لغة الجافا:Java Language

هي لغة حديثة أنتجتها عام 1995م شركة SUN Micro System لتناسب التطبيقات الحديثة .
وهي تناسب تطبيقات الإنترنت حيث أصبحت هي قلب برمجة الإنترنت بما توفره من إمكانيات .
وتتصف لغة الجافا بالصفات التالية :

1- لغة برمجية تعمل بواسطة الأهداف OOP.
2- لها بيئة تشغيل خاصة بها . JVM
3- لها مكتبة فصائل . Class Libraries
4- تقوم على لغة . C++ / C
5- تعمل على معظم نظم التشغيل .

وفيما يلي شرح هذه النقاط :

1-لغة تلتزم بقواعد البرمجة بواسطة الأهداف Object Oriented Programming (OOP) : حيث وفرت كثير من الجهد الذي كان يبذل باستخدام البرمجة التقليدية ، حيث كانت البرمجة التقليدية توفر للمبرمج مكتبة من الدوالي إضافة إلى تركيب تقليدي للبرنامج وعلى المبرمج أن يستعمل الدوالي مع تركيب البرنامج لإنشاء التطبيقات مما يضطره لكتابة السطور الكثيرة أكثر من مرة ؛ لقد كانت وحدة بناء البرنامج هي الدالة .. في حين أتت البرمجة بواسطة الأهداف بفكرة جديدة هي إنشاء عناصر متكاملة تحتوي على بيانات ودوالي هي أساس إنشاء البرنامج .. وبالتالي أصبحت وحدة بناء البرنامج وحدة كبيرة هي الفصيلة أو العنصر Object مما سهل واختصر الكثير .

2-لغة لها بيئة تنفيذ خاصة : JVM للغة الجافا بيئة تشغيل للبرنامج هي JVM التي تقوم بترجمة البرنامج للغة الآلة وبالتالي فإن لغة الجافا غير مرتبطة بنظام التشغيل .

3-لها مكتبة فصائل قوية: Class Libraries نظراً لأن لغة جافا تعتمد على مفهوم OOP فهي تحتوي على مكتبة فصائل قوية توفر معظم أو كل الفصائل المطلوبة للإعمال مثل التعامل مع الملفات وقواعد البيانات والشبكات و

الرسومات المجسمة والحركة وكذلك التعامل مع الإنترنت .

4-لغة مبنية على لغة الـ C,C++ : فعندما تم إنشاء لغة الجافا كان أساس بنائها لغة من أشهر وأقوى اللغات وهي C,C++ وبالتالي فهي لم تبدأ من حيث بدأ الآخرون بل من حيث انتهى الآخرون وهي لغة C++ و ثم إضافة الجديد في لغة الجافا .

تعريفات هامة :

1-بيئة تشغيل الـ JVM جافا .
2-مترجم برنامج . JIT
3- Java Applet .
4- Java Application تطبيق جافا .
5-مكتبة. SDK , JDK

وفي مايلي شرح هذه التعريفات :

1-بيئة التشغيل : "JVM"

الحروف JVM اختصار للعبارة JAVA Virtual Machine , وهي فكرة قامت جافا بإمشائها لتجعل لغة جافا تعمل على جميع أو معظم أنظمة التشغيل . وتقوم الفكرة على إنشاء طبقة وسيطة Software كأنها برنامج تشغيل للبرامج RunTime لكل نظام تشغيل يتم إنزاله أولاً على الأجهزة بحيث تفهم هي برامج جافا وتفسرها لنظام التشغيل ثم الجهاز ولهذا كان من مزايا لغة جافا أنها تعمل على كثير من نظم التشغيل الموجودة بعد إعداد JVM الخاصة بمعظم أنظمة التشغيل .. فلا يهم إذا كان البرنامج مكتوب لنظام التشغيل WINDOWS أو UNIX , المهم أن البرنامج يكتب ثم يحمل إلى الجهاز وعلى الجهاز يوجد JVM للنظام الموجود وبالتالي يعمل البرنامج .

