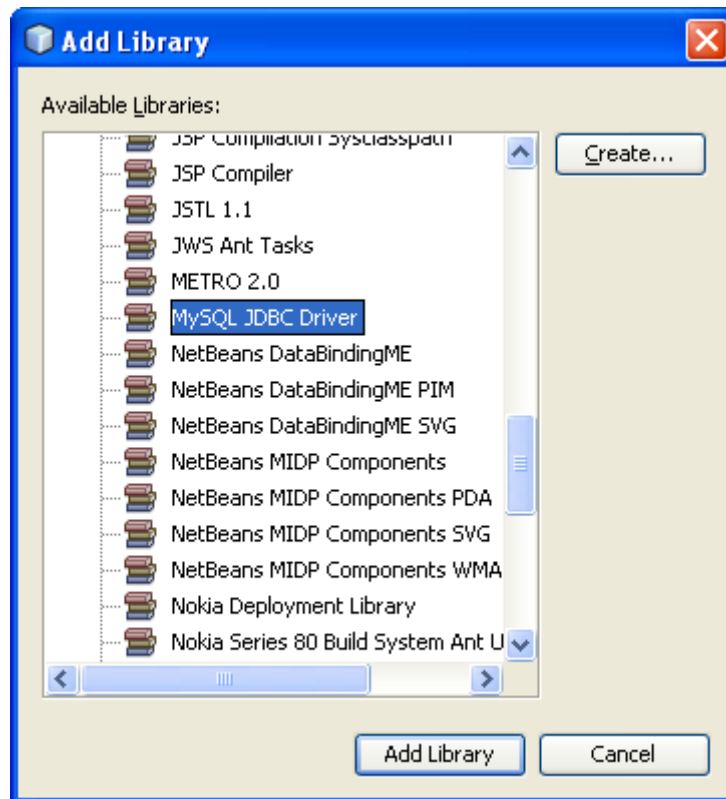
Column Name: Name
Data Type: Text

Column Name: Job
Data Type: Text

2_قم بفتح مشروع جديد فى NetBeans و ليكن MySQL مع إزالة علامة (الصح) من خانة create main class وذلك لاننا لانريد الان إنشاء الفئة الرئيسية....قم بإنشاء نافذة رسومية وضع عليها العناصر الرسومية اللازمة كما فى الشكل التالى:

3_قم بادراج المكتبة MySQL JDBC Driver و ذلك عن طريق تحديد ايقونة Library فى القائمة الجانبية للمشروع ومن القائمة التى ستظهر أختر الامر Add Library سيتم فتح لك مربع حوارى للمكتبات الموجودة على جهازك أختر منها MySQL JDBC Driver ثم ok لتجد المكتبة قد اضيفت للمشروع كما فى الصور التالية:





4_ نقوم استدعاء جميع الدوال التابعة للمكتبة **SQL** و ايضا ادراج المكتبة التي تتعامل مع صناديق الحوار **JOptionPane** كما تعلمنا سابقا و على النحو التالي:

```
import java.sql.*;
import javax.swing.JOptionPane;
```

5_ الاعلان عن الكائنات التي تستخدم لاتمام عملية ربط قاعدة البيانات مع البرنامج باستخدام الكائنات كما تعلمنا سابقا وهي : الكائن **Connection** وهو الخاص بنص الاتصال... الكائن **Statement** وهو الخاص بجمل التحكم.... و الكائن **Result Set** وهو الخاص بمؤشر البيانات وذلك عن طريق الكود التالي:

```
Connection con;
Statement st;
ResultSet rs;
```

6_ نقوم بإنشاء طريقة و ليكن مثلا DoConnect كما توعدنا فى الدروس السابقة و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجدولها

7_ داخل الطريقة سنقوم بالاعلان عن ثلاث متغيرات وليتم تخزين فيها رابط قاعدة البيانات واسم المستخدم و كلمة المرور و ذلك على النحو التالى:

```
String url = "jdbc:mysql://localhost:3306/employees ";
String user = "root"
String password = "123"
```

8_ نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManager من الاتصال بقاعدة بيانات employees و ذلك عن طريق الكود التالى:

```
con = DriverManager.getConnection(url, user, password);
```

ولكن فى حالة عدم القدرة على الاتصال بقاعدة البيانات سيحدث الاستثناء SQLException لذلك سوف يتم معالجة كما فعلنا فى الدرس السابق.

9_ نقوم بأستدعاء الطريقة createStatement لاجاء الكائن Statement والذي سيستخدم للاستعلام من قاعدة البيانات وذلك عن طريق الكود التالى:

```
st = con.createStatement();
```

10_ نقوم بجلب قيم الحقول فى الجدول و ذلك تنفيذ الاستعلام من خلال استدعاء الطريقة executeQuery الموجودة فى الصنف Statement و الذى قد عبرنا عنه بالكائن rs و ذلك على النحو التالى:

```
rs = st.executeQuery("select * from egypt");
```

11_ نقوم بإسناد حقول الجدول الى مربعات النصوص فى النافذة عن طريق الكود التالى:

```
rs.next();
int id_col = rs.getInt("ID");
String id = Integer.toString(id_col);
String name = rs.getString("name");
String job = rs.getString("job");

textid.setText(id);
textname.setText(name);
textjob.setText(job);
```

وهذا طبعا نفس الكود الذى تم استخدامه فى الدرس السابق... كما أننا قمنا بالاعلان عن المتغير الرقمية الصحيح id_col و ذلك لتخزين قيمة الحقل (ID) الموجود فى جدول البيانات الذى نوعه ايضا رقمى.. (integer) لكن المشكلة ان حقول النص فى الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصى (id) لتخزين قيمة المتغير id_col بعد تحويله الى متغير نصى باستخدام الامر Integer.toString

و فى النهاية فإن الكود النهائى سيكون كما فى الصورة التالية:

```
public void DoConnect() {

String url = "jdbc:mysql://localhost:3306/employees";
String user = "root";
```

```

String password = "123";

try {
    con = DriverManager.getConnection(url, user, password);
    st = con.createStatement();
    rs = st.executeQuery("select * from egypt");

    rs.next( );
    int id_col = rs.getInt("ID");
    String id = Integer.toString(id_col);
    String name = rs.getString("name");
    String job = rs.getString("job");

    textid.setText(id);
    textname.setText(name);
    textjob.setText(job);

    } catch (SQLException ex) {
    JOptionPane.showMessageDialog(NewJFrame.this, ex.getMessage());
    }
}

```

12_ الان قم بوضع DoConnect فى الدالة الرئيسية للبرنامج وستجد بيانات جدول قاعدة البيانات قد ظهرت فى مربعات النصوص كما فى الشكل التالى :

أما برمجة أزار التنقل و الحذف و الاضافة و خلافة فهى نفس فكرة الدرس السابق

الفصل الثالث : قواعد بيانات Access

مازلنا في الدروس الخاصة بقواعد البيانات في الجافا... ففي الدرس السابق تعلمنا سويا كيفية إنشاء قاعدة بيانات Java DB وكيفية التعامل معها... اليوم إن شاء الله تعالى سنتعلم سويا أيضا كيفية الاتصال و التعامل مع قاعدة بيانات أكسس وذلك على النحو التالي:

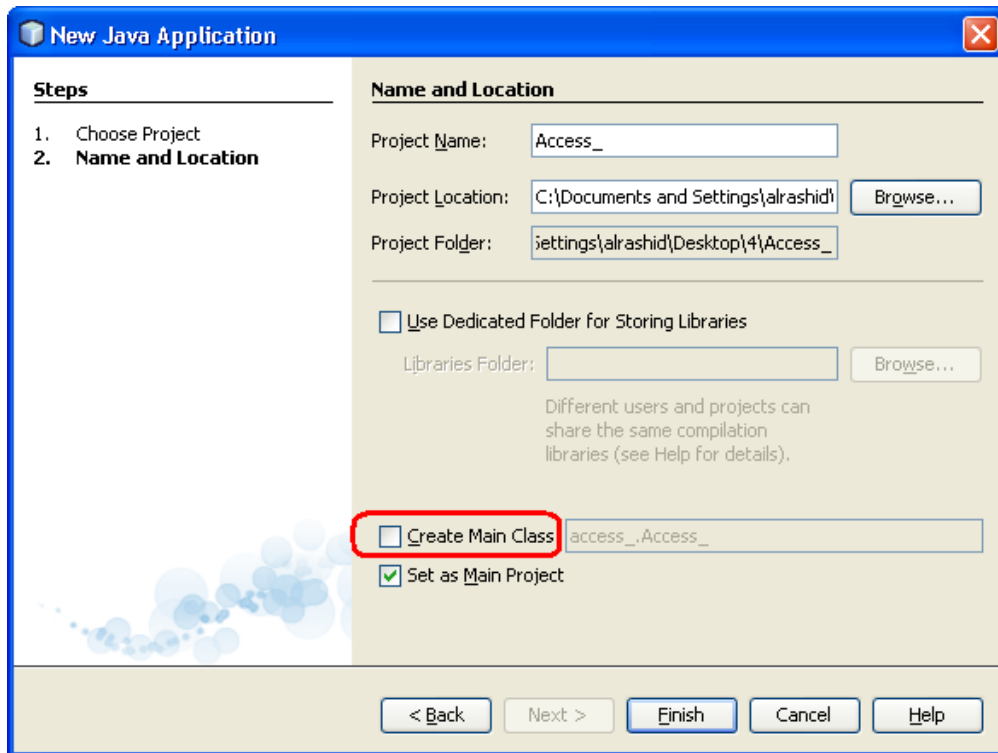
1_ قم بإنشاء قاعدة بيانات أكسس و ليكن db1 بها جدول وليكن Workers وبه الحقول التالية

Column Name: ID
Data Type: Number

Column Name: Name
Data Type: Text

Column Name: Job
Data Type: Text

2_ قم بفتح مشروع جديد في NetBeans و ليكن Access مع إزالة علامة (الصح) من خانة create main class وذلك لاننا لانريد الان إنشاء الفئة الرئيسية كما في الشكل التالي:



3_ قم بإنشاء نافذة رسومية وضع عليها العناصر الرسومية اللازمة كما في الشكل التالي:

4_ قم بنقل قاعدة البيانات db1 التي قمت بإنشائها الى المجلد الرئيسي للمشروع الذي قمت بإنشائه من خلال

5_ نقوم استدعاء جميع الدوال التابعة للمكتبة SQL و ايضا ادراج المكتبة التي تتعامل مع صناديق الحوار JOptionPane كما تعلمنا سابقا و على النحو التالي:

```
import java.sql.*;
import javax.swing.JOptionPane;
```

6_ الاعلان عن الكائنات التي تستخدم لاتمام عملية ربط قاعدة البيانات مع البرنامج باستخدام الكائنات كما تعلمنا سابقا وهي : الكائن Connection وهو الخاص بنص الاتصال ...الكائن Statement وهو الخاص بجمل التحكم و الكائن Result Set وهو الخاص بمؤشر البيانات وذلك عن طريق الكود التالي:

```
1  import java.sql.*;
2  import javax.swing.JOptionPane;
3
4  public class NewJFrame extends javax.swing.JFrame {
5
6      Connection con;
7      Statement stmt;
8      ResultSet rs;
9
10     public NewJFrame() {
11         initComponents();
12
13
14
15     }
```

ويمكنك ملاحظة أننا قمنا بالاعلان تلك الكائنات على مستوى الفئة حتى تتمكن من استخدامها في أي حدث من الاحداث داخل الفئة

7_ نقوم بإنشاء طريقة و ليكن مثلا DoConnect و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجداولها و يتم إنشاء الطريقة باستخدام الكود التالي:

```
public void DoConnect ( ) {
}
}
```

8_ نقوم بكتابة الكود التالي لتحميل JDBC Driver الخاص بقواعد البيانات اكسس و ذلك على النحو التالي:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

كما في السطر السابق قمنا باستدعاء الطريقة forName وذلك لتحميل الفصيلة Class التي تحتوي على مشغل قاعدة البيانات ... و تحميل هذه الفصيلة قد يتسبب في حدوث استثناء في حالة أن الفصيلة المطلوبة لا يمكن تحميلها وهذا الاستثناء هو ClassNotFoundException و بالتالي سنقوم بمعالجة هذا الاستثناء على النحو التالي:

```
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
} catch (ClassNotFoundException ex) {
    JOptionPane.showMessageDialog( NewJFrame.this, ex.getMessage());
}
```

9_ نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManager من الاتصال بقاعدة بيانات db1 وذلك عن طريق الكود التالى :

```
con = DriverManager.getConnection("jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=db1.mdb");
```

يقوم هذا السطر بعملية اتصال بقاعدة البيانات عن طريق نص الاتصال jdbc:odbc:Driver={Microsoft Access Driver ثم وضعنا أسم قاعدة البيانات db1.mdb ولكن فى حالة عدم القدرة على الاتصال بقاعدة البيانات سيحدث الاستثناء SQLException لذلك سوف يتم معالجة كما فعلنا فى الدرس السابق.

10_ نقوم بأستدعاء الطريقة createStatement لاجاء الكائن Statement والذي سيستخدم للاستعلام من قاعدة البيانات وذلك عن طريق الكود التالى:

```
stmt=con.createStatement();
```

11_ نقوم بالاعلان عن متغير نصى وليكن SQL لحفظ جملة الاستعلام فية و ذلك عن طريق الكود التالى:

```
String SQL = "select * from WORKERS";
```

12_ تنفيذ الاستعلام من خلال استدعاء الطريقة executeQuery الموجودة فى الصنف Statement و الذى قد عبرنا عنه بالكائن rs و ذلك على النحو التالى:

```
rs = stmt.executeQuery( SQL );
```

13_ نقوم بإسناد حقول الجدول الى مربعات النصوص فى النافذة عن طريق الكود التالى:

```
rs.next( );
int id_col = rs.getInt("ID");
String id =Integer.toString(id_col);
String name = rs.getString("name");
String job = rs.getString("job");

textid.setText(id);
textname.setText(name);
textjob.setText(job);
```

وهذا طبعا نفس الكود الذى تم استخدامه فى الدرس السابق...فبالنظرة السريعة على الكود سنجد أننا قمنا بالاعلان عن المتغير الرقمى الصحيح id_col و ذلك لتخزين قيمة الحقل (ID) الموجود فى جدول البيانات الذى نوعه ايضا رقمى (integer)... لكن المشكلة ان حقول النص فى الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصى (id)لتخزين فية قيمة المتغير id_col بعد تحويله الى متغير نصى باستخدام الامر Integer.toString

و فى النهاية فإن الكود النهائى سيكون كما فى الصورة التالية:

```

public void DoConnect ( ) {
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    } catch (ClassNotFoundException ex) {
        JOptionPane.showMessageDialog( NewJFrame.this, ex.getMessage ());
    }
    try {
        con = DriverManager.getConnection("jdbc:odbc:Driver=(Microsoft Access Driver (*.mdb));DBQ=db1.mdb;");
        stmt=con.createStatement();

        String SQL = "select * from WORKERS";
        rs = stmt.executeQuery( SQL );

        rs.next( );
        int id_col = rs.getInt("ID");
        String id =Integer.toString(id_col);
        String name = rs.getString("name");
        String job = rs.getString("job");

        textid.setText(id);
        textname.setText(name);
        textjob.setText(job);
    }
    catch (SQLException err) {
        JOptionPane.showMessageDialog( NewJFrame.this, err.getMessage ());
    }
}

```

14_ الآن قم بوضع DoConnect في الدالة الرئيسية للبرنامج وسنجد بيانات جدول قاعدة البيانات قد ظهرت في مربعات النصوص كما في الشكل التالي:

أما برمجة أزرار التنقل و الحذف و الاضافة و خلافة فهي نفس فكرة الدرس السابق

الفصل الرابع : قواعد بيانات Microsoft SQL Server

درس اليوم إن شاء الله تعالى عن كيفية الاتصال و التعامل مع قواعد بيانات Microsoft SQL Server من خلال تطبيقات الجافا وذلك على النحو التالي:

1_ قم بالنقر [هنا](#) وذلك للحصول على Microsoft JDBC Driver 4.0 for SQL Server أو سائق الاتصال بقواعد بيانات SQL Server و التي تدعم الإصدارات ٢٠٠٥، ٢٠٠٨، 2012 , R2 2008 ,

2_ قم بتشغيل الملف sqljdbc. EXE الذي تم تحميله ليتم استخراج الملفات إلى الدليل المحدد أو (الافتراضي و هو Microsoft JDBC Driver 4.0 for SQL Server) وسوف تجد ملفين بصيغة jar هما sqljdbc.jar ل (JDBC 3.0) و sqljdbc4.jar ل (JDBC 4.0)، بالإضافة إلى بعض ملفات DLL وملفات تعليمات HTML.

3_ قم بإضافة الملف sqljdbc.jar الى مشروعك إذا كنت تستخدم نسخة الجافا JDK 5.0 أو ملف sqljdbc4.jar إذا كنت تستخدم JDK 6.0 أو أى إصدار أعلى.... وذلك من خلال النقر يمين على المجلد Libraries ومن القائمة التي ستظهر اختر الامر Add Jar/Folder

4_ كتابة الشفرات اللازمة على النحو التالي:

أ_ إستيراد المكتبات اللازمة للعمل على النحو التالي:

```
import java.sql.Connection;  
import java.sql.DatabaseMetaData;  
import java.sql.DriverManager;  
import java.sql.SQLException;
```

ب_ نص الاتصال بقاعدة البيانات يكون عن طريق الكود التالي :

```
DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());
```

أو النص التالي:

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

ومع ذلك، هذا ليس مطلوباً منذ إصدار الجافا (JDK 6.0) مع السائق JDBC 4.0 لأنه يمكن الكشف عنه وتحميله تلقائياً طالما أن الملف sqljdbc4.jar موجود في مجلد المشروع.

ج_ بناء جملة URL لقاعدة بيانات SQL Server و الاتصال بها كما يلي:


```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress;user=sa;password=secret";
Connection conn = DriverManager.getConnection(dbURL);
if (conn != null) {
    System.out.println("Connected");
}
```

_حيث أن:

- localhost_ أسم المضيف أو عنوان ip للمعلم الذى عليه قاعدة البيانات
- sql_express_ قاعدة بيانات sql
- user_ أسم المستخدم للقاعدة
- password_ كلمة المرور للقاعدة

وكما ترى فى الشفرة أعلاه لعمل اتصال بالقاعدة نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManager مع تمرير رابط... url كما يمكنك استخدامه على النحو التالى:

```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
String user = "sa";
String pass = "secret";
conn = DriverManager.getConnection(dbURL, user, pass);
```

و ايضا يمكننا استخدام كائن java.util.Properties لتخزين خصائص الاتصال، كما فى المثال التالى:

```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
Properties properties = new Properties();
properties.put("user", "sa");
properties.put("password", "secret");
conn = DriverManager.getConnection(dbURL, properties);
```

فأختر الطريقة التى تناسبك.... على اى حال اليك الكود الكامل للاتصال بقاعدة بيانات SQL Server من خلال تطبيقات الجافا على النحو التالى:

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;

public class JdbcSQLServerConnection {

    public static void main(String[] args) {

        Connection conn = null;

        try {

            String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
            String user = "sa";
            String pass = "secret";
            conn = DriverManager.getConnection(dbURL, user, pass);
            if (conn != null) {
```

```

DatabaseMetaData dm = (DatabaseMetaData) conn.getMetaData();
System.out.println("Driver name: " + dm.getDriverName());
System.out.println("Driver version: " + dm.getDriverVersion());
System.out.println("Product name: " + dm.getDatabaseProductName());
System.out.println("Product version: " + dm.getDatabaseProductVersion());
}

} catch (SQLException ex) {
    ex.printStackTrace();
} finally {
    try {
        if (conn != null && !conn.isClosed()) {
            conn.close();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}
}
}
}
}

```

...هذا عن الاتصال بقاعدة البيانات.... أما من حيث التعامل معها كإضافة و الحذف و التعديل و التنقل بين السجلات فهي نفس فكرة الدرس التالي

الباب الرابع : برمجة تطبيقات الجوال

الفصل الاول : إنشاء المشاريع Building MIDlets

أهلاً وسهلاً أخواني الكرام و مرحباً بكم في منتدى مصر التقنى ...مازلنا مع دروة الجافا الرسومية و قد تحدثنا في أول درس من دروس تلك الدورة ان هناك ثلاث إصدارات الجافا وهي:

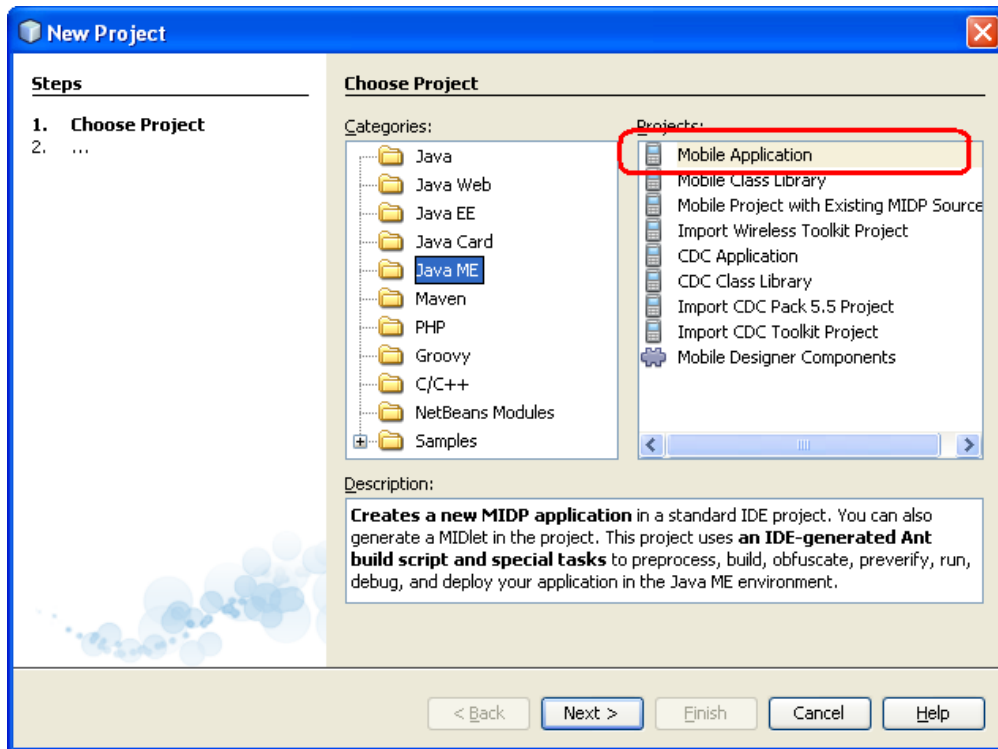
- اصدار الإنترنت برابس Java Enterprise Edition وهو الخاص بتطبيقات الويب و يختصر الى java EE
- الاصدار القياسي Java Standard Edition وهي الخاصة بتطبيقات سطح المكتب وهي التي درسناها من بداية الدرس الثاني و حتى نهاية الدرس السابع عشر و تختصر الى java se
- اصدار المايكرو Java Micro Edition و يختصر الى java me وقلنا انه هذا الاصدار خاص ببرمجة تطبيقات الجوال والأجهزة المحدودة في المصادر و هو الذي سنقوم بدراسته ابتداء من هذا الدرس إن شاء الله تعالى
- وقبل أن نبدأ ببرمجة تطبيقات الجوال باستخدام java me أحب أن انوه أن الاساسيات التي درسناها في java se من بداية الدرس الثاني وحتى الدرس السابع هي نفسها التي سيتم استخدامها في تلك التطبيقات ...مثل الاعلان عن الثوابت و المتغيرات و انشاء المصفوفات و الجملة الشرطية وحلقات التكرار ستكون بنفس القواعد التي درسناها في JAVA SE

اولاً : مفهوم MIDlet

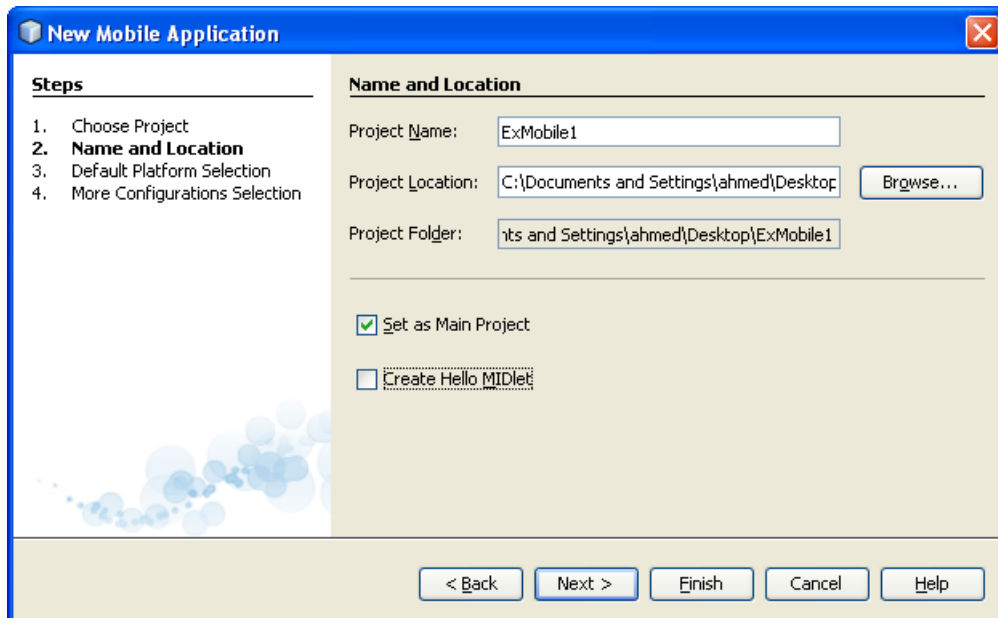
تسمى المشاريع القياسية لتطبيقات الجوال MIDlet وهي اختصار للجملة (Mobile Information Device Profile) وهي تطبيقات تستخدم معلومات الملف الشخصي للجهاز المحمول من مواصفات منصة java me التي تحتوى على المكتبات اللازمة لدعم برمجيات الجافا على الجوال ... تلك المنصة تسمى (CLDC) وهي اختصار للجملة الانجليزية... Connected Limited Device Configuration

ثانياً : خطوات إنشاء تطبيقات الجوال

1_ قم بفتح... NetBeans بعد ذلك قم بالنقر على ايقونة New Project ليتم فتح معالج إنشاء مشروع جديد من هذا المعالج ومن التويب (category) أختر JAVA ME ومن التويب project أختر الامر Mobile Application كما في الصورة التالية:

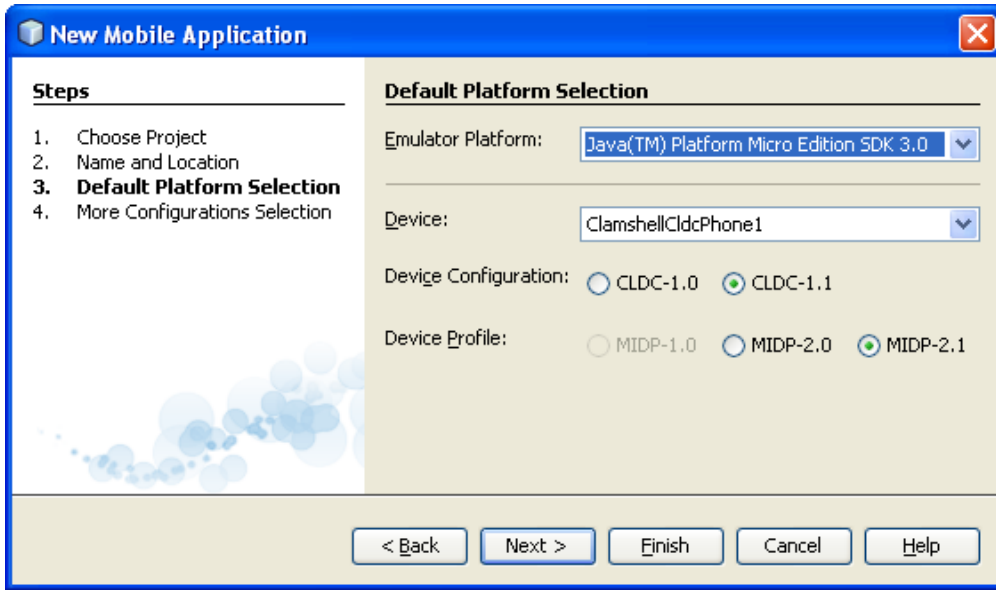


2_ بعد النقر على زر (NEXT) يظهر لك المربع الحوارى التالى:



وكما هو ملاحظ يتطلب المربع الحوارى السابق... اسم للمشروع و مسار حفظ المشروع بالإضافة الى ذلك يوجد خيارين... الاول set as main project وذلك لجعل مشروعك رئيسى فى حالة وجود عدة مشاريع اخرى... الخيار الثانى Create Hello MIDlet وهذا لانشاء تطبيق MIDlet جاهز باسم... Hello لذلك سنقوم بالغاء تحديد هذا الخيار و ذلك لانشاء التطبيق بانفسنا ليتثنى لنا معرفة خطوات انشاء مشروع قياسي MIDlet

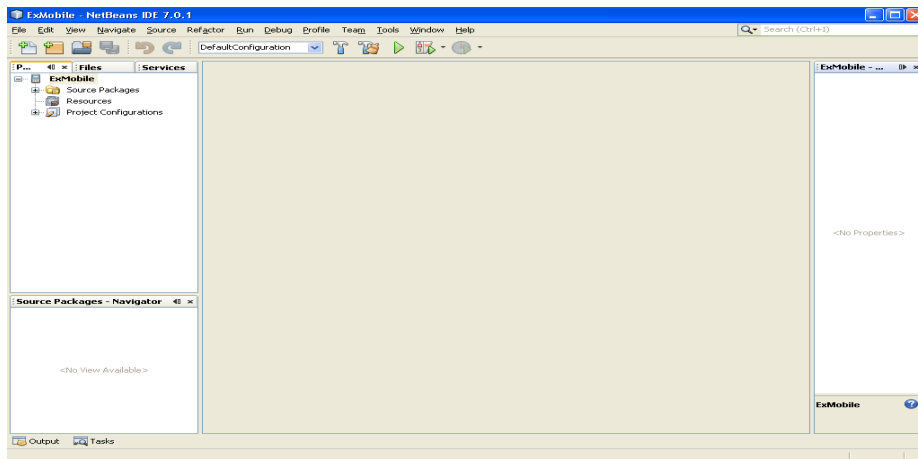
3_ بعد النقر على زر الامر (next) يظهر المربع Default Platform Selection و ذلك لتحديد المعلومات الافتراضية اللازمة لإنشاء تطبيق الجوال وهو كما فى الصورة التالية:



وكما هو ظاهر فى الصورة السابقة يتطلب المربع الحوارى المعلومات التالية:

- **device_** وهو لاختيار المحاكى الافتراضى الذى سوف تستخدمه لتشغيل عليه تطبيقاتك قبل نقلها الى الجوال.... ويوجد العديد من المحاكى و التى تختلف عن بعضها من حيث) شاشة احادية اللون او شاشة عديدة الالوان وموضع الازرار فى لوحة المفاتيح.... الخ. إذا كنت ترغب فى اختبار التطبيق فى بيئة معينة (على جهاز محاكى محددة) قد توفير الموارد المتاحة للاستخدام من قبل الشركة المصنعة؛ على سبيل المثال لأجهزة نوكيا، توفر لك device من أجيال مختلفة ونماذج هي forum.nokia.com
- **Device Configuration_** وهو الخاص بمعلومات تكوين الجهاز ولحد الان توجد اصدارين التى تدعمها الجوالات هما CLDC-1.0 و.... CLDC-1.1 سنقوم بأختيار الاصدار الاول فهو الاكثر انتشارا فى اجهزة الجوال
- **device profile_** وهو معلومات الملف الشخصى لاجهزة الجوال وهو أمتداد CLDC و سنتختار الاصدار MIDP-2.0 و السبب فى ذلك أن حوالى ٧٥ % من أجهزة الجوال التى صنعت بعد عام ٢٠٠٥ تدعم هذا الاصدار ...

4_ بعد الانتهاء من تحديد كافة المعلومات اللازمة يمكنك النقر على زر الامر (finsh) ليظهر لك بنية netbeans كما فى الشكل التالى :



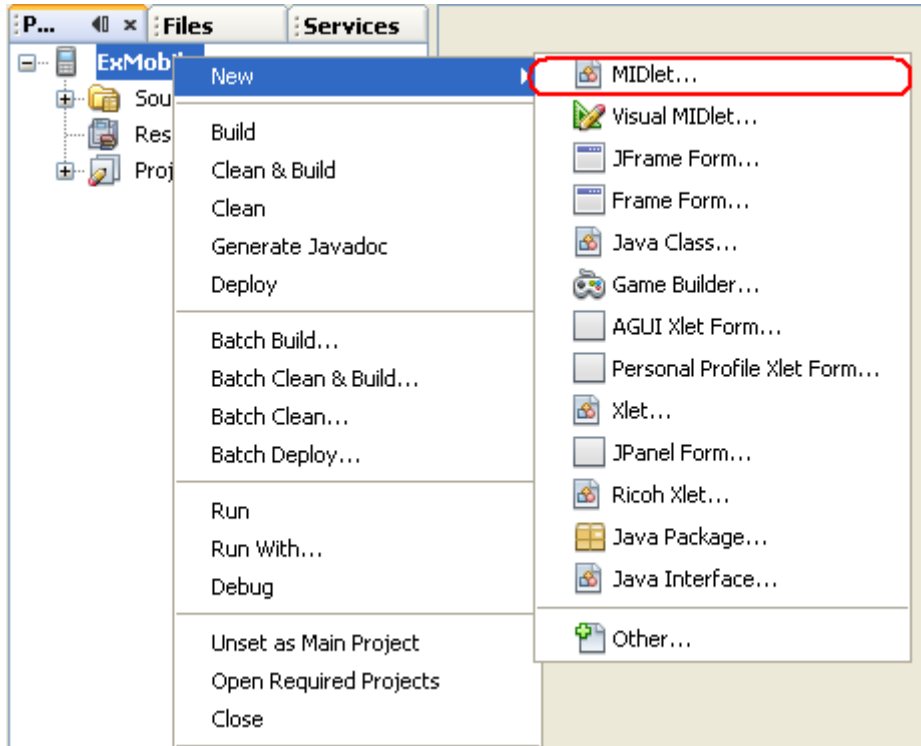
5_ كما هو ظاهر فى الصورة أعلاه حتى الان لم يتم الانتهاء من إنشاء المشروع....ولكى يتم اكتمال إنشاء المشروع يجب علينا تحديد السياق التي على أساسها سوف نبني التطبيق. وهذا يعني إدراج مورد أو ملف من نوع MIDlet ويوجد طريقتين لإدراج هذا الملف وهما:

الطريقة الاولى: وهو ادراج المورد أو الملف MIDlet فى بيئه العمل غير رسومية و التي تتيح لنا التعامل مع التطبيق من خلال الشفرة فقط

الطريقة الثانية: وهو ادراج المورد أو الملف MIDlet فى بيئة العمل الرسومية و التي تتيح لنا إنشاء التطبيق من خلال العناصر الرسومية كإضافه النوافذ و القوائم عن طريق السحب و الافلات

بناء المشاريع فى بيئة العمل غير الرسومية

تعالوا الان نتعرف على **الطريقة الاولى** أولا وبعد ذلك سنتعامل مع البيئة الرسومية باقى دروس الدورة.....لكى نستطيع إدراج المورد أو الملف MIDlet فى البيئة غير الرسومية نقوم بتحديد ايقونة التطبيق ثم نكليك يمين ومن القائمة التي ستظهر أختار الأمر New ثم الأمر MIDlet كما فى الصورة التالية:



سيظهر لك المربع الحوارى التالى و الذى يطالبك ببعض المعلومات كأسم الملف MIDlet و اسم الكلاس الخاص بMIDlet ويمكنك تغير البيانات الافتراضية أو تركها كما هى إن شئت ثم قم بالنقر على زر الامر (finish) ليتم فتح صفحة كتابة الشفرة كما فى الصور التالية:

```

1
2 import javax.microedition.midlet.*;
3
4
5 public class Midlet extends MIDlet {
6
7     public void startApp() {
8     }
9
10    public void pauseApp() {
11    }
12
13    public void destroyApp(boolean unconditional) {
14    }
15 }

```

يلاحظ في الكود أعلاه أنه تم انشاء الفئة أو الكلاس Midlet المشتق من المودر Midlet وبداخل هذا الكلاس ثلاث طرق وهي:

- **startApp** وتستخدم هذه الطريقة لتشغيل التطبيق أو بمعنى أوضح نقطة البداية للتطبيق.
- **pauseApp** وتستخدم هذه الطريقة لتنفيذ وظيفة ما في حالة وقوع حدث يؤدي الى عرقلة تشغيل التطبيق مثل تلقي مكالمة هاتفية أثناء تشغيل التطبيق
- **destroyApp** هي الطريقة المستخدمة لإغلاق التطبيق؛ يتم تنفيذه في نهاية التطبيق.
-

تعالوا الان أخواني الكرام نقوم بعمل تطبيق جوال بناء على المعلومات السابقة لظهار نص مكتوب فيه "مرحبا بكم في منتدى مصر التقنى"

لعرض نص أو رسالة على شاشة الجوال لابد من استخدام الكائن Display و الذي يقوم بتهيئة المورد MIDlet في بداية التطبيق.... هذا الكائن Display هو أحد الكائنات التابعة للمكتبة LcdUI لذلك سيتم استيرادها.... وبالتالي سيكون الكود على النحو التالي :

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Midlet extends MIDlet {
private Display display = null;

public void startApp() {

if(display==null)
display = Display.getDisplay(this);
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
}

```

وكما تلاحظون في الكود السابق ... ففي السطر الثاني قمنا باستيراد المكتبة... LcdUI وفي السطر الرابع قمنا بإنشاء الكائن Display في طريقة startApp استخدمنا جملة شرطية للتأكد أن الكائن display يساوي null فإن كان الأمر كذلك يتم تهيئة المورد من خلال الأمر getDisplay(this) وكلمة this تعني هذا المورد

الكود السابق حتى الآن ليس كافياً لعرض الرسالة وذلك لأننا نحتاج إلى وعاء للرسالة وليكن مربع نص وهي أحد العناصر الرسومية للجافا لتطبيقات الجافا وهي تختلف عن مربعات النصوص المستخدمة في تطبيقات سطح المكتب.... ويتم إضافة مربع النص باستخدام الكود التالي :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Midlet extends MIDlet {
    private Display display = null;
    private TextBox textmass;

    public void startApp() {

        if(display==null)
            display = Display.getDisplay(this);
        textmass=new TextBox("
        (أول تطبيق جوال"، "مرحبا بكم في منتدى مصر التقني" ١٠٠٠٠٠);
        display.setCurrent(textmass);

    }

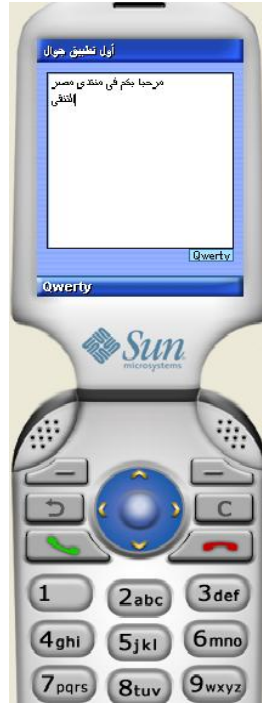
    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

وكما تلاحظون الأكواد الجديدة في الكود السابق عما قبله ... ففي السطر الخامس قمنا بإعلان عن مربع نص وتم تسميته textmass وفي السطر التاسع قمنا اشتقاق كائن جديد من مربع النص و إسناده إلى textmass مع وضع البيانات اللازمة لإظهار الرسالة ... أما في السطر العاشر قمنا باستخدام الطريقة setCurrent للكائن display لتشغيل و إظهار مربع النص الآن قم بالنقر على زر التشغيل في شريط أدوات netbeans



ليظهر لك المحاكى وبه رسالت الترحيب كما في الصورة التالية :



و يمكنك ايضا النقر على لوحة مفاتيح جهاز المحاكى على زر (رجوع) لترى ايقونة تطبيقك كما فى الشكل التالى



رغم بساطة التطبيق فهو قد فهمنا من خلاله كيفية إنشاء تطبيق جوال من خلال البنية غير الرسومية وهكذا نكون قد أنهينا من هذه الطريقة فى انشاء المشاريع و الى اللقاء إن شاء الله تعالى فى شرح الطريقة الأخرى الرسومية لإنشاء تطبيقات الجافا و ايضا دورس أخرى إن شاء الله تعالى

بناء المشاريع فى بيئة العمل الرسومية

إخوانى الكرام أهلا وسهلا بكم ... فى الدرس السابق تحدثنا عن كيفية إنشاء تطبيقات الجوال MIDlet فى البيئة غير الرسومية وقمنا بعمل تطبيق بسيط باستخدام الشفرة فقط لعرض رسالة ترحيب ... فى هذا الدرس إن شاء الله تعالى سنتعلم سويا كيفية إنشاء تلك التطبيقات فى البيئة الرسومية و ذلك عن النحو التالى:

أولا : خطوات إنشاء تطبيقات الجوال فى البيئة الرسومية

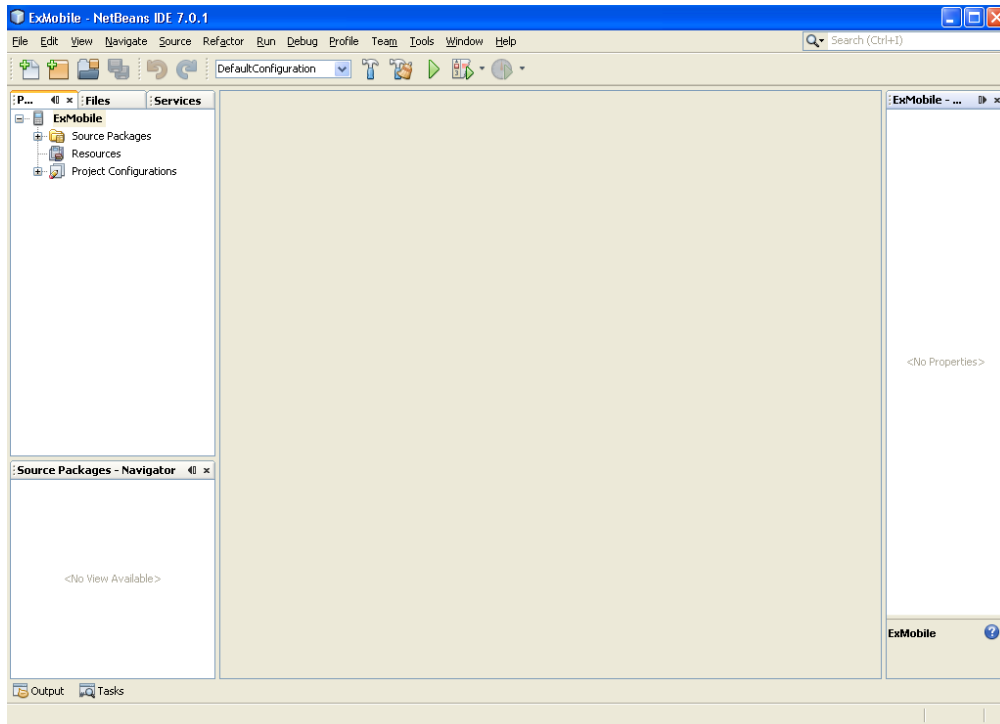
1_ قم بفتح NetBeans ... بعد ذلك قم بالنقر على ايقونة New Project ليتم فتح معالج إنشاء مشروع جديد ... من هذا المعالج ومن التويب (category) أختَر JAVA ME ومن التويب project أختَر الامر Mobile Application ثم قم بالنقر على زر الامر Next

2_ سيظهر لك مربع حوارى يطالبك باسم للمشروع و مسار حفظ المشروع بالإضافة الى ذلك يوجد خيارين ... الاول set as main project و ذلك لجعل مشروعك رئيسى فى حالة وجود عدة مشاريع اخرى ... الخيار الثانى Create Hello MIDlet وهذا لإنشاء تطبيق MIDlet جاهز باسم ... Hello لذلك سنقوم بإلغاء تحديد هذا الخيار و ذلك لإنشاء التطبيق بانفسا ثم قم بالنقر على زر الامر Next