2- Java Applet :

نوع من أنواع التطبيقات الذي صمم خصيصاً للإنترنت حيث يقوم المطوّر بإعداد هذا البرنامج Applet ثم يستدعيه من خلال إستخدام ملف HTML بشرط تحميل برنامج Applet على الخادم server الموجود عليه ملف الـHTML . أما طريقة إنشاء Applet وطريقة إستدعائها من داخل ملف HTML فهذا ما سنتعلمه إن شاء الله خلال الدروس القادمة .

3- Java Application تطبيق الجافا :

هو تطبيق يشبه التطبيقات المنشأة بجميع لغات البرمجة الأخرى يعمل مع نظام التشغيل بعيداً عن شبكة الإنترنت والمشهور عن لغة جافا أنها تعدّ برامج للإنترنت ولكن غير المشهور أيضاً أنها توفر كثير من نقاط القوة في إعداد أي تطبيق سواء مكتبي DISKTOP أو خاص بالشبكات CLIENT\SERVER .

طرق كتابة برامج الجافا :

توجد أكثر من طريقة لكتابة برامج الجافا وترجمتها منها :

(1)استعمال المكتبة JDK مباشرة مع استعمال أي محرر سطور:

تعتبر هذه الطريقة التقليدية هي استعمال أدوات JDK التي أنتجتها شركة SUN مع أي محرر سطور لإعداد البرنامج وهي الطريقة المتبعة عند شرح أجزاء لغة جافا ونبدأ كما يلي :

الأدوات المطلوبة لإستعمال هذه الطريقة :

1/ محرر سطور وليكن " NoteBad المفكرة " الموجود مع ويندوز .

2/ مجموعة : JDK ويمكنك الحصول على مكتبة JDK من موقع SUN من هنا :J2SE download.

3/ أدوات المجموعة : JDK

- الملف : Javac وهو الملف التنفيذي المستعمل في ترجمة الملف المصدر إلى الصورة التنفيذية .
- الملف : Javaهو البرنامج النسئول عن تنفيذ برامج Java التنفيذية بعد تحويلها .
- الملف : Applet Viewer لعرض برنامج Applet للإختبار.

(2)استعمال برامج وسيطة مثل KAWA أو : JCreator
يمكنك إنزالها من هنا :. Download JCreator

(3)استعمال البرامج المعدّة للغة الجافا مثل , Forte : JBuilder :
يمكنك إنزالها من هنا : Download Borland JBuilder , Download Forte .

وفيما يلي توضيح بسيط لهذه الطرق :

(1)استعمال المكتبة JDK مباشرة وهي الطريقة التقليدية وذلك بإعداد مكتبة جافا أو مترجم لغة جافا على جهازك وهي JDK : Java Developmenet Kit والتي توفرة شركة SUN مجاناً على موقعها هنا : JAVA Development Kit.

(2)استعمال برامج وسيطة : وهي برامج معدّة لتسهيل كتابة برنامج لغة الجافا ولكنها ليست بيئة متكاملة . حيث توفر كتابة البرنامج وتنفيذه من خلال بيئة رسومية بشرط

وجود مكتبة ومترجم جافا مسبقاً على الجهاز JDK يوجد برامج كثيرة لهذا الغرض مثل : برنامج Creator وكذلك Kawa .

(3)البرامج المعدّة للغة جافا : وهي أفضل وأقوى الطرق وذلك بإستعمال أحد البرامج التي تطلق عليها برامج visualحيث تتوفر جميع متطلبات إعداد تطبيق جافا مثل : البيئة السهلة للإعداد والمكتبة والمترجم ، ومن هذه البرامج الشهيرة Forte وهو من إنتاج شركة SUN وكذلك برنامج visual café وبرنامج الـ JBuilder وبرنامج الـ Java Developer .

مكونات : (Javax.swing) Swing GUI

إن المكونات swing هي مجموعة غنية من تحكمات واجهة تداخل المستخدم الرسومية ؛ كتبت من أجل أن تبدو وتتصرف بشكل نظام التشغيل نفسه على جهاز المستخدم . على العكس من مكونات AWT فالأخيرة لا تعتمد على مكونات GUI الأصلية ؛ إن زر AWT سيبدو مثل زر ويندوز على جهاز ويندوز ، زر ماكنتوش على كمبيوتر ماكنتوش ، وهكذا ..
تستخدم مكونات SWING نموذج الحدث نفسه مثل مكونات AWT و JavaBeans مع أن هذه المكونات تعرّف بعض الأحداث الجديدة .
تستطيع إنزالها من هنا :. Javax.swing

**ما الفرق بين Java وJavaScript **

الفرق بينهما شاسع .... فهما لغتان مختلفتان تماما او لنقل لغة و سكريبت ...

لكن قبل ان نقوم بالتفصيل هذه بعض الفروق الاساسية بينهما:

- 1لغة الـ Java يكون النص البرمجي لها "code" محفوضا في ملفات متعددة بينما في JavaScript فان نصها البرمجي مضمن ضمن صفحات الـ HTML التي تحتويها.

- 2لغة الـ JavaScript تستخدم لتعزيز قدرة الـ HTML على التحكم بطريقة عرض المعلومات على الشاشة بينما باستخدام Java يمكن عمل "برامج" منفصلة كطبيقات سطح المكتب او حتى الـ .... applets او عمل الربط بين العميل و الخادم Client/Server مثل PHPاوASP

- 3يمكن دائما رؤية و تعديل نص الـ JavaScript من خلال اي برنامج تحرير نصوص ومن ثم حفظ الملف بصيغة html و عرضه على اي متصفح انترنت .... بينما برامج الـ Javaتحتاج الى عمل compile ومن ثم interpret لها قبل "تنفيذها".

البعد التاريخي:

الـ Java بدأت كمشروع صغير عام 1995 في معامل شركة SUN Micro System الضخمة باسم green projectوكان الهدف منها لتحكم بالاجهزة مبدئيا الى ان تطورت واصبحت كما نرى الآن .... اما الـ JavaScript فهي لغة منتجة من قبل شركة Netscape و هي تعتبر scripting languageو قد قامت NetScape بشراء الاسم

Javaمن sun لكي يساعد في تسويق لغتهم الجديدة .....

اهم مميزات الجافا:

-يمكن لاي برنامج معمول بلغة الجافا ان يعمل بشكل مباشر على اي framework بمعنى ان البرنامج يمكن ان يعمل على Windows Xp او Linux او Mac على عكس امكانيات لغات البرمجة الاخرى مثل C++ او حتى.C#

-هي الرائدة في تقنية الـ OO او برمجة المتجهات و تعتبر اكثر لغة تطبق الفكرة كاحد مميزاتها الجبارة.

البرامج المعمولة بالجافا تنقسم الى 3 انواع رئيسية

1- Appletsوهي كائنات تعرض في صفحات الانترنت "ويمكن ان تنفصل عنها لتظهر في نافذة مستقلة" و توفر تفاعل على مستوى عالى مع المستخدم و اشهر امثلة عليها برامج المحادثة الشهيرة في Yahoo او Digichat المشهورة في المواقع العربية

امثلة :

) يجب ان يكون جهازك يحتوى على(Java VM

2- Applications

يمكن ان تنتج الجافا برامج تعمل على سطح المكتب مستقلة تماما مثل برامج تحرير النصوص (عملت واحدا بنفسي !!) شبيهه جدا بالنوتباد مثلا او حتى متصفحات الانترنت

او مثل هذا البرنامج الذي يستخمد كعارض للصور

http://www.cs.umd.edu/hcil/photomesa/

3-تطبيقات الانترنت web applications

هنا نجد البرامج تعمل بوصل العميل مع الخادم تماما مثل لغة PHP و ASP و تدعم قواعد البيانات و غيرها الكثير

وافضل مثال هو موقع sun نفسه ....

أسئلة و اجوبة عامة لكل مبتدئ في الجافا

س1: ماهي بدايات لغة الجافا ؟؟

لغة الجافا هي من تطوير شركة صن المعروفة وكانت في البداية جزء منها مكتوب ب سي ++ وسي اما الان فهي مكتوبة من اولها الى اخرها بلغة الجافا.
طبعا اهم ما يميز الجافا انها لا تعتمد على (platform) معين لانها تعمل على آلة الجافا الافتراضية JVM لذلك هي مستقلة عن طبيعة platform وهذا هو السبب الرئيسي لإنتشارها الواسع جدا.