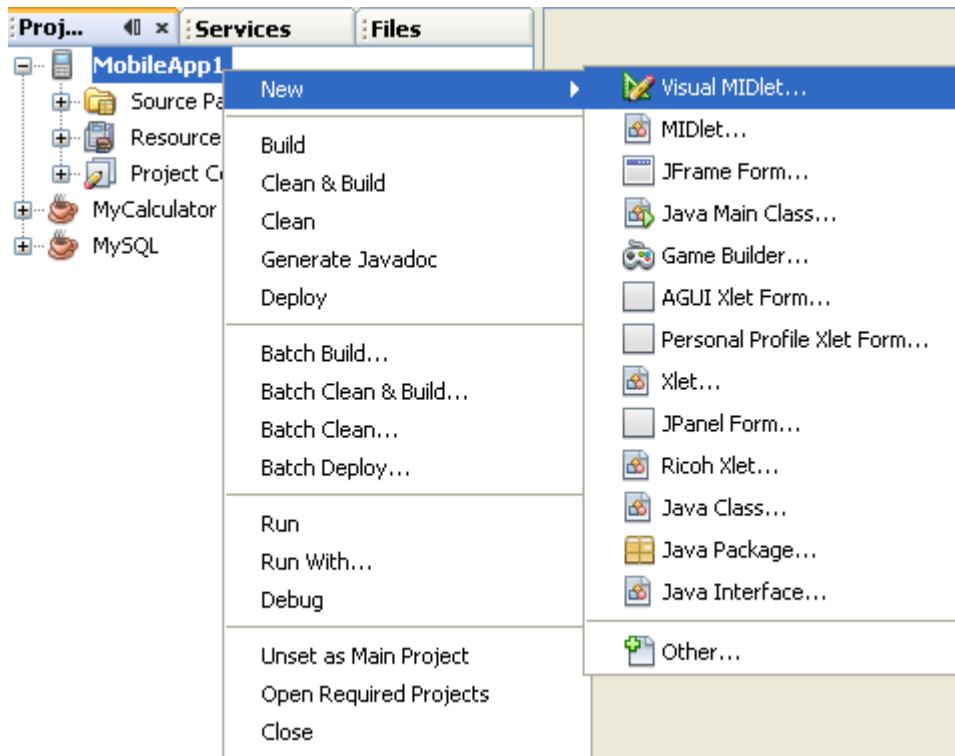
3_ بعد النقر على زر الامر Next يظهر المربع Default Platform Selection و ذلك لتحديد المعلومات الافتراضية اللازمة لإنشاء تطبيق الجوال و التى قد تحدثنا عنها فى الدرس السابق مثل

device , Device Configuration , device profile

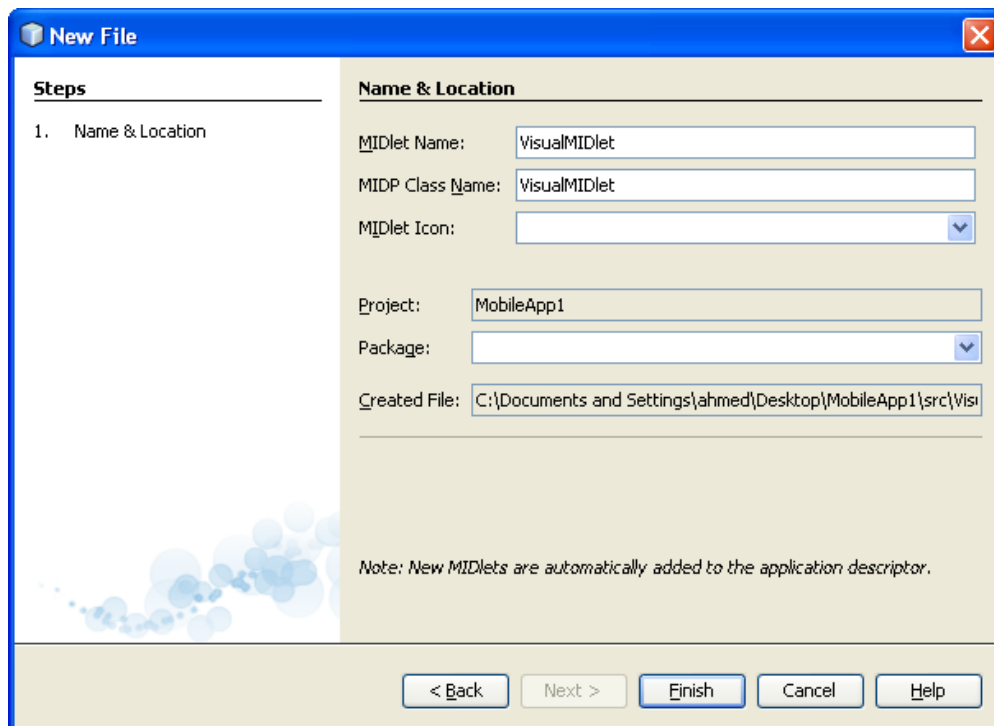
4_ بعد الانتهاء من تحديد كافة المعلومات اللازمة يمكنك النقر على زر الامر (finish) ليظهر لك بنية netbeans كما فى الشكل التالى:



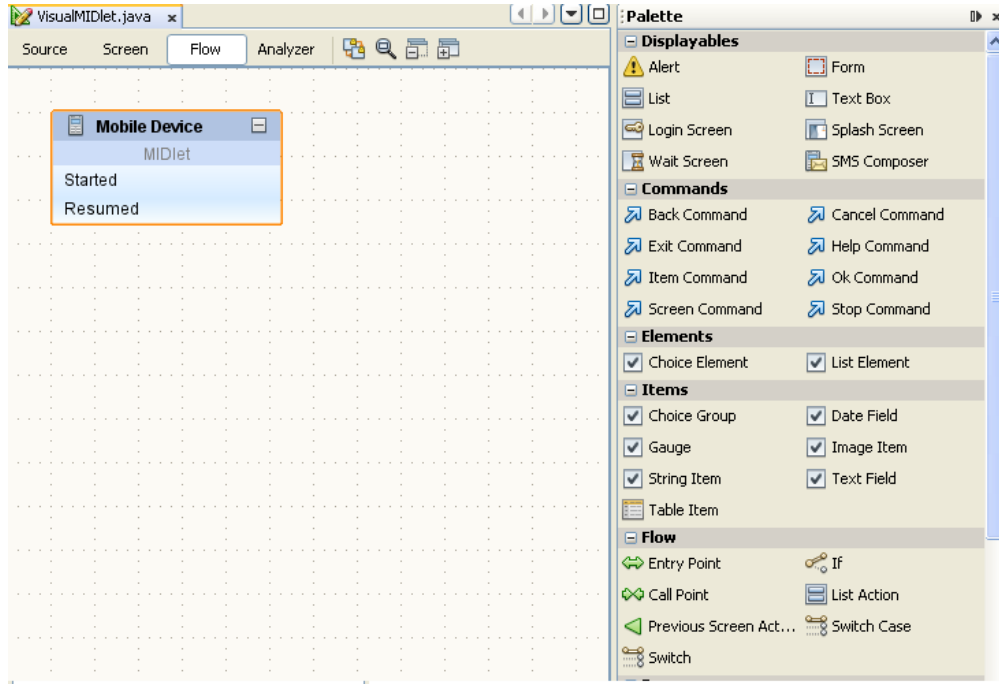
5_ نقوم بإدراج المورد أو الملف MIDlet فى البيئة الرسومية و ذلك عن طريق تحديد ايقونة التطبيق ثم كليك يمين ومن القائمة التى ستظهر أختَر الامر New ثم الامر Visual MIDlet كما فى الصورة التالية:



6_ سيظهر لك المربع الحوارى التالى و الذى يطالبك ببعض المعلومات كأسم الملف MIDlet و اسم الكلاس الخاص ب MIDlet ويمكنك تغيير البيانات الافتراضية أو تركها كما هى إن شئت كما فى الصورة التالية:

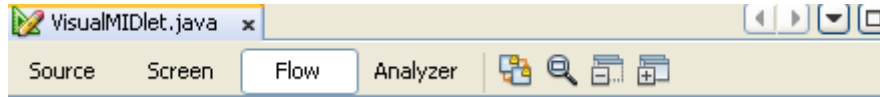


7_ بعد الانتهاء من تحديد كافة المعلومات اللازمة للمربع الحوارى السابق قم بالنقر على زر الامر (next) لتظهر لك البنية الرسومية كما فى الشكل التالى:



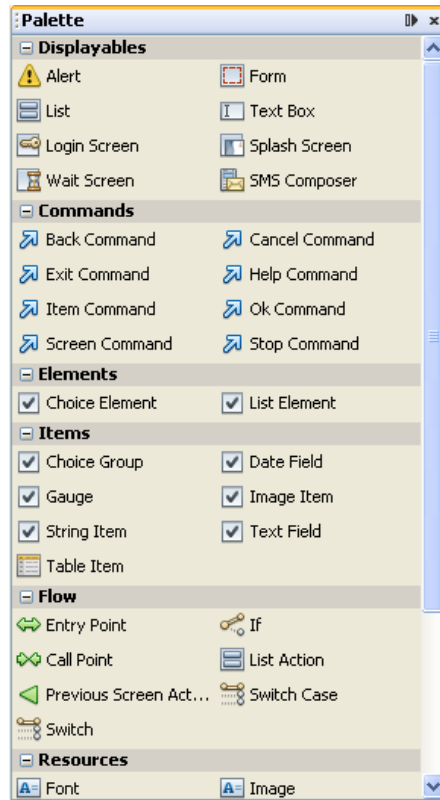
وكما ترى أخى الكريم فى الصورة أعلاه ستجد ملف MIDlet كمخطط بياني فى شاشة Flow و تعالوا الان الاخوة الافاضل نتعرف على تلك البنية الرسومية:

أولا : يحتوى شريط التويب العلوى الذى فى الصورة التالية على اربعة خيارات للتعامل مع التطبيق وهى



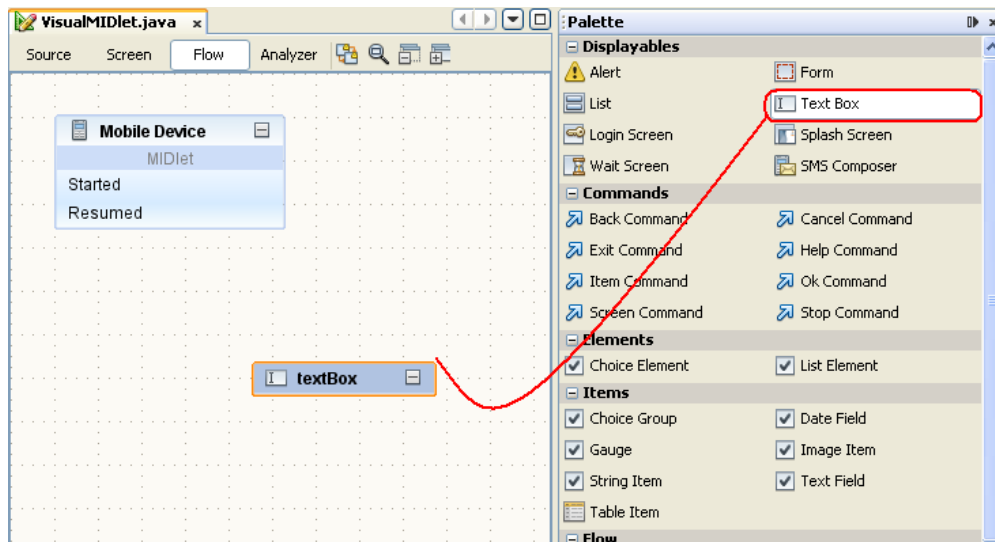
- **Source** وهذا التويب عند النقر عليه يفتح صفحة كتابة الشفرة و ذلك لكتابه اى كود او شفرة تريد تنفيذها
- **Screen** وهذا التويب عند النقر عليه يفتح واجهة رسومية تحاكي وجهة الجوال لتستطيع بعدها إدراج العناصر الرسومية عليها عن طريق السحب و الافلات.
- **Flow** وهذا التويب عند النقر عليه يفتح صفحة كمخطط للعناصر الرسومية و اوامر الربط بينهما و ايضا يتح لنا ادراج الكائنات التابعة للفئة Displayable للتطبيق عن طريق السحب و الافلات
- **analyzer** وهذا التويب عند النقر عليه يفتح صفحة تحليل لجميع العناصر و الموارد و الملفات التى تم إدراجها للتطبيق

ثانيا : فى الجهة اليمنى ستجد القائمة palette و التى تحتوى على الكثير من المكونات الرسومية و أوامر التنفيذ command التى يمكنك استخدامها فى تطبيقك منها ما ينتمى الى الفئة Displayable ومنها ما ينتمى الى الفئة Item كما فى الصورة التالية:



و سيتم شرح تلك المكونات كلا على حد في دروس متقدمة.... الان وبعدها ما تعرفنا على تلك البنية الرسومية
تعالوا نقوم بإنشاء نفس التطبيق السابق الذي تم شرحه في البنية غير الرسومية وهو عبارة عن تطبيق لظهور
رسالة ترحيب.... ويتم ذلك عن طريق الخطوات التالية:

1_ قم بإدراج المكون textbox الى المشروع و ذلك عن طريق سحب المكون من القائمة palette ثم إفلاته
الى المشروع من خلال التويب Flow كما في الصورة التالية:

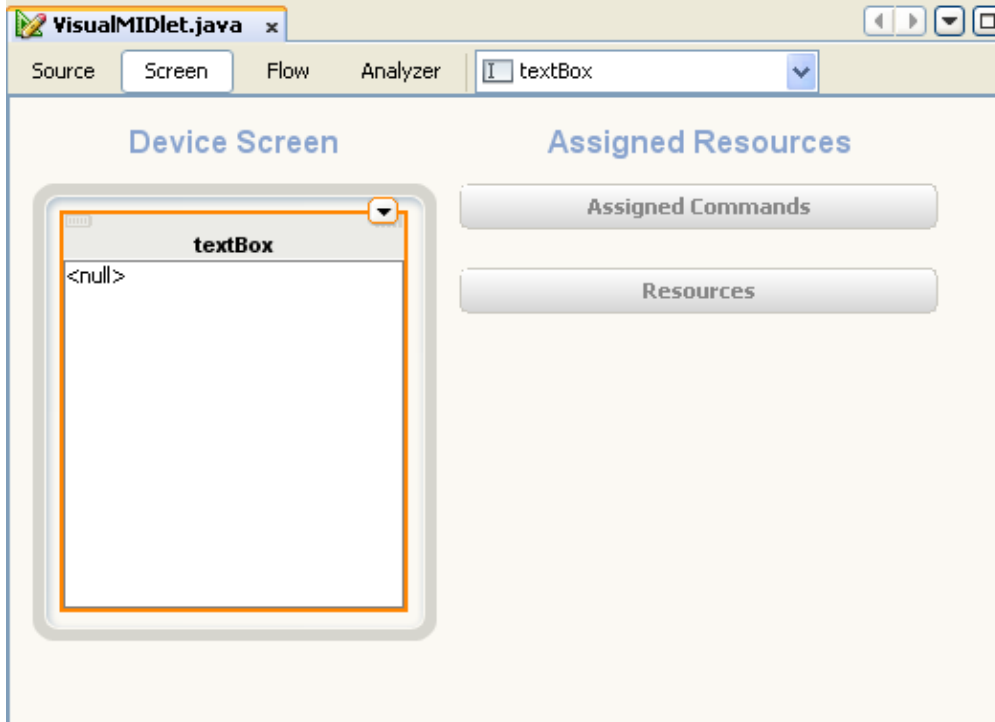


ملحوظة/ جميع الكائنات التي تنتمي للفئة Displayable لا يمكن إدراجها للمشروع عن طريق السحب و الافلات إلا من خلال التويب Flow

2_ قم بتحديد المكون في التويب ثم كليك يمين ومن القائمة التي ستظهر أختار الامر propertites لتظهر لك نافذة الخصائص كما في الشكل التالي:



ومن نافذة قمنا بكتابة رسالة الترحيب المراد عرضها من خلال الخاصية text وايضا قمنا بوضع عنوان للرسالة من خلال الخاصية Titel بعد ذلك قمنا بإغلاق نافذة الخصائص..... ويمكنك فعل كل ذلك من خلال التويب Screan والذي يظهر لك العنصر الرسومي على محاكي الجوال بالشكل التالي:



يقم النقر المزدوج على عنوان المكون textBox الموجود اعلى العنصر و الذي يأخذ القيمة textbox لتسطيع

كتابة عنوان للعنصر وبعد ما يتم الكتابة قم بالنقر على زر الادخال (Enter) كما في الصورة التالية:

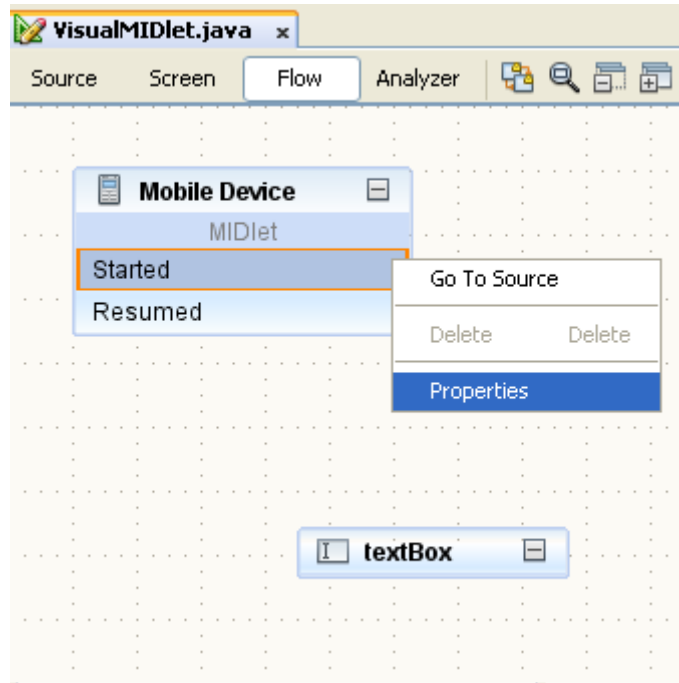


ايضا قم بالنقر المزدوج على القيمة <u> واذلك لاضافة نص الرسالة "مرحبا بكم في منتدى مصر التقنى" كما في الصورة التالية:

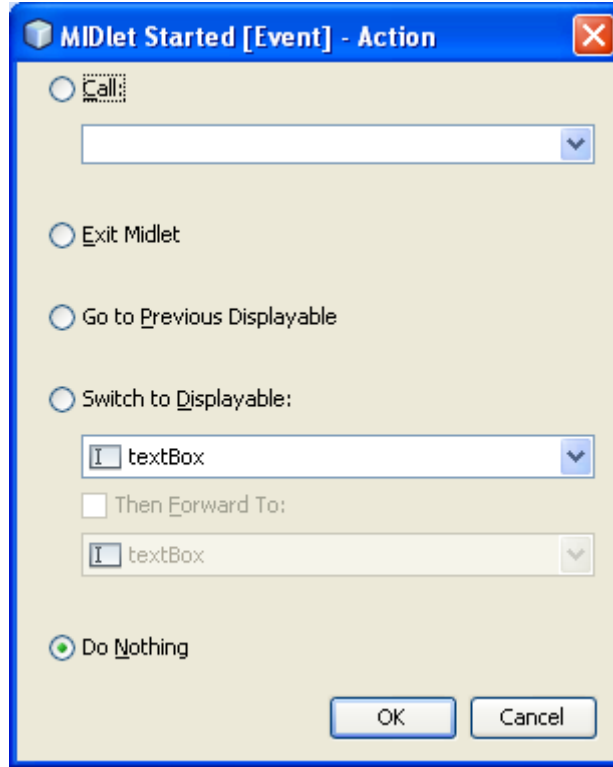


3_قم بربط المكون textbox بنقطة البداية للمورد او الملف MIDlet حتى يتم تشغيل التطبيق ويتم ذلك عن طريق الاوامر command ويتم ذلك عن طريقين:

الطريقة الاولى: النقر على التويب Flow ثم تحديد الحدث Started للمورد MIDlet ثم النقر يمين ومن القائمة التي ستظهر اختر الامر properties لتظهر لك النافذة MIDlet Started Event كما في الصور التالية:



ويمكنك ملاحظة أن نافذة الخصائص تلك لا تحتوي إلا على خاصية واحدة وهي... Action قم بالنقر على الزر الصغير الذي بجوارها ليتم لك فتح النافذة التالية:



النافذة السابقة تحتوي على ٥ خيارات للاحداث المتاحة وهي على النحو التالي:

1_الخيار call هذا الخيار يتم استخدامه في حاله اذا اردنا الاتصال بأحد ملفات MIDlet الاخرى الموجودة بالتطبيق عند بداية التشغيل للتطبيق

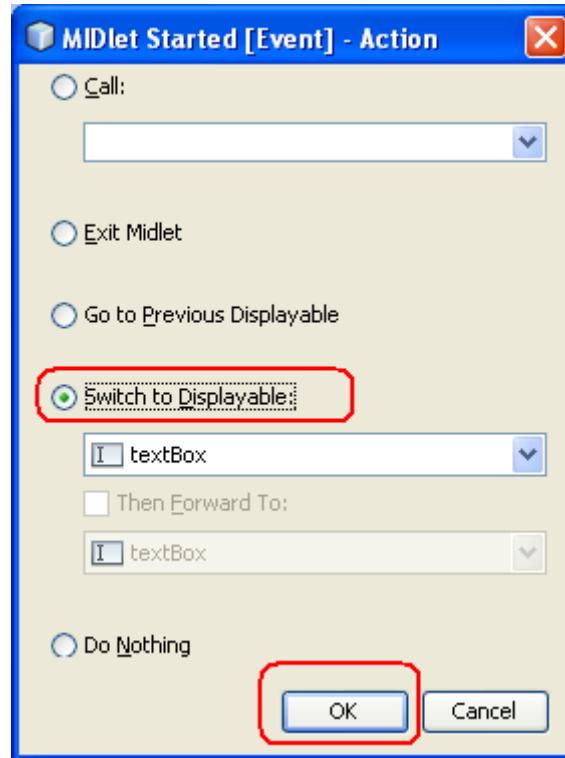
2_الخيار Eext MIDlet هذا الخيار يتم استخدامه اذا اردنا الخروج من MIDlet الحالي.

3_الخيار Go to previous Displayable هذا الخيار يتم استخدامه في حاله وجود عده مكونات تابعة للفئة Displayable وتريد الرجوع للخلف الى المكون السابق

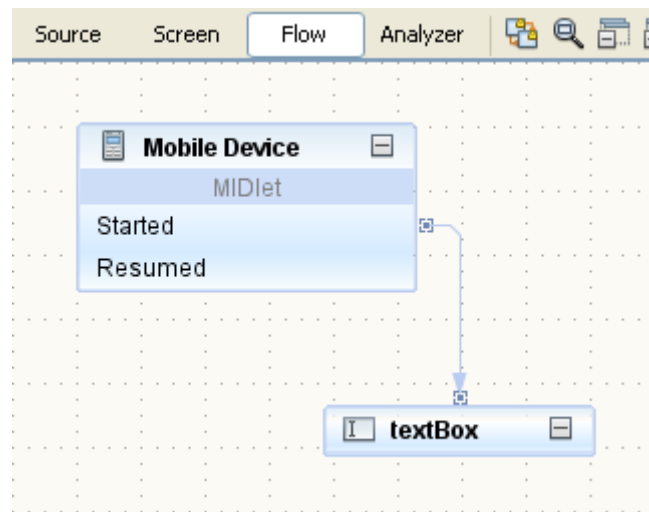
4_الخيار Switch to Displayable وهذا الخيار يتم استخدامه في حاله اننا نريد تشغيل احد مكونات الفئة Displayable وتوجد قائمة بها جميع المكونات التي أضفتها لتطبيقك.

5_الخيار Do Nothing وهذا الخيار لعدم فعل اي حدث.

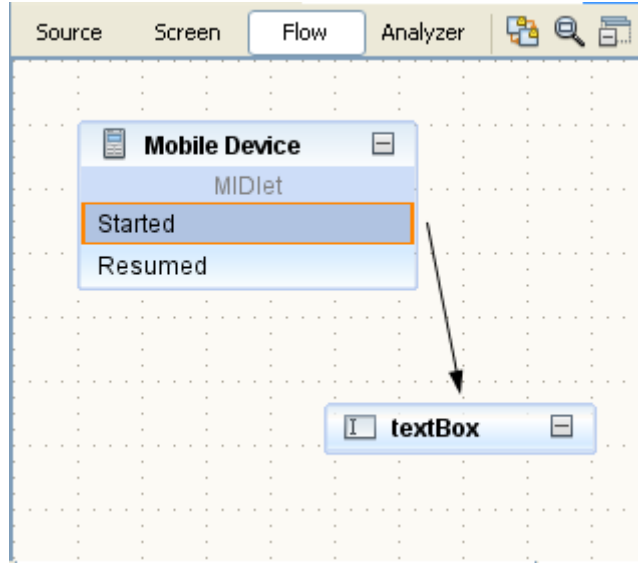
بعد ما تعرفنا على تلك النافذة نريد الان تشغيل المكون textbox بربطة بنقطة البداية Started للمورد MIDlet و ذلك من خلال الخيار الرابع Switch to Displayable و ستجد بجانبه قائمة بها المكونات التي تم إضافتها للتطبيق و في مثالنا هذا هو مكون واحد فقط وهو textbox لذلك سنقوم بتحديد هذا الخيار ثم قم بالنقر على زر الامر (OK) كما في الصورة التالية:



بعد ذلك ستجد مخطط التطبيق في التويب Flow كما في الصورة التالية:



الطريقة الثانية: قم بتحديد الحدث Started للمورد MIDlet ثم انقر بزر الماس شمال مع السحب في وضعية النقر ليظهر لك سهم أسود أثناء السحب ثم قم بإقلاته على المكون المراد ربطه بنقطة البداية Started وفي مثالنا هذا هو المكون textBox كما في الشكل التالي:

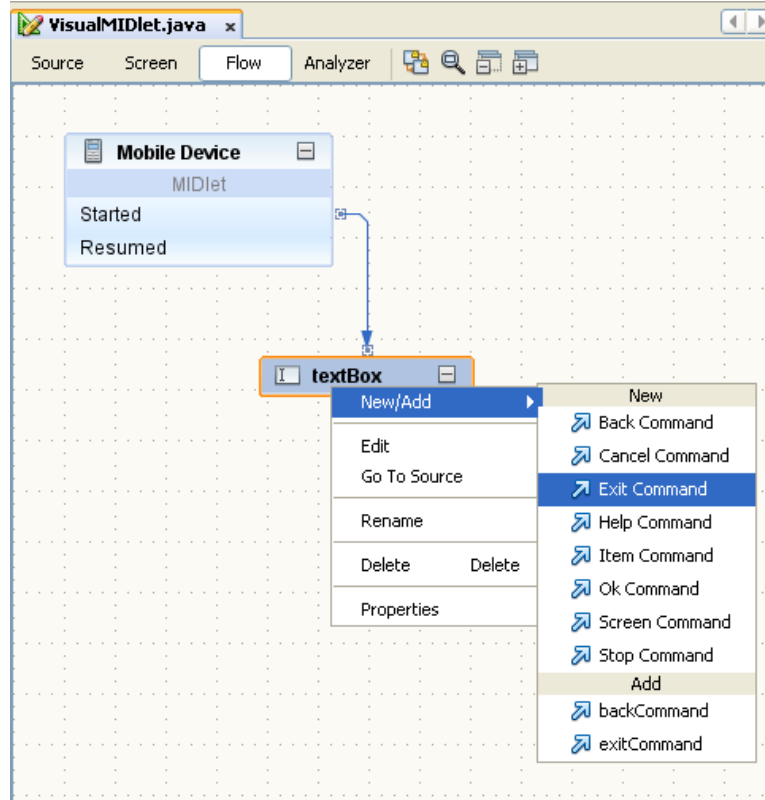


الآن يمكنك تشغيل التطبيق لعرض رسالة الترحيب و يظهر كما في الصورة التالية:

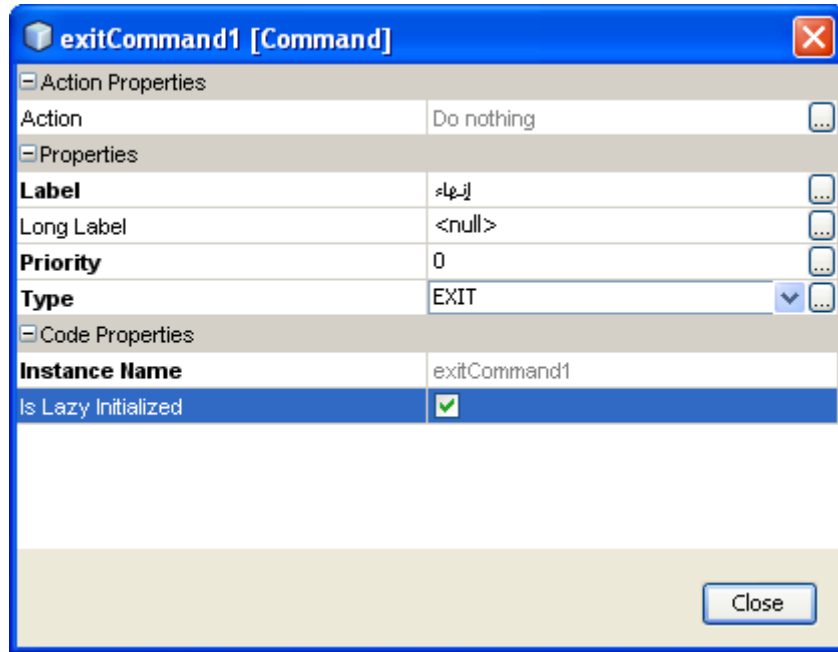


ويمكنك ملاحظة أننا قد قمنا بتشغيل التطبيق على محالكي آخر غير الذي تم استخدامه في الدرس السابق... الآن تعالوا نقوم بإضافة بعض التنسيقات للتطبيق... فمثلا نقوم بوضع أمر للخروج من التطبيق وذلك على النحو التالي:

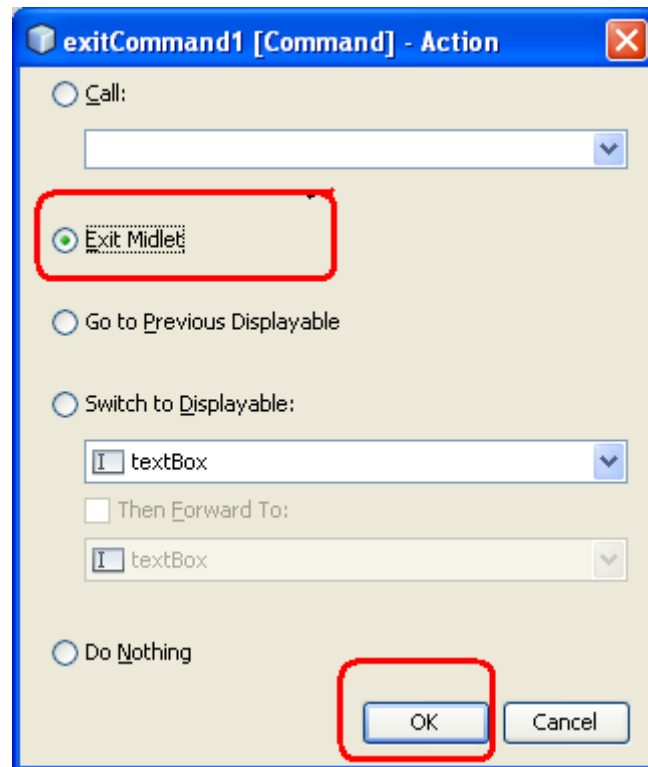
_ قم بتحديد المكون textBox من خلال التويب Flow ثم انقر يمين ومن القائمة التي ستظهر اختر الأمر New/Add ثم اختر الأمر Exit command كما في الصورة التالية:



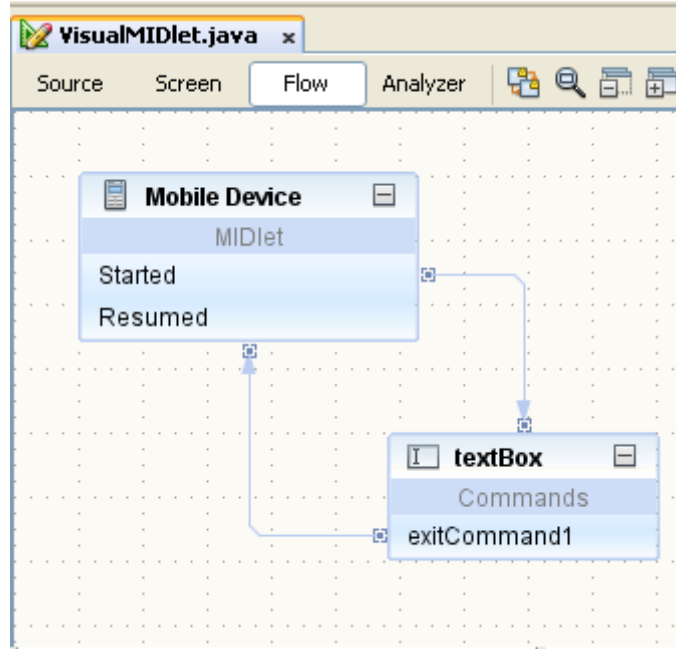
ستجد انه قد تم إضافة command للمكون textBox قم بتحديدده ثم النقر يمين ومن القائمة التي ستظهر اختر الامر properties لتظهر لك نافذة الخصائص كما في الصور التالية:



وكما هو ملاحظ في نافذة الخصائص... كتبنا كلمة إنهاء (عند الخاصية label وذلك لوضع تسميه للامر تظهر في الجوال... أما الخاصية Action قم بالنقر على الزر الذي بجوارها لتحديد خيار الخروج من MIDlet كما في الصورة التالية:



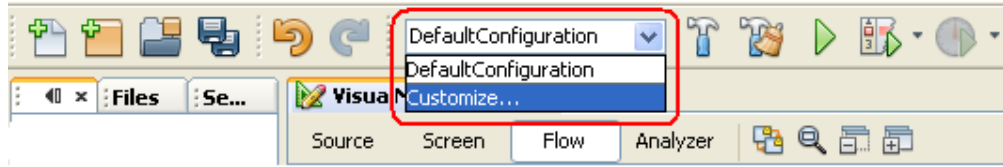
بعد النقر على زر (ok) ستجد في التويب Flow التطبيق كما في الشكل التالي:



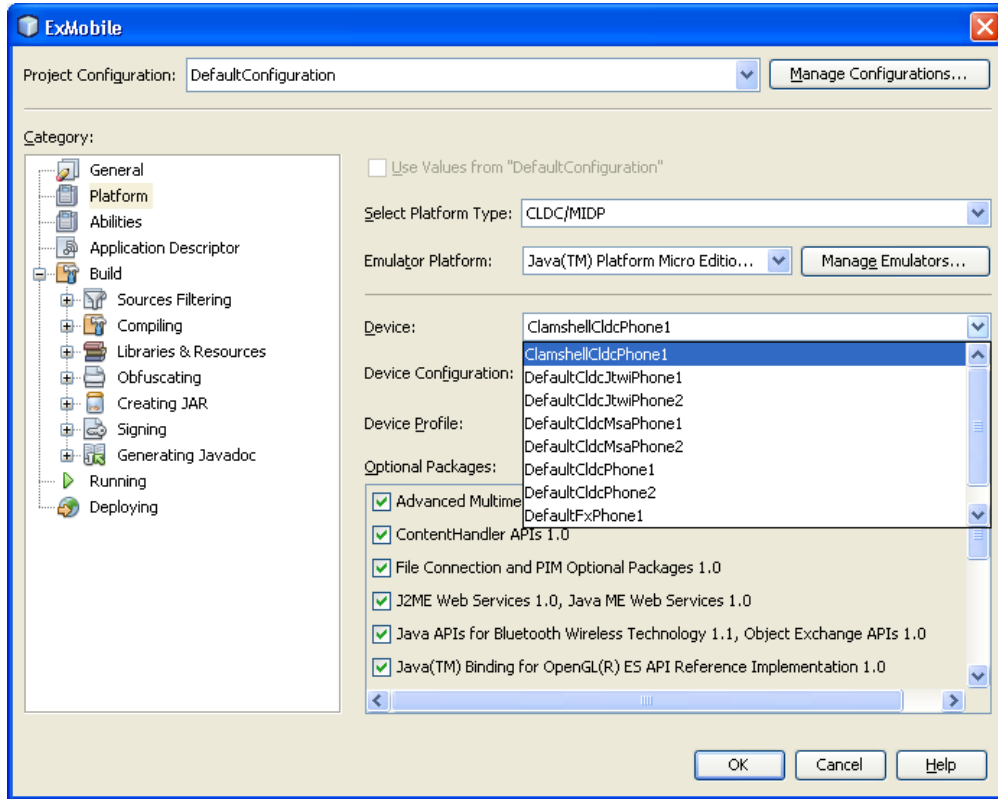
الآن قم بتشغيل التطبيق ليظهر لك كما في الشكل التالي:



ايضا يمكنك ملاحظة اننا قد استخدمنا محاكي آخر غير المستخدم بالدرس السابق و غير المستخدم في المثال السابق وهذا المحاكى الحالى يمثل أجهزة الجوال التي تعمل بخاصية اللمس وتستطيع أن تختار محاكي Device مختلف عما قد حددته في بداية إنشاء المشروع عن طريق تحديد الامر Customize من قائمة المهام في NetBeans كما في الصورة التالية:

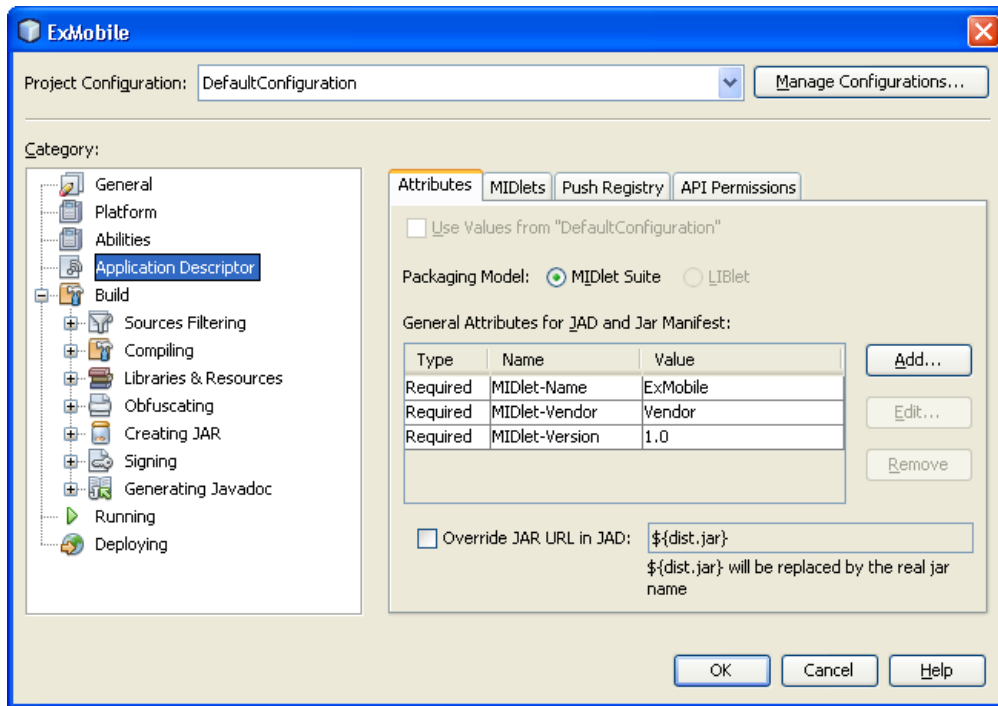


بعد ذلك سيظهر لك مربع حوارى الاول Add Configuration وهذا ليس لنا به علاقة الان لذلك قم بالنقر على زر الامر Cancel و المربع الحوارى الاخر كما فى الصورة التالية:



وكما هو ظاهر إعلاه يمكنك تحديد المحاكى الذى يناسبك من خلال القائمة Device و يجب ملاحظة أن كل محاكى يختلف عن الآخر و كلما كان المحاكى أحدث كلما كانت له مميزات كثيرة

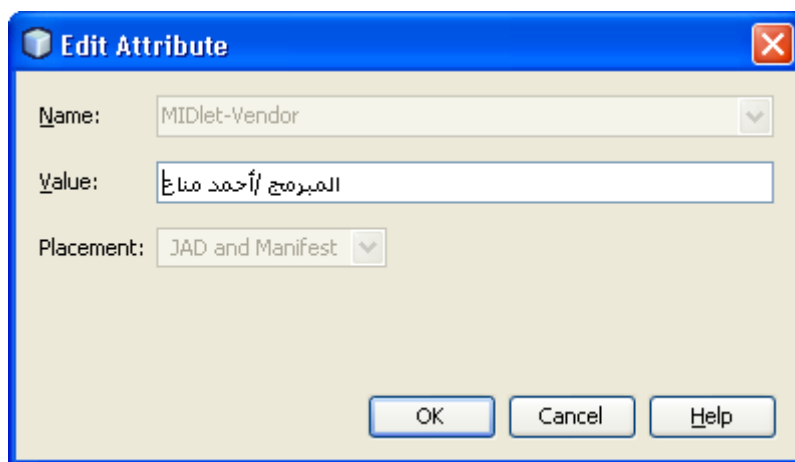
تعالوا الان نضيف بعض الخصائص الأخرى للتطبيق.... فمن نفس المربع الحوارى السابق نستطيع إضافة رقم إصدار التطبيق و معلومات عن مبرمج التطبيق وايضا إضافة ايقونة للتطبيق كل ذلك من خلال المربع الحوارى السابق التويب Application Description ليظهر لك المربع الحوارى كما فى الصورة التالية:



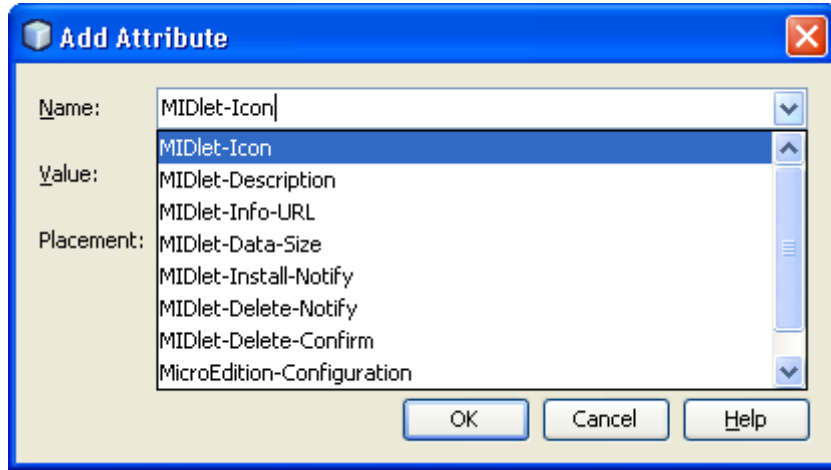
_ كما هو ظاهر أعلاه ستجد جدول The General Attributes table به بعض المعلومات الافتراضية وهي على النحو التالي:

MIDlet-Name وهذا الخيار لوضع اسم للتطبيق
MIDlet-Vendor وهذا الخيار لوضع معلومات عن مبرمج التطبيق
MIDlet-version. وهذا الخيار لوضع رقم إصدار للتطبيق

وجميع تلك الخصائص أساسية لايمكنك حذفها ومع ذلك يمكنك تعديل القيم فقط بما يناسبك..... فقط قم بتحديد الخاصية المراد تعديلها و ستجد زر الامر (Edit) الموجود على يمين تلك الخصائص أصبح نشط لتقوم بالنقر عليه ليفتح لك مربع حوارى آخر تستطيع من خلاله تعديل البيانات لتلك الخصائص..... فأنا مثلا سأقوم بتعديل الخاصية-MIDlet Vendor لوضع اسم المبرمج لذلك سأقوم بتحديد تلك الخاصية ثم النقر على زر (Edit) ليتم فتح المربع الحوارى التالى:



_ كما يمكنك إضافة بعض الخصائص الأخرى وذلك عن طريق زر الامر Add ليفتح لك مربع حوارى يحتوى على قائمة الخصائص المتاحة الأخرى كما فى الشكل التالى:

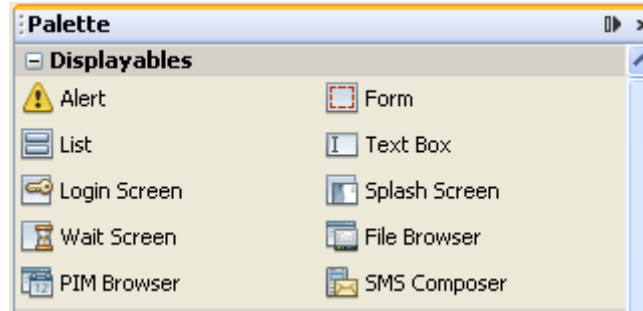


فمثلا أريد إضافة ايقونة للتطبيق لذلك سأقوم أولا بتحديد تلك الايقونة و يجب أن تكون بأمتداد BMP أو PNG و أن يكون حجمها مناسب لا تزيد عن ٦٤ بيكسل ثم وضع تلك الصورة في مجلد البرنامج src سأختار من قائمة الخصائص الخاصة MIDlet-Icon ثم أكتب أسم و أمتداد الصورة في مربع النص value كما في الشكل التالي:

الفصل الثانى : التعامل مع المكونات التابعة للتويب

Displayable الجزء الاول!

خوانى الكرام أهلا وسهلا بكم فى منتدى مصر التقنىاليوم إن شاء الله تعالى سنتعلم سويا كيفية التعامل مع المكونات التابعة للتصنيف Displayable تلك المكونات يتم وضعها على الشاشة بدون أى حاوى لها وهى المكونات التالية:



1_المكونAlert

2_المكونForm

3_المكونList

4_المكونTextBox

والمكونات الاربعة اعلاه تنتمى للحزمة javax.microedition.lcdui

5_المكونLogin Screen

6_المكونSplash Screen

7_المكونWait Screen

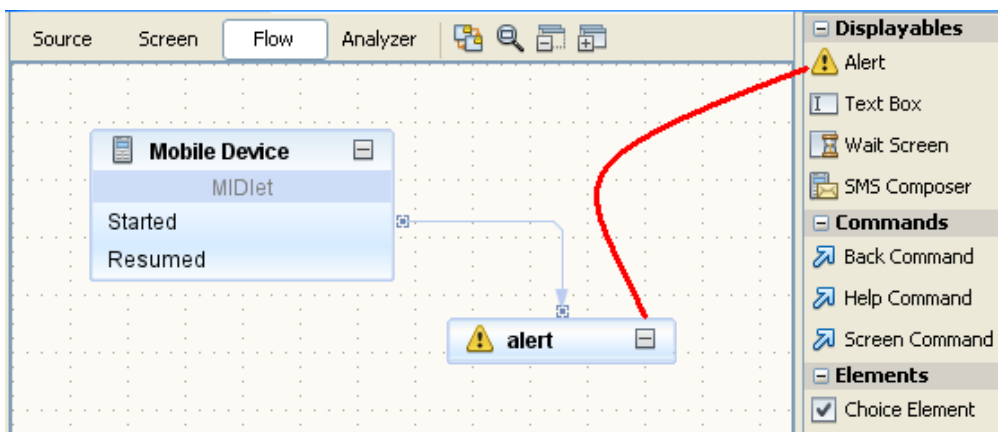
8_المكونFile Browser

9_المكونPIM Browser

10_المكونSMS Composer

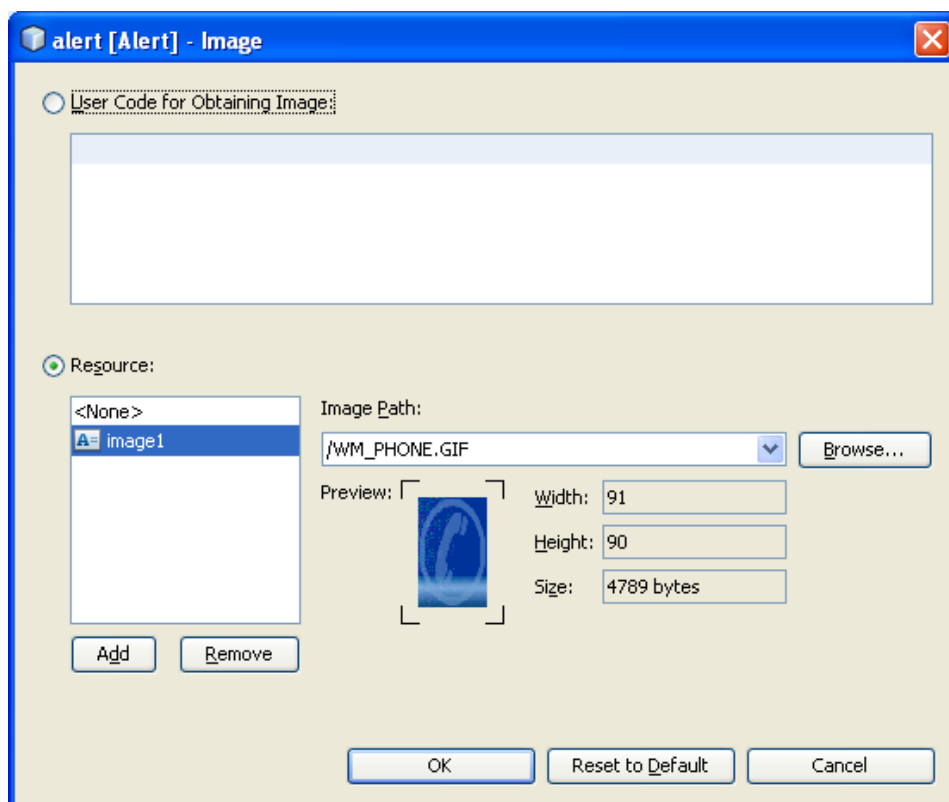
والمكونات الستة من ٥ الى ١٠ أعلاه تنتمى للحزمة org.netbeans.microedition.util وهى مكونات خاصة بـ NetBeans وبالتالى ربما لن تجدها إذا كنت تستخدم محرر آخر....تعالوا أن نتعرف على تلك المكونات

1 المكون Alert



هذا المكون يستخدم لعرض رسائل تنبيهيه أو تحذيرية أو معلومات وتنتظر لفترة معينة قبل الانتقال الى مكون آخر ... و تستطيع استخدام النصوص بهذه الرساله أو استخدام صوره معينه و أهم الخصائص لهذا المكون ما يلي:

_ الخاصية... Image و التي تتيح لك وضع صورة لرسالة التنبيه مع أو بدلا من النص ويمكنك الوصول لتلك الخاصية عن طريق نافذة الخصائص للمكون Alert ثم تحديد الخاصية Image ثم النقر على الزر الصغير الذي بجوارها لتظهر لك النافذة التالية و التي تستطيع إدراج صورة للمكون: Alert

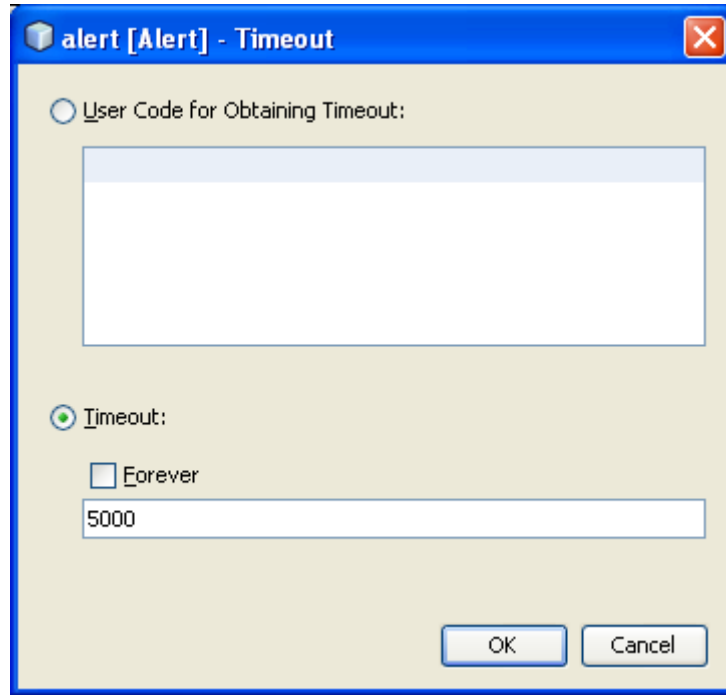


_ الخاصية... String و التي تتيح إدراج نص معين ليظهر في المكون Alert

_ الخاصية... Ticker و التي تتيح إنشاء شريط متحرك أسفل الرسالة.

_ الخاصية... Timeout و التي تتيح وضع مقدار الزمن التي تستغرقه الرسالة في ظهورها على الشاشة و تأخذ القيمة الافتراضية FOREVER و التي تعنى تبقى في الظهور الى أن يتم النقر على حدث معين لكنك بالطبع تستطيع تغييرها

ووضع مده معينه و ذلك بالنقر على الخاصية ليتم فتح المربع التالي...كل ما عليك فعله هو إزالة الخيار الموجود بجوار FOREVER ليتم تنشيط مربع الزمن وضع فيه المدة التي ترغب فيها و ليكن ٥ ثواني لذلك سنتكتب ٥٠٠٠ كما في الصورة التالية:



2_المكون.....Form وهو عبارة عن نافذة لاحتواء المكونات التي لا يمكن استخدامها إلا في وجودها مثل text fields و.... choice groups الخ ذلك من العناصر التي تطلب وجود نافذة...أما اهم خصائص هذا المكون الخاصية Title التي تتيح وضع اسم للنافذة

3_المكون.....List وهي عبارة عن قائمة تحتوي على مجموعة من الخيارات ليختار من بينها المستخدم و لكي تستطيع إضافة عناصر تلك القائمة قم بتحديد المكون ثم النقر يمين ومن القائمة التي ستظهر أختار الامر Go to Source ليتم فتح نافذة الشفر و ستجد الكود التالي:

```
public List getList () {
if (list == null) {
list = new List ("list", Choice.MULTIPLE);
list.setCommandListener (this);
list.setFitPolicy (Choice.TEXT_WRAP_OFF);
}
return list;
}
```

سنقوم بإضافة العناصر باستخدام الطريقة Append و ذلك بإضافة الكود التالي الى السابق وعلى النحو التالي:
الرمز:

```
public List getList () {
if (list == null) {

list = new List ("list", Choice.IMPLICIT);
list.setCommandListener (this);
list.setFitPolicy (Choice.TEXT_WRAP_OFF);
list.append ("العنصر الاول", null);
}
```

```

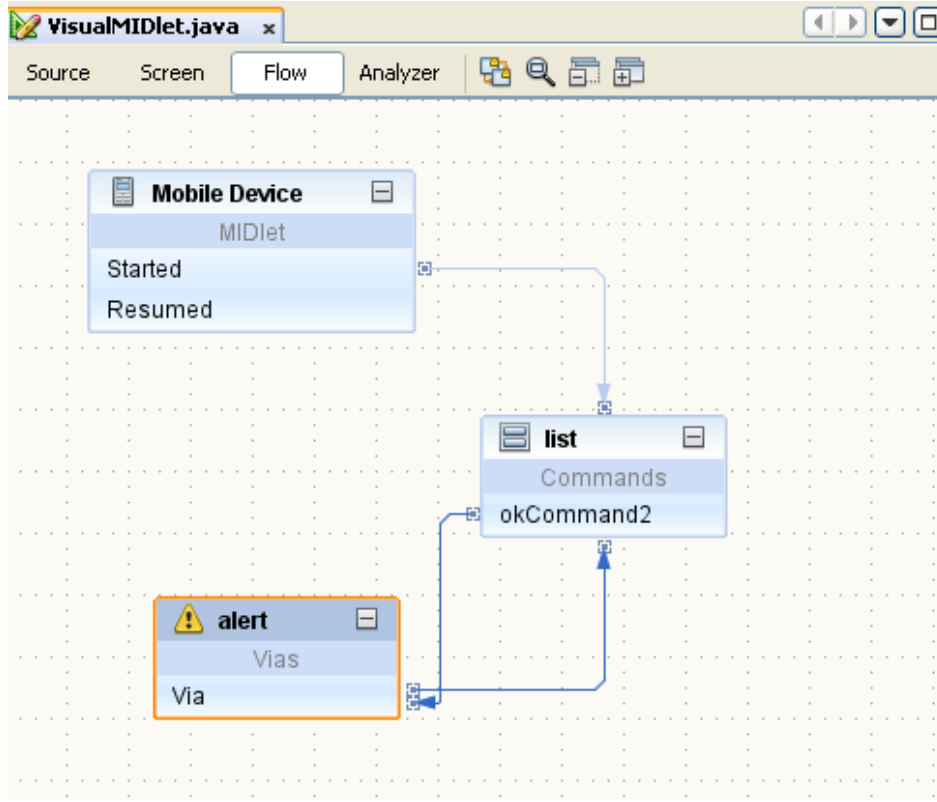
", null);list.append("
", null);list.append("
", null);list.append("
", null);list.append("
}
return list;
}

```

و النتيجة كما فى الصورة التالية:



_كيفية تحديد عنصر من القائمة ...يمكنك من تحديد العنصر المختار من قبل المستخدم عن طريق الاسلوب `getSelectedIndex()` و لتوضيح الفكرة سنقوم بادراج المكون `Alert` مع المكون `List` مع وضع `commandOK` و تغيير اسمه الى (موافق) كما فى الشكل التالى:



الآن المطلوب هو عند تحديد عنصر من القائمة ثم النقر على موافق تظهر رسالة Alert بأسم العنصر المختار.... لذلك سنقوم بتحديد المكون Alert ثم النقر يمين ليفتح لك نافذة الشفرة لتجد الكود التالي بها:

الرمز:

```
public Alert getAlert () {
if (alert == null) {

alert = new Alert ("alert");
alert.setTimeout (Alert.FOREVER);
}
return alert;
}
}
```

من جانبنا سنقوم بإضافة الكود التالي اليه:

```
int no=list.getSelectedIndex();
alert.setString(list.getString(no));
```

وكما هو ظاهر في الكود أعلاه قمنا بالإعلان عن المتغير no ليقوم بحفظ الرقم المفهرس لكل عنصر.... و في السطر الثاني النص الخاص بالرسالة alert قمنا بالحصول على اسم العنصر من خلال الرقم المفهرس بعد الحصول اسم العنصر المقابل لهذا للرقم المفهرس باستخدام الطريقة getString



_ومن أهم خصائص المكون list الخاصة Type و التي تشمل ثلاث خياراتوهي الخيار IMPLICIT و التي تقوم بعرض العناصر كما في الامثلة السابقة ...و الخيار EXCLUSIVE و التي تقوم بعرض العناصر على خانات اختيار كما في الشكل التالي :



أما الخيار الاخير هو MULTIPLE و الذى يقوم بعرض العناصر على هيئة مربعات اختيار متعدد كما فى الشكل التالى:



4_المكون TextBox هو عبارة عن شاشة تسمح للمستخدم بعرض وتحرير نص معين ويتميز بأنه يسمح بالاسطر المتعدده و أهم خاصية لهذا المكون هي الخاصية Text والتي من خلالها تستطيع تحرير النص و

ايضا الخاصية maximum size وهو الحد الاقصى لعدد الاحرف التى يمكن تخزينها فيه...

أما أهم الطرق أو الأساليب لهذا المكون هي الطريقة getString و التى يمكنك استخدامها للحصول على النص الموجود فى المكون و ايضا الطريقة setString و التى تستطيع من خلالها وضع نص معين اثناء التشغيل... ايضا الطريقة size و التى تقوم بجلب عدد الاحرف الموجودة فى مربع النص

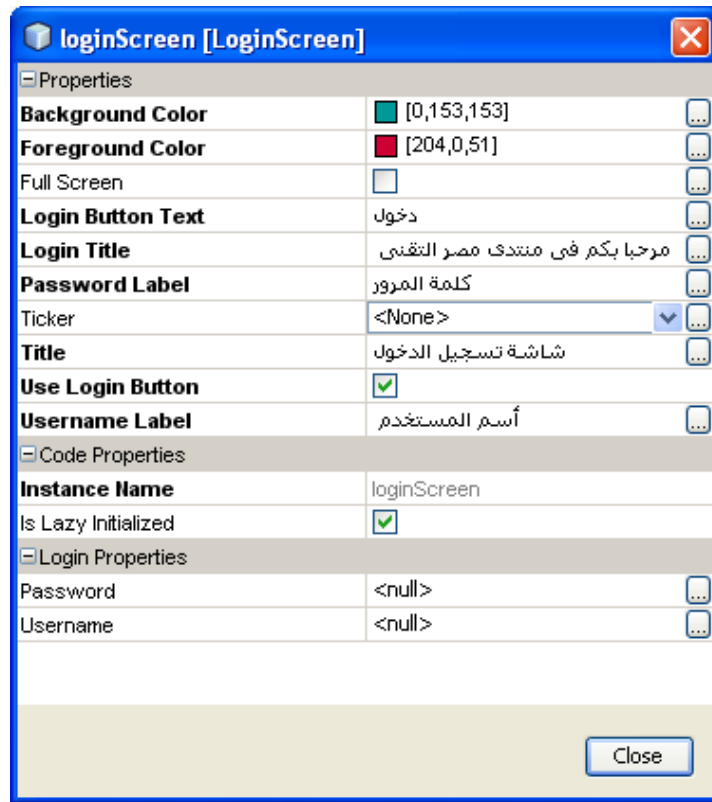
أما باقى العناصر التى تنتمى الى التويب Displayable سيتم شرحها فى درس آخر إن شاء الله تعالى فالى اللقاء

الفصل الثالث : التعامل مع المكونات التابعة للتويب Displayable الجزء الثانى!

اهلا بكم إخوانى الكرام ... فى الجزء الثانى من درس كيفية التعامل مع المكونات التابعة للتصنيف Displayable فى الجزء الاول أوضحنا أن هناك مكونات تابعة للحزمة javax.microedition.lcdui فى هذا الدرس إن شاء الله تعالى سنتعلم كيفية التعامل مع المكونات التابعة للحزمة org.netbeans.microedition.lcdui وهى على النحو التالى:

1 المكون LoginScreen هذا المكون عبارة عن شاشة تسجيل دخول للتطبيق و التى تتطلب أسم المستخدم و كلمة مرور و هذا المكون مفيد للتطبيقات البنكية لما يتطلبه سرية المعلومات.

ومن أهم الخصائص التى يتميز بها هذا المكون ما يلى:

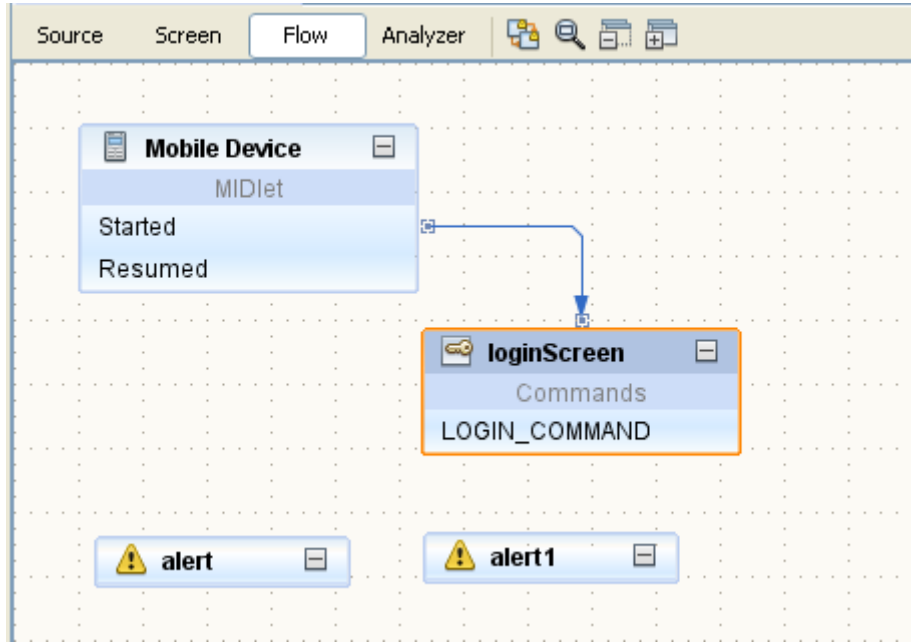


_الخاصية Background Color لتحديد لون خلفية شاشة تسجيل الدخول
_الخاصية Foreground Color لتحديد لون الخط المستخدم فى شاشة الدخول
_الخاصية Use Login Button لعمل زر الدخول تحت حقل اسم المستخدم و حقل كلمة المرور
_الخاصية Login Button Text لكتابة نص على زر الدخول
_الخاصية Login Title وهى لوضع عنوان أو نص أعلى حقل أسم المستخدم وحقل كلمة المرور

تعالوا الان إخوانى الكرام نأخذ مثال بسيط على كيفية استخدام هذا المكون ...سنقوم بعمل تطبيق يطلب من المستخدم إدخال أسمه و كلمة المرور و إذا كانت البيانات التى أدخلها المستخدم صحيحة يتم اظهار رسالة بانه تم الدخول و العكس و ذلك على النحو التالى:

_قم بإضافة المكون LoginScreen وقم بتنسيق الخصائص اعلاه ...ثم قم بربط المكون بالحدث Started للمورد MIDlet

_قم بإضافة عدد ٢ مكون... Alert ومع تنسيق خصائصهم....فالمكون الاول أجل الخاصية String مفداها) عفوا يوجد خطأ في أسم المستخدم و كلمة المرور.... (و المكون الثاني) لقد تم الدخول بنجاح.....(على أن يكون شاشة التطبيق بهذا الشكل:



_قم بالذهاب الى صفحة كتابة الشفرة ثم حدد الطريقة (public void commandAction (Command command, Displayable displayable) ...):

```
if (displayable == loginScreen) {
if (command == LoginScreen.LOGIN_COMMAND) {
```

سنقوم بتعديل الكود السابق بإضافة بعض الاكواد الية ليصبح كما يلي:

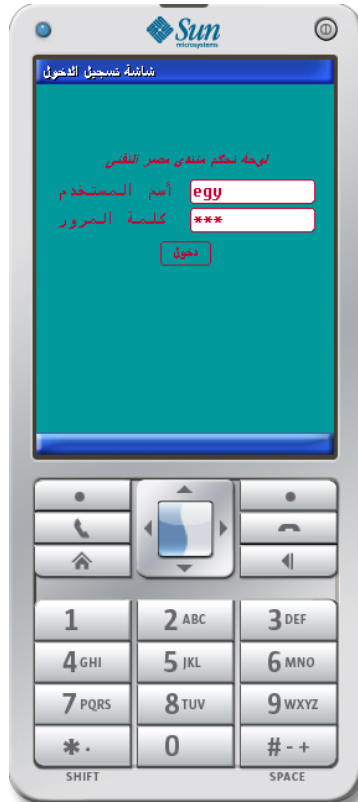
```
public void commandAction (Command command, Displayable displayable) {
```

```
if (displayable == loginScreen) {
if (command == LoginScreen.LOGIN_COMMAND) {
```

```
String getuser=loginScreen.getUsername();
String getpass=loginScreen.getPassword();
```

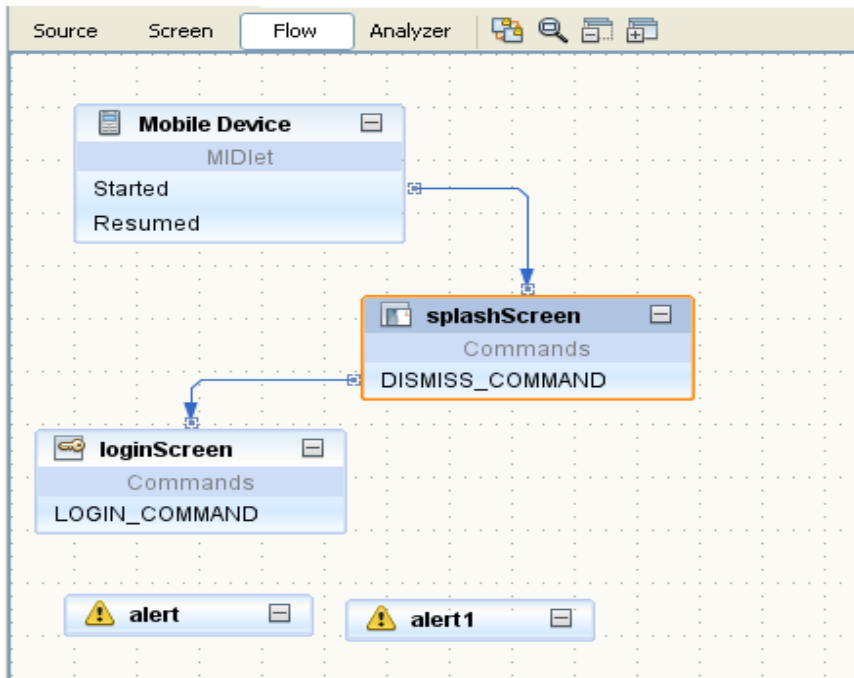
```
if (getuser.equals("egy") && getpass.equals("123")) {
switchDisplayable (getAlert (), getLoginScreen ());
} else {
switchDisplayable (getAlert1 (), getLoginScreen ());
} } }
```

في السطر الثالث قمن بالاعلان عن المتغير getuser وذلك لحفظ اسم المستخدم المدخلة من المستخدم وذلك عن طريق الاسلوب.... loginScreen.getUsername() اما في السطر الرابع قمننا بالاعلان عن المتغير getpass لحفظ كلمة المرور من المستخدم بأستخدام الطريقة.... getPassword() بعد ذلك أستخدمنا جملة شرطية للتأكد من القيم المدخلة من المستخدم و مقارنتها بأسم المستخدم (egy) و كلمة المرور (123) فإن كانت القيم المدخلة صحيحة يتم اظهار رسالة يخبرنا بذلك و العكس صحيح.... و النتيجة كما في الشكل التالي:





2 المكون..... SplashScreen وهو عبارة عن شاشة افتتاحية تظهر لمدة معينة أثناء تحميل التطبيق.... ومن اهم خصائص هذا المكون الخاصية **Image** التي تتيح وضع صورة للشاشة.... الخاصية **Text** و التي تتيح لك كتابة نص في الشاشة.... الخاصية **Timeout** و التي تحدد المدة الزمنية لابقاء الشاشة ظاهرة ثم تختفي.... في مثالنا السابق سنقوم بادراج المكون **SplashScreen** ثم قم بربط المكون بالحدث **Started** للمورد.... **MIDlet** بعد ذلك ربطه بالمكون **loginScreen** كما في الشكل التالي:



و النتيجة كما في الشكل التالي:



3 المكون..... Wait Screen وهو عبارة عن شاشة انتظار حتى تنفيذ أمر ما...ومن اهم خصائص هذا المكون الخاصية Image التي تتيح وضع صورة للشاشة....الخاصية Text و التي تتيح لك كتابة نص في الشاشة....وسيتم التعرف على كيفية استخدامة في الامثلة التالية

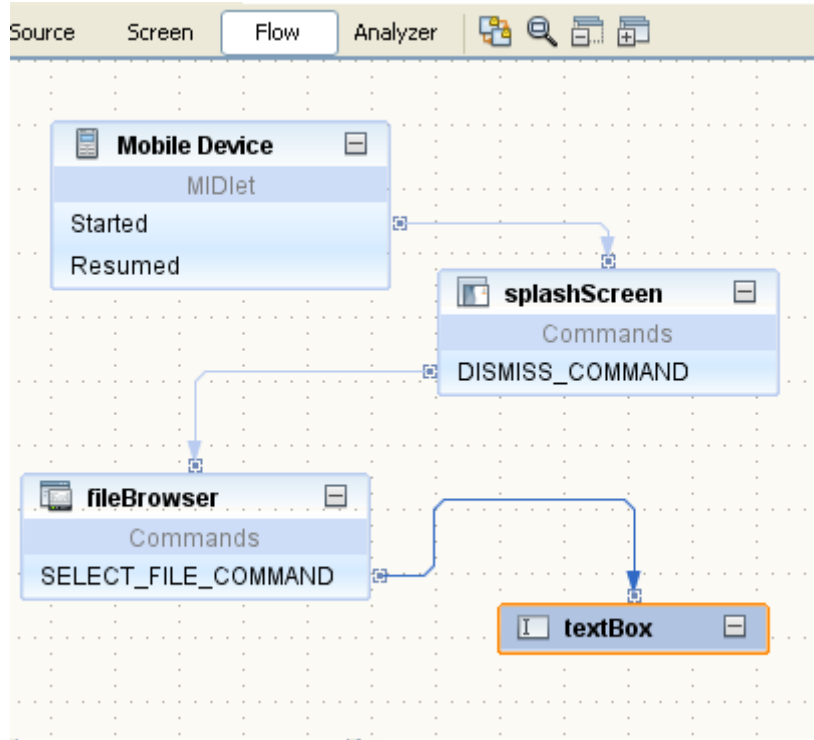
4 المكون... File Browser وهو عبارة عن مكون مخصص الذي يوفر واجهة مستخدم للعمل مع نظام ملفات الجهاز. حيث يستخدم لتصفح محتويات ذاكرة التخزين و الجوال.....اهم ما يميز هذا المكون الطريقة التابعة له `getSelectedFile` والتي من خلالها تستطيع تحديد او الحصول على الملف الذي تبحث عنه داخل الذاكرة...

مثال :لقراءة ملف من ذاكرة الهاتف....

_قم بوضع المكون SplashScreen و ربطة بالحدث Started للمورد MIDlet

_قم بوضع المكون.... File Browser و ربطة بالحدث `DISMISS_COMMAND` الخاص بالمكون SplashScreen

_قم بإضافة المكون `TextBox` و ربطة بالحدث `SELECT_FILE_COMMAND` الخاص بالمكون File Browser كما في الصورة التالية:

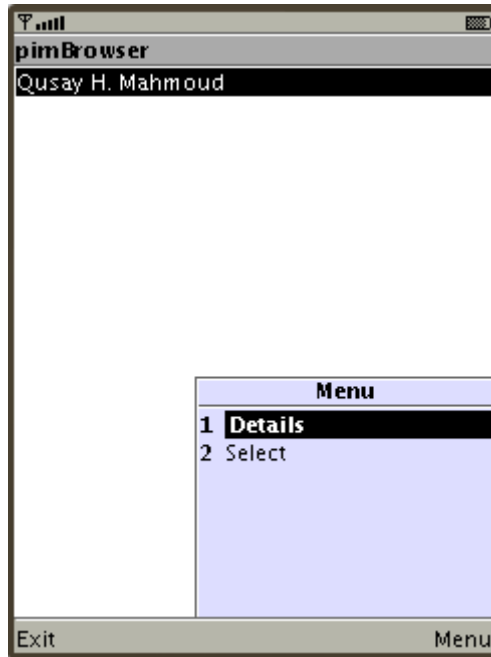


_سنقوم بإنشاء الطريقة التالية:

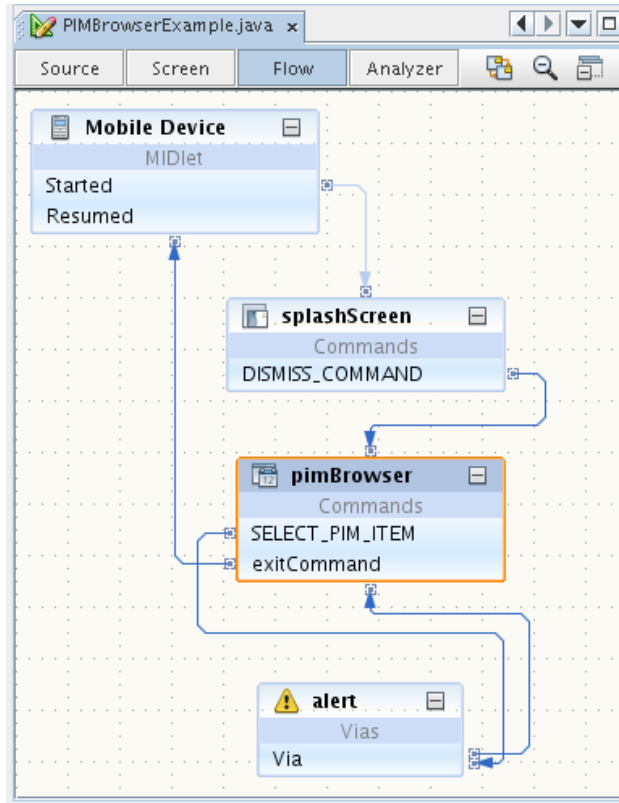
```
private void readFile() {
    try {
        FileConnection textFile = fileBrowser.getSelectedFile();
        getTextBox().setString("");
        InputStream fis = textFile.openInputStream();
        byte[] b = new byte[1024];
        int length = fis.read(b, 0, 1024);
        fis.close();
        if (length > 0) {
            textBox.setString(new String(b, 0, length));
        }
    } catch (IOException ex) {
    }
}
}
```

_نقوم بتحديد الطريقة `commandAction` وبعد سطر `readFile()` الامر `FileBrowser.SELECT_FILE_COMMAND` نقوم بادراج الطريقة التي قمنا بإنشائها:

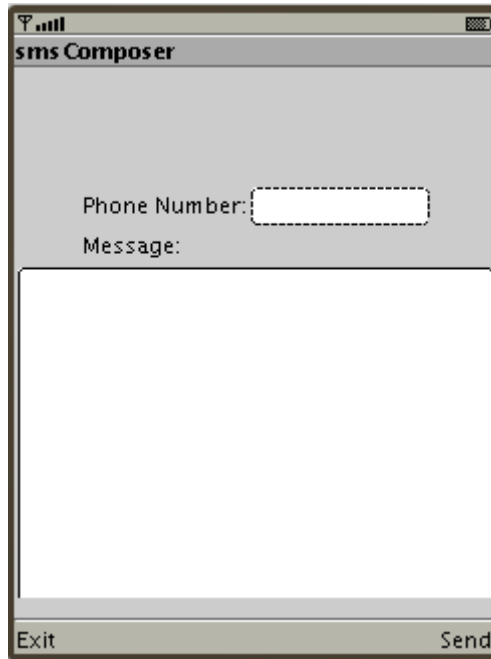
4 المكون المكون PIM Browser وهو عبارة عن مستعرض يمكنك للوصول الى مهام الجوال بسهولة مثل دفتر العناوين و الاسماء المخزنة و التقويم و قائمة المهام و غيرها ... واهم ما يميز هذا المكون ان يحتوى على عدد ٢ زر امر `commna` الاول `details` و الذى يقوم بعرض كافة البيانات المتعلقة بالاختيار المحدد ... فمثلا لو تم اختيار احد الاسماء تستطيع معرفة كافة التفاصيل لهذا الاسم



الزر الثاني SELECT و الذى من خلاله تستطيع تنفيذ اى امر... فمثلا نريد اظهار رسالة باسم المحدد... نقوم بعمل البرنامج كما فى الشكل التالى:



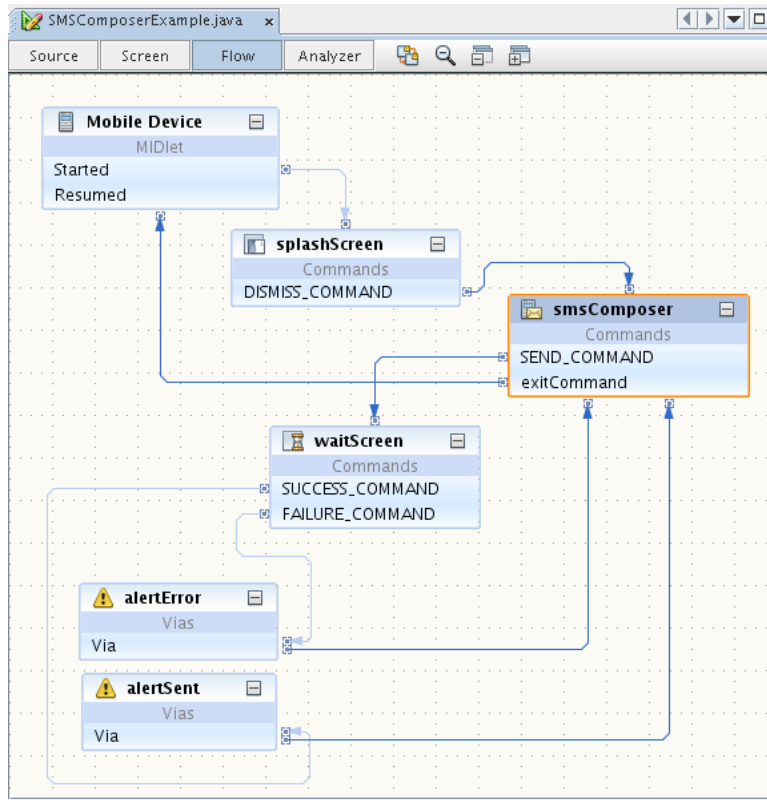
5 المكون **SMS Composer** يستخدم هذا المكون لإرسال الرسائل النصية القصيرة.... sms ومن أهم الخصائص المميزة لهذا المكون الخاصية **Message Label** و هي عنوان مربع الرسالة...الخاصية **Phone Number Label** وهي عنوان مربع رقم المرسل اليه...الخاصية **Message** وهي الخاصية التي تتيح لنا كتابة الرسالة المراد ارسالها....الخاصية **Phone Number** و التي تتيح لنا كتابة رقم جوال المرسل اليه



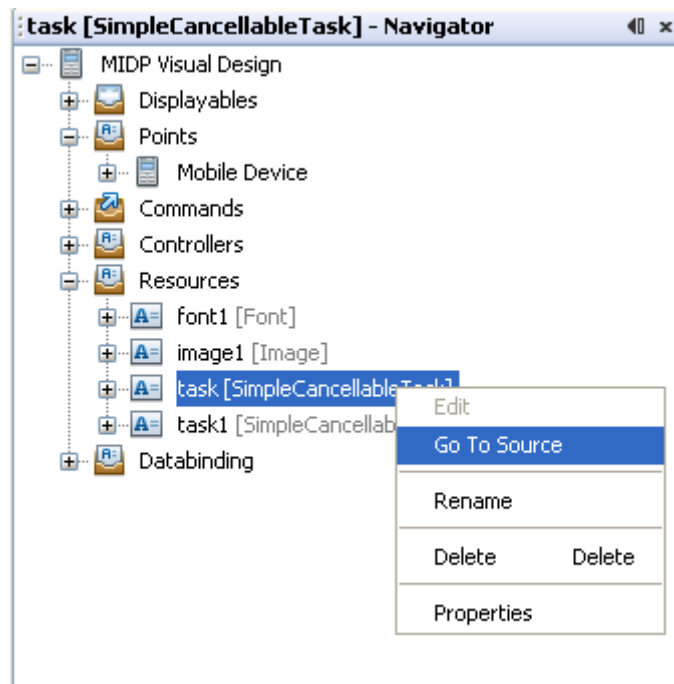
مثال.... نقوم بادراج المكونات التالية

Splash Screen
 SMS Composer
 Wait Screen
 Alert (عدد ٢)

نقوم بربط المكونات السابقة كما في الشكل التالي:



_من قائمة Navigator نحدد التصنيف Resources ثم نبحث عن task ثم النقر بزر الفأرة اليمين ومن القائمة التي ستظهر أختار الأمر go to source كما في الصورة التالية:



لتفتح لك نافذة كتابة الشفرة فنكتب الامر التالى:

```
smsComposer.sendSMS();
```

طبعاً سيتم كتابة الامر فى المنقطة التى يوجد بها التعليق here write task-execution user code

الفصل الرابع : التعامل مع المكونات التابعة للتويب Items

أهلا وسهلا بكم إخواني الكرام... في الدرسين السابقين تعرفنا على المكونات التابعة للتصنيف Displayable اليوم أن شاء الله تعالى سنتعرف سويا على المكونات التابعة للتصنيف... Items وتلك المكونات لابد لها من حاوي بمعنى أنك لا تستطيع وضعها على الشاشة بدون نافذة وبالتالي استخدام تلك المكونات لابد أولا من إدراج Form ليتم وضع تلك المكونات عليه..... واهم تلك المكونات ما يلي:

1_المكون... ChoiceGroup هو عبارة عن قائمة من العناصر يسمح للمستخدم للاختيار فيما بينهما

مثال _ : قم بإضافة المكون Form لمشروعك حتى يتسنى لنا إضافة المكون ChoiceGroup عليه
_ من خلال التويب... Screen قم بسحب المكون ChoiceGroup ثم وضعه على النافذة Form
_ قم بتحديد المكون ChoiceGroup ثم النقر يمين و من القائمة التي ستظهر أختار الأمر Go to Source ليتم فتح نافذة الشفرة
_ سنقوم بكتابة الكود التالي في المنطقة المكتوب فيها التعليق write post-init user code here وذلك على النحو التالي:

```
" , null); الفول; choiceGroup.append ("
" , null); الطعمية; choiceGroup.append ("
" , null); الفراخ; choiceGroup.append ("
" , null); الكباب; choiceGroup.append ("
" , null); المحشى; choiceGroup.append ("
" , null); امكرونة; choiceGroup.append ("
```

الكود السابق هو لاضافة العناصر الى هذا المكون وذلك من خلال الاسلوب append و النتيجة كما في الشكل التالي:



كما يمكنك تحديد نوع القائمة تلك و ذلك من خلال الخاصية... Type ففي المثال السابق كانت نوع القائمة تساوى MULTIPLE التي تجعل القائمة على هيئة مربعات أختيار متعدد كما في الصورة أعلاه.... كما يمكنك جعل القائمة على هيئة خانات أختيار فردى بجعل الخاصية Type تساوى EXCLUSIVE كما في الصورة التالية:

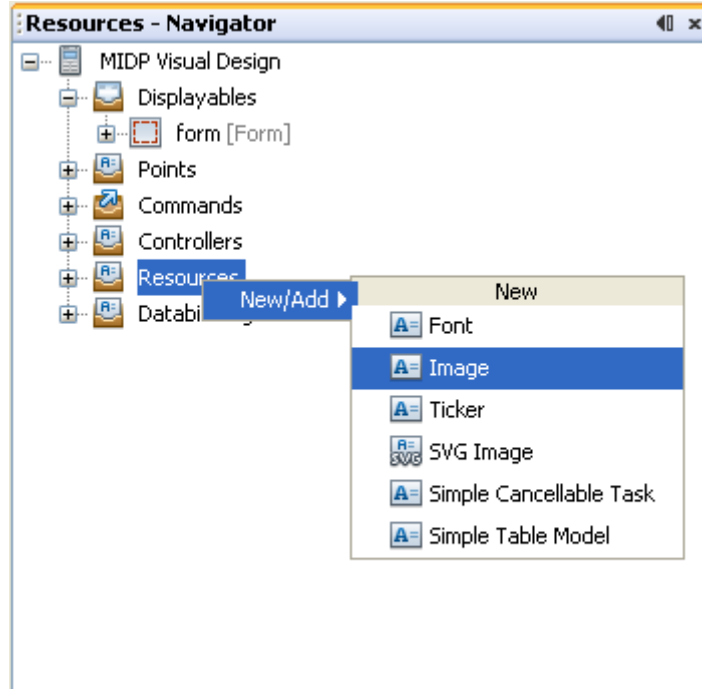


اما اذا أردت أن تجعل القائمة على هيئة مربع سرد combobox أجعل الخاصية Type تساوى POPUP فتصبح القائمة كما في الشكل التالي :

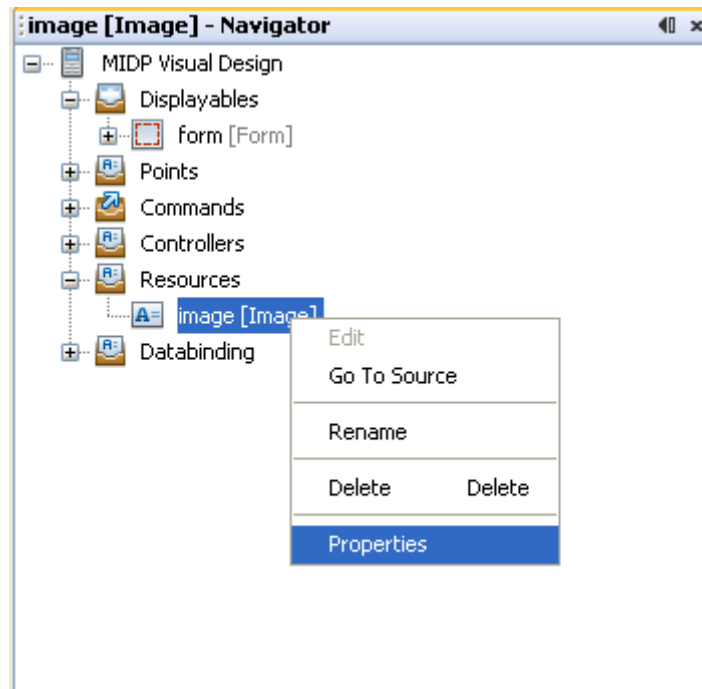


كما يمكنك وضع ايقونة أمام كل عنصر من عناصر تلك القائمة.... وذلك على النحو التالي:

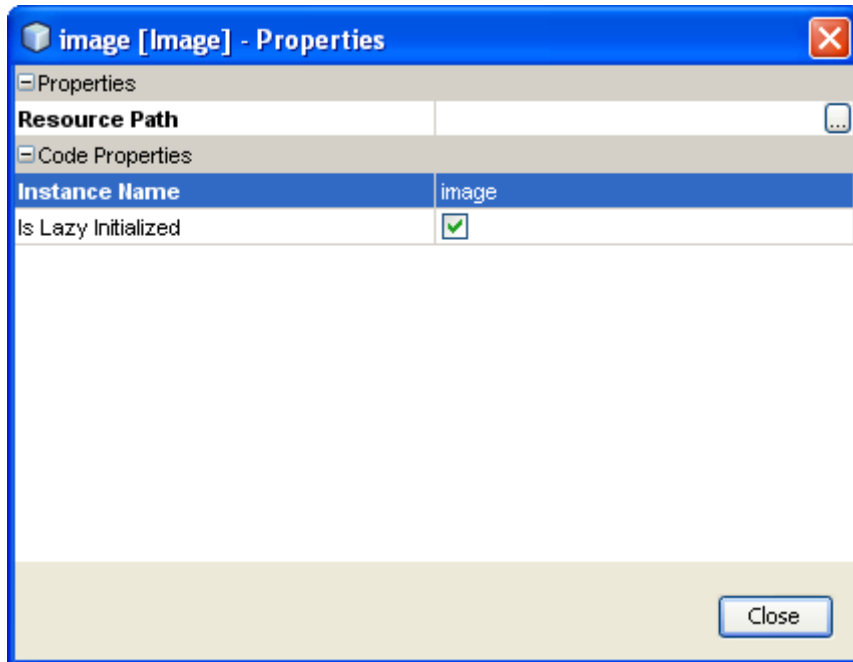
من قائمة Navigator نحدد التصنيف Resources ثم النقر يمين و من القائمة التي ستظهر اختر الامر New/Add ثم أختار الامر Image كما في الصورة التالية:



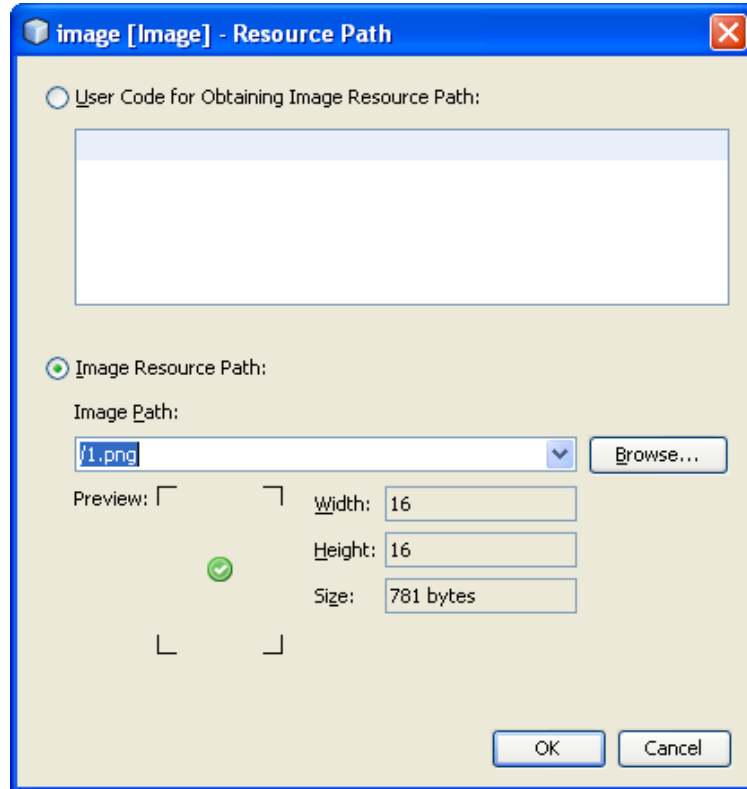
ستجد بعد ذلك ايقونة Image قم بالنقر يمين عليها و من القائمة التي ستظهر أختار الامر properties كما في الصورة التالية:



ليتم فتح لك نافذة الخصائص كما في الشكل التالي:



حدد الخاصية Resource Path ثم انقر زر الامر الصغير الذى بجانبها ليتم فتح النافذة التالية:



من خلال المربع الحوارى السابق يمكنك اختيار صورة مناسبة من خلال زر الامر (أستعراض) ثم انقر (موافق)

_نذهب الان الى صفحة كتابة الشفرة ...ونحدد الكود الذى قد كتبناه لاضافة العناصر الى القائمة ثم نقوم بتعديله و ذلك من خلال مسع الامر null ووضع مكانه الامر getImage ليصبح كما فى الشكل التالى:

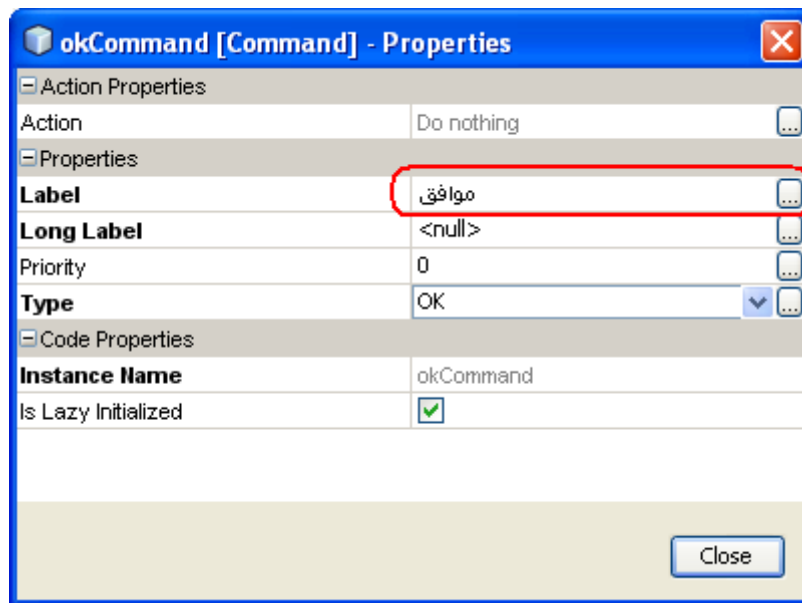
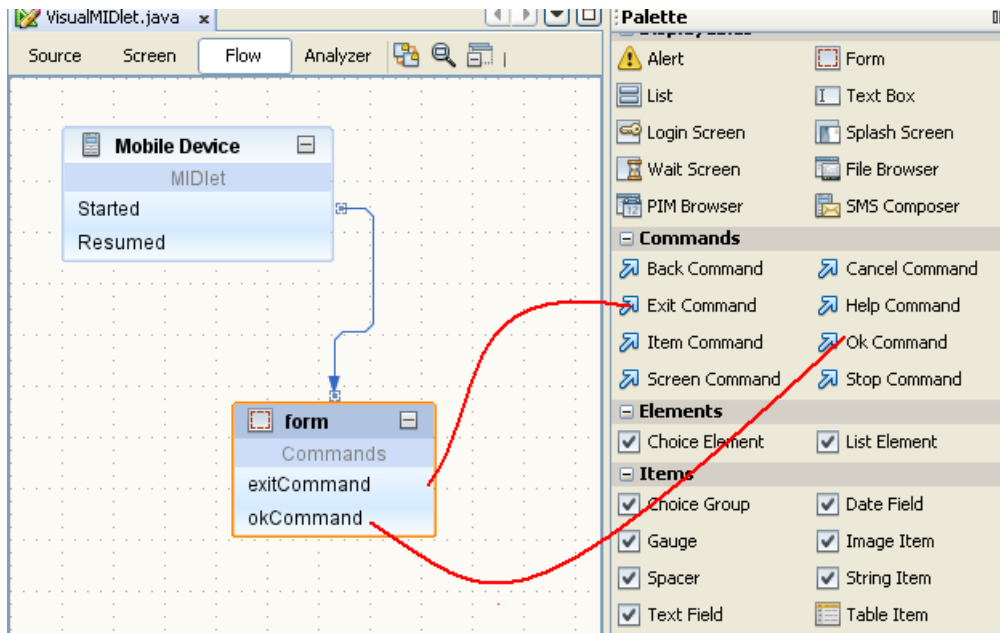
```
" , getImage ());القول choiceGroup.append("
" , getImage ());الطعميةchoiceGroup.append("
" , getImage ());الفراخchoiceGroup.append("
" , getImage ());الكبابchoiceGroup.append("
" , getImage ());المحشىchoiceGroup.append("
" , getImage ());امكرونةchoiceGroup.append("
```

و النتيجة كما فى الاشكال التالية:



كيفية اختيار عنصر من القائمة....

لكى نستطيع فعل ذلك قم بإضافة عدد 2 command الاول قم بتنسيقه و ذلك بجعل الخاصية label تساوى خروجو تحديد الخاصية type تساوى ... exit أما الثانى أجعل الخاصية label تساوى موافق و الخاصية type تساوى ... ok وطبعاً يتم اضافتهم من خلال التويب flow كما فى الشكل التالى:



_ نضيف ايضا المكون Alert و ذلك لاطهار رسالة بالعنصر المختار من قبل المستخدم
 _ نقوم بحديد المكون Alert ثم النقر يمين و من القائمة التى ستظهر أختار الامر GO TO Source ليتم فتح لك نافذة
 الكود و بداخل الطريقة Alert نكتب الكود التالى:
 الرمز:

```
String selectedvalue=choiceGroup.getString(choiceGroup.getSelectedIndex());
alert.setString(" "+selectedvalue+".");
```

و أعتقد أن الكود السابق قد تم شرحه عندما تكلمنا عن المكون list في الدروس السابقة.

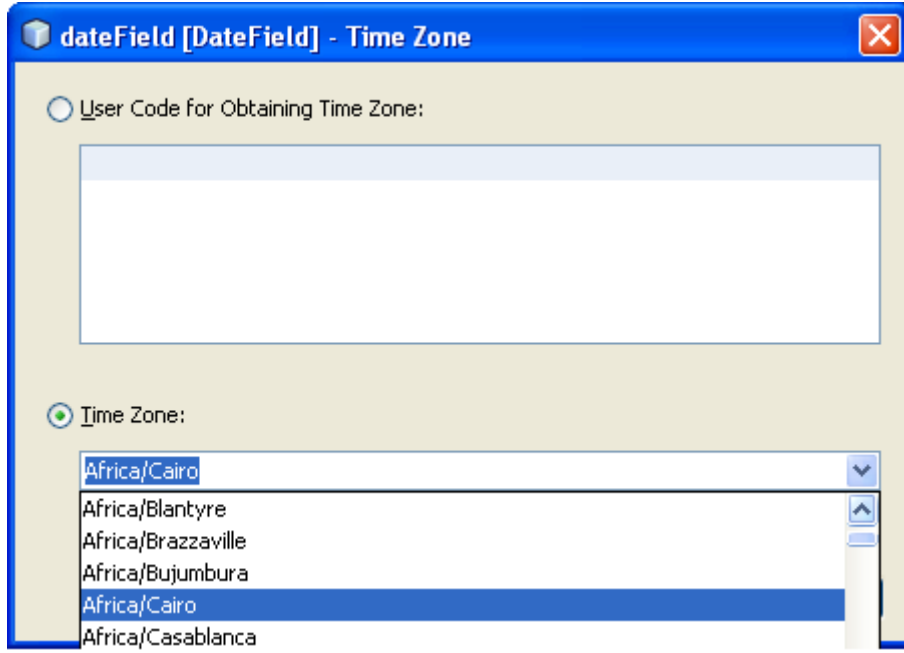
المكون 2.... DateField يستخدم هذا المكون لعرض بيانات التاريخ و الوقت (التقويم) و على النافذة..



ويعتمد حسابات هذا التقويم في على اللغة الافتراضية للجهاز ومنطقة زمنية محددة...ومن خصائص هذا المكون الخاصية Input Mode التي تشمل ثلاث خيارات:

_ DATE ... و التي تقوم بعرض التاريخ فقط
_ DATE_TIME ... و التي تقوم بعرض التاريخ و الوقت معنا
_ TIME و التي تقوم بعرض الوقت فقط

و من اهم الخصائص ايضا لهذا المكون الخاصية Time Zone و التي تقوم بعرض الوقت و التاريخ العالمي و الموحد...فقط ما عليك فعله هو النقر على زر الصغير بجوار تلك الخاصية ليتم فتح النافذة التالية:



ومن النافذة اعلاه حدد المنطقة المراد تشغيل الوقت لها.

3المكون.... **Gauge** هو عبارة عن رسم بياني أو شريط تقدمي يقدم بعض المعلومات من قيمة عددية... التي تقع بين الصفر وأقصى قيمة... ويمكنك استخدام هذا المكون في تطبيقات الميديا كمؤشر لقيمة الصوت



من اهم الخصائص المميزة لهذا المكون الخاصية **Gauge Value** و التي تحدد القيمة الحالية للعنصر... و الخاصية **Maximum Value** و التي تحدد اقصى قيمة لهذا العنصر.... أما الخاصية **Is Interactive** تجعلك قادر

على تغيير القيمة الحالية للعنصر أثناء التشغيل و يأخذ العنصر الشكل التالي:



ومن أهم الطرق التابعة لهذا المكون ما يلي:

.... `getMaxValue` _ للحصول على اقصى قيمة للعنصر
.... `getValue` _ للحصول على القيمة الحالية للعنصر

.... `ImageItem` _4 هو أحد العناصر الذي يمكنك وضع صورته به لعرضها على النافذة مثلا من أهم الخصائص المميزة له الخاصية `Image` التي من خلالها تستطيع وضع الصورة التي تريدها...

.... `StringItem` هو احد العناصر الذي يستخدم لعرض نص معين ولا يمكن المستخدم من تعديله وهو شبيه للعنصر `JLabel` او بمعنى اصح مربع تسمية من اهم الخصائص المميزة لهذا العنصر الخاصية `Text` و هو النص الذي سيتم عرضه و ايضا الخاصية `Appearance` و التي تحدد مظهر العنصر و تأخذ ثلاث قيم `PLAIN` وهو المظهر الافتراضي `HYPERLINK` _ وهو التي تعطى للمكون مظهر ارتباط تشعبي `BUTTON` _ وهي القيمة التي تعطى للمكون مظهر زر الامر كما في الصورة التالية:

ومن أهم الطرق الخاصة بهذا المكون:....

_ الطريقة..... `getText` للحصول على النص الموجود بهذا المكون

_ الطريقة.... `setText` لوضع نص بالمكون

_ الطريقة... `setPreferredSize` و التي تقوم بتحديد قيم الطول و العرض لهذا المكون

_6_المكون..... TextField هو عبارة عن حقل نصي قابل للتعديل مثل مربع النص **textbox** الذي تعرفنا عليه في بداية هذه الدروس و الفرق بينهما هو أن حقل النص **TextField** لا قبل كتابة أسطر متعددة على غرار مربع النص.... **textbox** وايضا حقل النص لا بد أن يضع على نافذه و لا يمكن استخدامه بدون نافذه... كما أن حقل النص له حد اقصى من الحروف وهي يقدر بحوالي ٣٢ حرف غير ذلك سيكون حجمة أكبر من حجم شاشة الجوال....

من أهم الخصائص المميز لهذا المكون ما يلي :

_الخاصية..... Input Constraints وهي عبارة عن قيود أذخال مختلفة يمكنك استخدام أحدها و ذلك لاجبار المستخدم أذخال بيانات معينة و التي تشمل القيود التالية:

ANY
EMAILADDR
NUMERIC
PHONENUMBER
URL
DECIMAL

فمثلا اذا قمنا باختيار القيمة **NUMERIC** معنى ذلك سيتم اجبار المستخدم من أذخال ارقام فقط...ولو استخدمنا **PHONENUMBER** يتم اجبار المستخدم استخدام الفواصل أرقام وعلامات الترقيم بما يتناسب مع الاتفاقيات لرقم الهاتف قيد الاستعمال.... كما يمكنك تنسيق القيمة المدخلة من خلال عدة خيارات اخرى و تشمل:

PASSWORD
UNEDITABLE
SENSITIVE
NON_PREDICTIVE
INITIAL_CAPS_WORD
INITIAL_CAPS_SENTENCE

_الخاصية..... Input Modes وهي خاصية تتيح لك تحديد وضع الادخال للعنصر عند التركيز عليه....بمعنى عندما يتم تحديد العنصر و لغة الكتابة الافتراضية لجوالك هي الانجليزية يمكنك تحديد تلك الخاصية بحيث يتم تعديل وضع الكتابة من الانجليزية الى العربية مباشرة بمجرد تحديد هذا المكون بدلا من الذهاب إعدادات الجوال لتغير لغة الكتابة الى العربية.... أو يتم تحديد وضع الادخال على الارقام

الفصل الخامس : التعامل مع نظام إدارة السجلات RMS

أهلاً وسهلاً بكم إخوانى الكرام....تعرفنا فى الدروس السابق على المكونات الرسومية التابعة للتويب
Displayable و ايضا المكونات التابعة للتويب.... Items درس اليوم سنتعلم فيه سوياً كيفية التعامل
مع Record Management System وقبل أن نبدأ تعالوا نتعرف على بعض المفاهيم:

أولاً : مفهوم نظام إدارة السجلات (RMS)

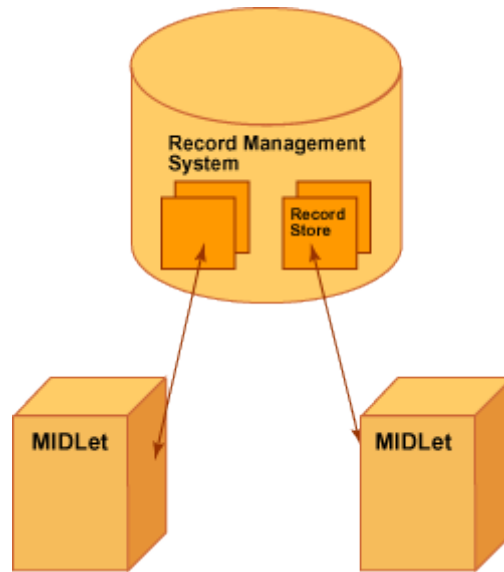
نظام إدارة السجلات RMS هو اختصار للجملة الانجليزية Record Management System وهو
آلية أو قاعدة بيانات بسيطة المنحى للتخزين للبيانات باستمرار واسترجاعها في وقت لاحق و التى يوفره
المورد MIDlets أو معلومات الملف الشخصي للجهاز المحمول

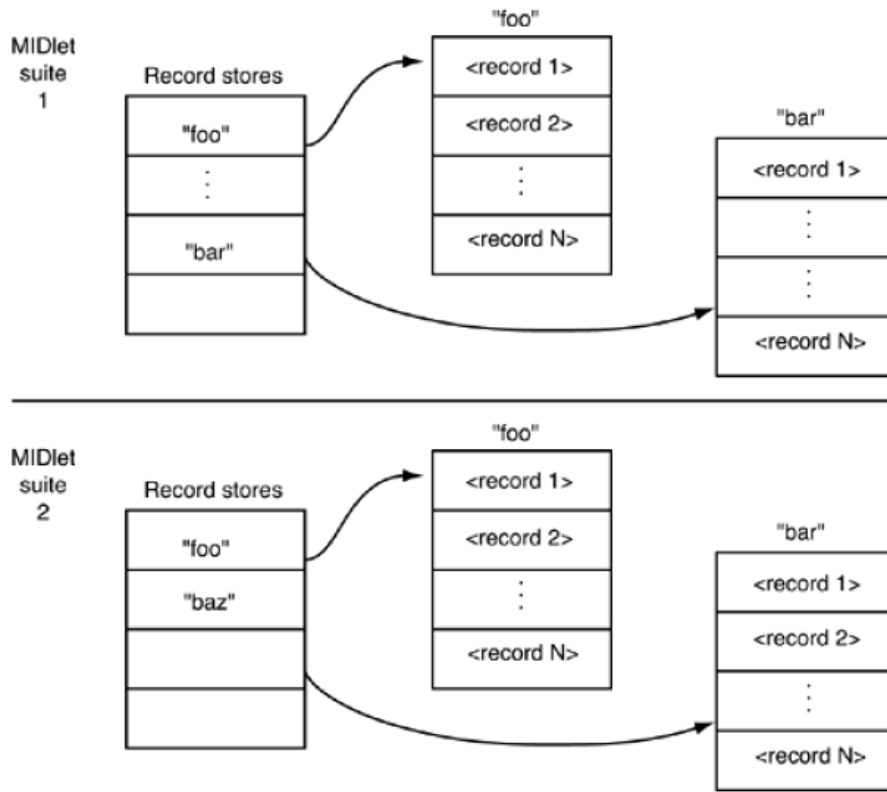
ثانياً : مفهوم السجلات Records

السجلات هى عبارة عن مصفوفة من البيانات من نوع bytes تمكن المطور من تخزين البيانات فيها عن
طريق الكائن DataOutputStream و الكائن DataInputStream و ايضا
الكائن ByteArrayOutputStream ويتم تحديد السجلات بشكل فريد و إعطاء كل سجل مفتاح اساسى
بحيث يبدأ برقم واحد للسجل الاول داخل ما يسمى "مخزن السجلات RecordStore"

ثالثاً : مفهوم مخزن السجلات RecordStore

هو عبارة عن مجموعة من السجلات المرتبطة بمنصة MIDlets و التى ستبقى ثابتة معه...ويتم تسمية كل
مخزن سجل RecordStore بأسم فريد لتميز كل مخزن عن الاخر ويجب أن يكون الاسم من
نوع String ولا يزيد عن ٣٢ حرف كما أن أسماء مخزن السجلات حساسة لحالة الاحرف
فمثلاً "EgyTech" ليست مثل "egytech"





1_ إنشاء مخزن السجلات RecordStore توجد عدة طرق و اساليب لانشاء مخزن سجل RecordStore و ذلك على النحو التالي:

بداية يجب أستيراد المكتبة اللازمة rms.RecordStore فى اى من الطريق التى سيتم التعرف عليها و ذلك عن طريق الكود التالي:

```
import javax.microedition.rms.RecordStore;
```

الطريقة الاولى : الاعلان عن مخزن السجل RecordStore و إعطاء اسم مناسب له بأستخدام الكود التالي:

```
private RecordStore myegy = null;
```

كما ترى أخى الكريم فى الكود أعلاه قمنا بالاعلان عن مخزن السجل myegy وجعلنا القيمة له تساوى null

الطريقة الثانية: فتح و انشاء مخزن السجل بدون أى شروط بأستخدام الكود التالي:

```
RecordStore.openRecordStore("myegy",true);
```

الطريقة الثالثة: فتح و انشاء مخزن السجل بشرط جعل منصات MIDlet الاخرى قادرة للوصول اليه و السماح بالكتابة فيه و ذلك با ستخدام الكود التالي:

```
RecordStore.openRecordStore("myegy", true,2, true);
```


وكما ترون في الكود أعلاه يتضمن ٤ عوامل:

.... myegy_ وهو أسم مخزن السجل المراد إنشائه

... true_ وهي لإنشاء المخزن الذى قمنا بتحديد اسمه و يمكنك جعلها false إذا كنت قد أنشأتها بالفعل

_ الرقم ٢.... وهو متغير من نوع int للتعبير عن حالة الوصول هذا المخزن و تأخذ قيمتين
١ AUTHMODE_PRIVATE_ و التى تعنى جعل المخزن myegy لا يمكننا الوصول اليه من خلال المنصات
الاخري ٢ AUTHMODE_ANY_ اى يمكن الوصول اليه

.... True_ متغير منطقي للتعبير عن أمكانية الكتابة فى هذا المخزن من خلال المنصات الاخرى

2_ الاستثناءات الناتجة من فتح و إنشاء مخزن السجل RecordStore

...
عند استخدامك لاحد الطرق السابقة فى انشاء وفتح مخزن السجل RecordStore سينتج الاستثناءات التالية:

.... RecordStoreException_ وهو خطأ يظهر فى حالة وجود مشكلة ما اثناء فتح وانشاء مخزن السجل

... RecordStoreNotFoundException_ وهو خطأ يظهر فى حالة محاولة فتح مخزن السجل غير موجود
.... كأستخدامك مثلا الطريقة الاولى لإنشاء المخزن واستخدام الامر التالى لفتحه:

```
RecordStore.openRecordStore("myegy",false);
```

فى الكود السابق نحاول فتح المخزن myegy وقد قمنا باستخدام الامر false أى أن المخزن قد انشأناه من قبل

.... RecordStoreFullException_ هذا الخطأ يظهر فى حالة كون مخزن السجل ممتلئ

.... IllegalArgumentException_ هذا الخطأ يظهر فى حالة كونك تستخدم اسم لمخزن السجل غير صالح
للاستخدام.

3_ إغلاق مخزن سجل

```
myegy.closeRecordStore();
```

4_ حذف مخزن سجل

```
RecordStore.deleteRecordStore("myegy");
```

5_ الحصول على حجم مخزن السجل:

```
myegy.getSize();
```

و الكود السابق يقوم بإرجاع مقدار المساحة، بالبايت،

6_ إضافة سجل الى مخزن السجلات:.

يتم إضافة سجل الى مخزن السجلات باستخدام الكود التالي:

```
myegy.addRecord(byte[] data, int offset, int numBytes)
```

وكما ترون في الكود أعلاه يطلب منك ثلاث عوامل لإضافة سجل جديد وهي على النحو التالي:

المعامل الاول data وهي البيانات التي سيتم تخزينها في هذا السجل ويجب أن تكون البيانات ذات طول صفري أي مصفوفة من نوع byte..... قد تكون هذه المعلمة فارغة null اذا اردت إضافة سجل فارغ
المعامل الثاني... offset رقم المؤشر إلى المخزن المؤقت للبيانات من أول بايت ذات الصلة لهذا السجل
المعامل الثالث.... numBytes وهي عدد وحدات البايت المراد حجزها من المخزن myegy لهذا السجل

مثال:

```
byte[] name ="ahmed".getBytes();  
myegy.addRecord(name,0,name.length);
```

في السطر الاول قمنا بعمل مصفوفه من نوع byte اسمها name ووضع فيها القيمة... ahmed أما السطر الثاني فقمنا بإضافة سجل جديد لمخزن السجل RecordStore باستخدام الاسلوب addRecord مع تحديد البيانات المراد تخزينها بالسجل وهي في مثالنا هذا المصفوفة name ثم وضعنا رقم المؤشر لهذا السجل (٠)... وايضا عدد الواحدات لهذا السجل و ذلك باستخدام الاسلوب name.length

_قراءة سجل

```
byte[] getRecord(int recordId)
```

كما ترون في الكود اعلاه تجد أن الاسلوب getRecord يقوم بإرجاع قيمة السجل بناء على رقم المعرف الخاص به record Id فقلد أخيرنا بداية الدرس أنه يتم حديد السجلات بشكل فريد و إعطاء كل سجل مفتاح اساسي id بحيث يبدأ برقم واحد للسجل الاول...أذن السجل الذي قمت بإضافة في المثال السابق المفتاح الاساسي له هو رقم ١...و بالتالي فإن القراءة ستكون كما يلي:

```
byte[] data = myegy.getRecord(1);
```

_تحديث سجل... اذا اردت تحديث سجل موجود من قبل ببعض البيانات نستخدم الكود التالي:

```
byte[] name ="syed".getBytes();  
myegy.setRecord(1,name,0,name.length);
```

طبعا رقم ١ هذا هو الرقم المعرف للسجل الذي قمنا بإضافته في بداية المثال

_حذف سجل.... اذا اردت حذف سجل أستخدم الكود التالي:

```
myegy.deleteRecord(1);
```

_ عدد السجلات في مخزن السجل..... يمكنك تحديد عدد السجلات في مخزن السجل الحالي باستخدام الكود التالي:

```
myegy.getNumRecords();
```

_ الحصول على حجم السجل بالبايت

```
myegy.getRecordSize(1);
```

مثال أكثر توضيحا :

بعد ما تعرفنا في الجزء الاول من هذا الدرس عن كيفية التعامل مع نظام إدارة السجلات سنتعلم سويا انشاء تطبيق عملي وليكن مخزن كتب....

الخطوة الاولى : قم بإضافة المكون form وذلك من خلال التويب flow وسحب المكون من القائمة Displayable

الخطوة الثانية : قم بإضافة المكون TextField ويتم وضعه على النافذة form

الخطوة الثالثة : نقوم بإضافة عدد ٨ اوامر Command من نوع itemcommand ويتم تنسيقهم على النحو التالي:

إنهاء Label

Instance Name exit

فتح المخزن Label

Instance Name open

إضافة Label

Instance Name add

حذف Label

Instance Name delet

عدد السجلات Label

Instance Name no

عرض Label

Instance Name view

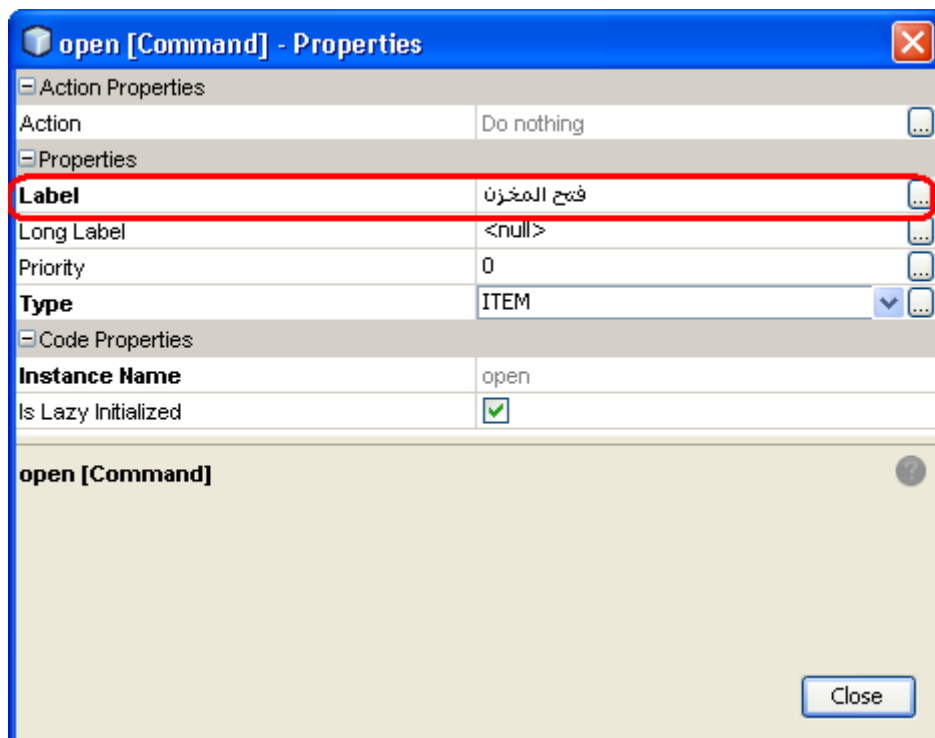
إغلاق Label

Instance Name close

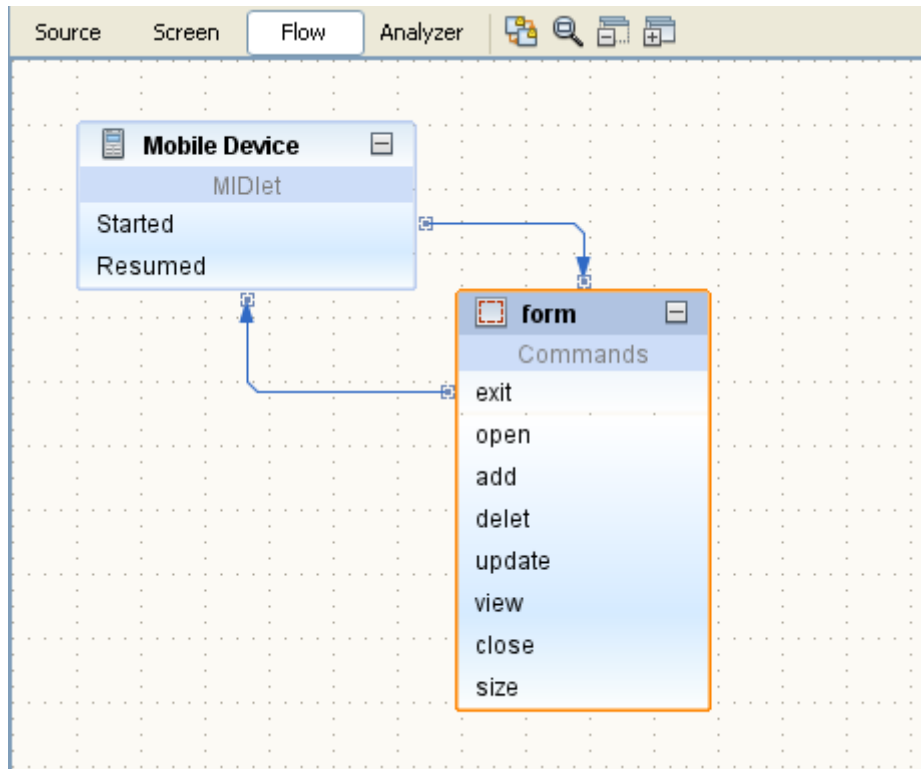
حجم المخزن Label

Instance Name size

و ذلك من خلال نافذة الخصائص لوضع اسم مناسب لكل امر كما فى الشكل التالى:



كما يمكنك تغيير Instance Name لاي أمر عن طريق تحديد الامر ثم النقر يمين ومن القائمة التى ستظهر أختار الامر..... rename وتلك هى الصورة النهائية للتصميم:



The screenshot shows the 'Screen' tab of a development tool. On the left, a 'Device Screen' is shown with a 'form' component. The 'form' component has a text input field with the text 'أدخل أسم الكتاب' and a '<null>' placeholder. On the right, there is an 'Assigned Resources' panel. The 'Assigned Resources' panel has an 'Assigned Commands' list containing: exit, open, add, delet, update, view, close, size. Below this list are buttons for 'Assigned Item Commands' and 'Resources'.

الخطوة الرابعة : كتابة الاكواد و ذلك على النحو التالي:

اولا : إنشاء الطرق اللازمة

_الاعلان عن متغير للتعبير عن المخزن الذى سنقوم بإنشائه....و ذلك باستخدام الكود التالى :

```
private RecordStore rs = null;
```

_نقوم بعمل طريقة و ليكن openRecStore لفتح و انشاء المخزن و ذلك على النحو التالى:

```
public void openRecStore(){
try{
rs=RecordStore.openRecordStore("myegy", true );
}catch (Exception e){}
}
```

و يجب ملاحظة اننا استخدمنا التعليمة catch try و ذلك لمعالجة الاستثناءات التى اشرنا لها فى شرح الدرس
_نقوم بعمل طريقة لاجلاق المخزنعلى النحو التالى

```
public void closeRecStore(){
try{
rs.closeRecordStore();
}catch (Exception e){}
}
```

_نقوم بعمل طريقة لاضافة سجل للمخزن ،،،،،،، وذلك على النحو التالى:

```
public void writeRecord(String str){
byte[] rec = str.getBytes();
try{
rs.addRecord(rec, 0, rec.length);
}catch (Exception e){}
}
```

و يمكنك ملاحظة تننا قمنا بإنشاء طريقة ذات معامل str

_نقوم بعمل طريقة لعرض جميع سجلات المخزنوذلك على النحو التالى:

```
public void readRecords(){
try{
byte[] recData = new byte[5];
int len;

for(int i = 1; i <= rs.getNumRecords(); i++){
if(rs.getRecordSize(i) > recData.length){
recData = new byte[rs.getRecordSize(i)];
}
```

```

}
len = rs.getRecord(i, recData, 0);
form.append("-----");
form.append("_ " + i + " : " + new String(recData, 0, len));
form.append("-----");
}
}catch (Exception e){}

```

ويمكنك ملاحظة اننا قمنا باستخدام حلقة تكرارية للحصول على رقم id الخاص بكل سجل من سجلات المخزن باستخدام الطريقة `getRecordSize` ومن ثم استخدام رقم id للحصول على قيمة السجل عن طريق الطريقة `getRecord`

_نقوم بعمل طريقة لجلب عدد السجلات في المخزنعلى النحو التالي

```

public void noRecord(String str){

try{
rs.getNumRecords();
}catch (Exception e){}

}

```

_نقوم بعمل طريقة لمعرفة حجم المخزن بالبايتعلى النحو التالي:

```

public void sizeRecStore(){
try{
rs.getSize();
}catch (Exception e){}
}

```

_نقوم بعمل طريقة لحذف سجل من المخزنوذلك على النحو التالي:

```

public void deletRecord(int id){

try{
rs.deleteRecord(id);
}catch (Exception e){}
}

```

ثانيا : كتابة الاكواد اللازمة تحت كل أمر **command** وذلك على النحو التالي:

_الامر open المطلوب منه فتح المخزن أذن سنقوم بتحديد الامر ثم النقر يمين ومن القائمة التي ستظهر نختار الامر `go to source` لفتح فافذة الكود ووضع مؤشر الكتابة تحت الامر كما في الشكل التالي:

```

} else if (command == delet) {
// write pre-action user code here

// write post-action user code here
} else if (command == exit) {
// write pre-action user code here
exitMIDlet ();
// write post-action user code here
} else if (command == no) {
// write pre-action user code here

// write post-action user code here
} else if (command == open) {
// write pre-action user code here
// write post-action user code here
} else if (command == size) {
// write pre-action user code here

```

هنا سيتم كتابة الكود

و نكتب الكود التالي وهو تنفيذ الطريقة التي قمنا بانشائها نوا لفتح المخزن....

```
openRecStore();
```

_الامر..... add نكتب الكود التالي

```
writeRecord(textField.getString());
```

الكود اعلاه يقوم باخذ القيمة التي يكتبها المستخدم في حقل النص textField وتخزنها في المخزن كسجل.

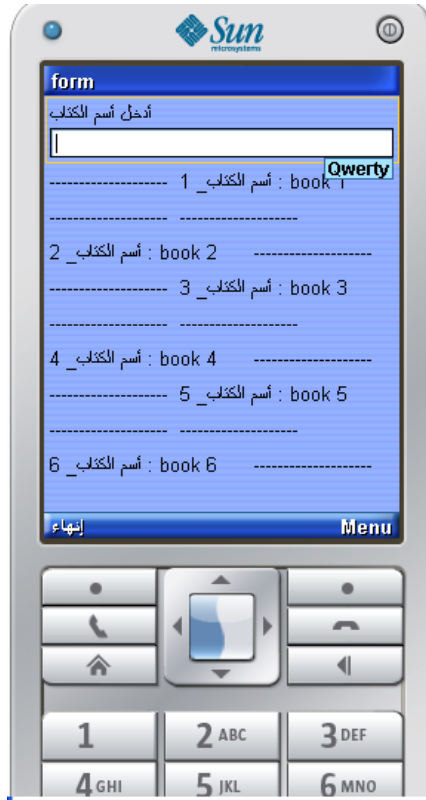
الامر..... viwe نكتب الكود التالي:

```
readRecords();
```

الامر..... delet نكتب الكود التالي:

```
String s = textField.getString();
int no =Integer.parseInt(s);
deletRecord(no);
```

أكمل باقي الاوامر بنفسك على نفس المنوالونتيجة هذا البرنامج على النحو التالي:



الفصل السادس : الاتصال بالويب web

إخواني الكرام أهلاً وسهلاً بكم... في هذا الدرس إن شاء الله تعالى سنتعلم سوياً كيفية الاتصال بالإنترنت و الشبكات من جوالك وذلك من خلال المكتبة `javax.microedition.io` والتي تمكنك من عملية الاتصال بالشبكات و المكتبة `java.io` والتي توفر الإدخال / الإخراج (I / O) في مختلف الطبقات والواجهات ونظام البيانات والجداول....

أولاً :الاتصال بشبكات HTTP

شبكات HTTP لمن لا يعرفها هي الشبكات التي تستخدم البروتوكول (HTTP/Hypertext Transfer Protocol) وهو المستخدم لنقل الملفات النصية أو النصوص الكبيرة عبر الإنترنت وهذا البروتوكول هو مثال واضح حيث يظهر لنا في شريط العنوان في متصفحات الإنترنت.... <http://www> ومعظم مواقع الويب تستخدم هذا البروتوكول

ويمكنك الاتصال بتلك الشبكات من خلال الحزمة `javax.microedition.io` وذلك باستخدام الكائن

`javax.microedition.io.HttpConnection`

يتم إرجاع `HttpConnection` من `Connector.open` عندما يتم الوصول إلى رابط..... <http://> " وذلك على النحو التالي:

```
String url = "http://egy-tech.forumegypt.net/simple";
InputConnection ic = (InputConnection)Connector.open(url);
InputStream in = ic.openInputStream();
ic.close();
```

وكما ترون في الكود أعلاه قمنا بالاعلان عن متغير `url` وذلك لوضع رابط الاتصال بالويب.... ثم فتح هذا الرابط باستخدام الاسلوب `Connector.open` وقراءة البيانات من `InputStream`

مثال / نفترض أننا نريد الاتصال بصورة من موقع معين....سيكون الحل على النحو التالي:

1_ قم بإنشاء مشروع جديد وليكن `web` ثم أنشئ كلاس بأسم `mainForm` مثلاً

2_ من خلال التويب `Flow` قم بإضافة نافذة

3_ قم بإضافة أمر تحكم `exitCommand` و اجعل الخاصية `label` تساوى "إنهاء "

4_ أفتح صفحة الكود ستجد الكود الخاص بإنشاء الكلاس `mainform` كما يلي:

```
public class main Form extends MIDlet implements CommandListener {
```

سنقوم بتعديل الكود السابق و ذلك بإضافة الواجهة `Runnable` ليصبح كما يلي:

الرمز:

```
public class MainForm extends MIDlet implements CommandListener, Runnable {
```

5_ نقوم باتسيراد المكتبات التالية:

```
import java.io.*;
import javax.microedition.io.*;
```

6_ وبما أن مشروعنا يحتوي على الواجهة Runnable فلا بد من إنشاء الطريقة Run وذلك على النحو التالي:

```
public void run() {
}
}
```

7_ داخل تلك الطريقة سنقوم بكتابة الاوامر اللازمة للاتصال للموقع و قراءة الصورة و ذلك على النحو التالي:

نقوم بالاعلان عن كائنات الاول من نوع HttpURLConnection و الذى سيقوم بالاتصال ببروتوكول http و الثانى من نوع DataInputStream و الذى سيقوم بقراءة البيانات.... على النحو التالي:

```
HttpURLConnection hc = null;
DataInputStream in = null;
```

نقوم بفتح الاتصال بالرابط و ذلك باستخدام الكود التالي::

```
try {
    String url = "http://xpedition009.com/simple/java.png";
    hc = (HttpURLConnection) Connector.open(url);
```

نقوم بقراءة البيانات و ذلك باستخدام الكود التالي:

```
int length = (int) hc.getLength();
byte[] data = null;
if (length != -1) {
    data = new byte[length];
    in = new DataInputStream(hc.openInputStream());
    in.readFully(data);
} else {
    int chunkSize = 512;
```

```

int index = 0;
int readLength = 0;
in = new DataInputStream(hc.openInputStream());
data = new byte[chunkSize];
do {
    if (data.length < index + chunkSize) {
        byte[] newData = new byte[index + chunkSize];
        System.arraycopy(data, 0, newData, 0, data.length);
        data = newData;
    }
    readLength = in.read(data, index, chunkSize);
    index += readLength;
} while (readLength == chunkSize);
length = index;
}

```

كما تلاحظ في الكود أعلاه.... في السطر الأول قمنا بالإعلان عن المتغير length لتخزين بيانات الملف فيه وذلك باستخدام الأسلوب `getLength` لكائن الاتصال.... `hc` بعد ذلك قمنا بعمل مصفوفة من نوع `byte` وهي... `data` بعد ذلك استخدمنا جملة شرطية للتأكد من وجود بيانات للقراءة... ويتم قراءة البيانات باستخدام `DataInputStream(hc.openInputStream())` وعن طريق حلقة تكرارية يتم تخزين البيانات في المصفوفة `data`

_ نقوم بتحويل البيانات التي تم تخزينها في المصفوفة `byte` الى صورة و ذلك لكوننا قمنا بقراءة الصورة كبايتات.... وذلك على النحو التالي:

```

Image image = Image.createImage(data, 0, length);

ImageItem imageItem = new ImageItem(null, image, 0, null);
form.append(imageItem);

```

أذن الكود بالكامل لتلك الطريقة سيكون على النحو التالي:

```

public void run() {

    HttpURLConnection hc = null;
    DataInputStream in = null;

    try {

        String url = "http://xpedition009.com/simple/java.png";
        hc = (HttpURLConnection) Connector.open(url);

        int length = (int) hc.getLength();
        byte[] data = null;

```

```

if (length != -1) {
    data = new byte[length];
    in = new DataInputStream(hc.openInputStream());
    in.readFully(data);
} else {
    int chunkSize = 512;
    int index = 0;
    int readLength = 0;
    in = new DataInputStream(hc.openInputStream());
    data = new byte[chunkSize];
    do {
        if (data.length < index + chunkSize) {
            byte[] newData = new byte[index + chunkSize];
            System.arraycopy(data, 0, newData, 0, data.length);
            data = newData;
        }
        readLength = in.read(data, index, chunkSize);
        index += readLength;
    } while (readLength == chunkSize);
    length = index;
}
Image image = Image.createImage(data, 0, length);
ImageItem imageItem = new ImageItem(null, image, 0, null);
form.append(imageItem);

");
form.setTitle("تم التحميل");
} catch (IOException ioe) {
    StringItem stringItem = new StringItem(null, ioe.toString());
    form.append(stringItem);
    form.setTitle("Done.");
} finally {
    try {
        if (in != null) {
            in.close();
        }
        if (hc != null) {
            hc.close();
        }
    } catch (IOException ioe) {
    }
}
}
}
}

```

بعد ذلك نقوم بتحديد الطريقة startApp ثم نكتب الكود التالي وهو المسؤول عن تشغيل الطريقة أعلاه RUN وذلك على النحو التالي:

```
Thread t = new Thread(this);  
t.start();
```

و النتيجة كما فى الصورة التالية:



مثال آخر .. فى المثال أعلام قمنا بفتح اتصال بالويب و قراءة صورة لكن كيف يمكنك فتح موقع ويب .. سنكون نفس الكود السابق مع بعض التعديل البسيط فقط سنقوم بإنشاء الطريقة run كما يلى:

الرمز:

```
HttpConnection hc = null;  
InputStream is = null;  
  
try {  
    hc = (HttpConnection) Connector.open("http://google.com");  
    strItem.setText("جارى التحميل");  
    hc.setRequestMethod(HttpConnection.GET);  
    if (hc.getResponseCode() == HttpConnection.HTTP_OK) {  
        String str;  
        is = hc.openInputStream();  
        int length = (int) hc.getLength();  
        if (length != -1) {  
            byte incomingData[] = new byte[length];
```

```

is.read(incomingData);
str = new String(incomingData);
} else {
    ByteArrayOutputStream bytestream = new ByteArrayOutputStream();
    int ch;
    while ((ch = is.read()) != -1) {
        bytestream.write(ch);
    }
    str = new String(bytestream.toByteArray());
    bytestream.close();
}

form.append(str);

```

```

} else {
    غير قادر على الفتح;")
    strItem.setText("
}

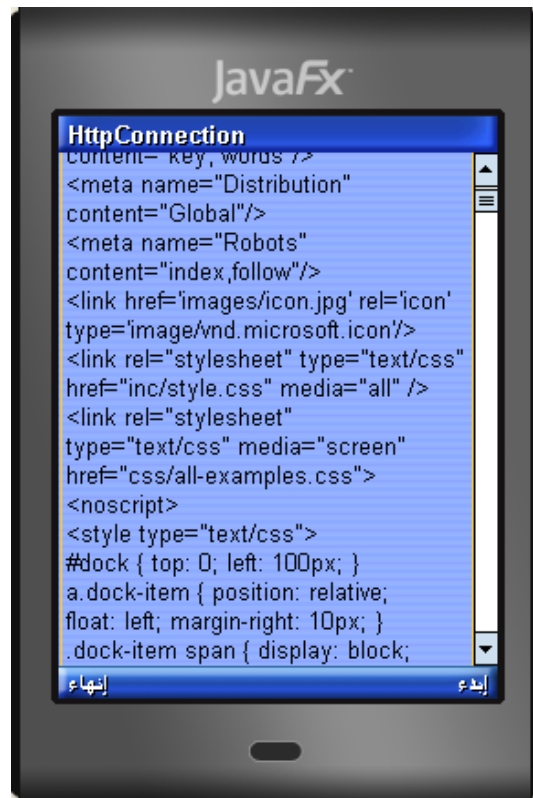
```

```

} catch (Exception exe) {
} finally {
    if (is != null) {
        try {
            is.close();
        } catch (Exception error) {
        }
    }
    if (hc != null) {
        try {
            hc.close();
        } catch (Exception error) {
        }
    }
}
}
}
}

```

الكود أسهل من سابقه.... لكن عند تشغيلك لهذا الكود ستجد صفحة الويب هكذا



وهذا لان تلك البيانات عبارة بيانات نصية ويجب تحويلها الى... html ففي المثال السابق قمنا بتحويل البيانات النصية الى صورة أما عن كيفية التحويل في هذا المثال فيوجد بعض الادوات التي تدعم html موجودة على موقع أوراكل و لم نستطيع ان احملها لكوني ساكن في أحد الدول المفروض عليها عقوبات من قبل الويات المتحدة الامركية

أقرأ أيضا :

[تعلم كيفية الاتصال بقواعد بيانات Oracle من خلال تطبيقات الجافا](#)

[الاتصال بقواعد بيانات Mongo من خلال تطبيقات الجافا](#)

[الاتصال بقواعد بيانات PostgreSQL من خلال تطبيقات الجافا](#)

[الاتصال بقواعد بيانات SQLite من خلال تطبيقات الجافا](#)