س2:هل الجافا هي الجافا سكريبت ؟

لا
يخلط كثير من المبتدئين بين هاتين اللغتين ولكن دعوني اوضح ماهي الجافا سكريبت في البداية التي هي من انتاج شركة نتسكيب وكانت في البداية تسمى live code و اخذت تسمية الجافا سكريبت لسبب تجاري فقط لان الجافا كانت في اشد انتشارها.
الجافا سكريبت هي لغة بسيطة جدا مقارنة بالجافا فهي لغة Client Side أي انه يتم تنفيذها على متصفح الويب فالكود يرسل من السيرفر بدون معالجة ,حيث انه يعالج على جهاز الزبون (.client)

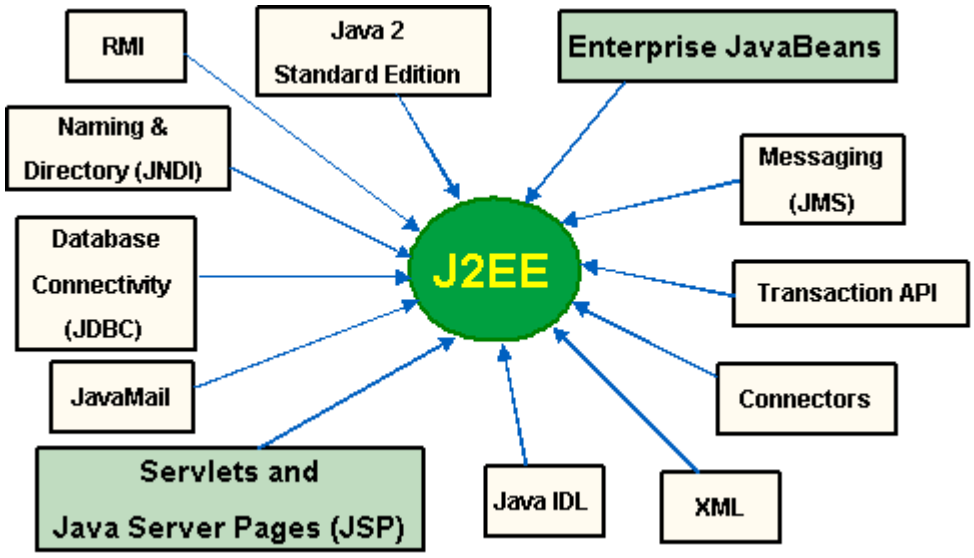
س3: ماهي الجافا ؟

توجد ثلاث نسخ للجافا وهي:
J2EE و J2SE و. J2ME لن ندخل في تفاصيلها ولكن سوف اتكلم على J2EE لانها هي الي عليها الكلام اما J2ME فهي تخص ال (wireless devices) بشكل عام يعني على اجهزة الجوال وغيرها .

س4:ماهي J2EE ؟

هي إختصار ل Java 2 Enterprise Edition وهي تزودنا بالتطبيقات الكبيرة على مستوى الشركات الكبيرة وهي

تحتوي على تقنيات كثيرة (حول 12 تقنية) أنظر الصورة الي اسفل سوف توضح لك أكثر.



س5:ماهي مترجمات (IDE OR Compilers) للجافا ؟

يوجد هناك العديد منها ولكنني شخصيا أفضل Jbuilder من شركة بورلاند المرموقة والان يوجد النسخة التاسعة منة فهو افضل ماجربت في الحقيقة, ولكنه يحتاج الى ذاكرة رام كبيرة لكي يعمل بصورة جيدة (طبعا عيبة البطء لانة مكتوب كامل بلغة الجافا والجافا عيبها انها بطيئة نوعاً ما ولكن مع تطور الاجهزة ربما يندثر هذا العيب). طبعا هناك ايضا forte من صن وغيرها كثير.

س6:ماهو مدى انتشار الجافا خصوصا في الوطن العربي ؟

جواب:

في الحقيقة الجافا منتشرة في الدول المتقدمة بصورة كبيرة جدا وخصوصا في (امريكا وبريطانيا) ولكن مع الاسف فهي عالمنا العربي قليلة الانتشار وان كانت الجامعات أدخلتها ولكن منذ مدة بسيطة نسبياً.

س7:ماذا عن دعم الشركات العملاقة للجافا ؟

في الحقيقة أن ميكروسفت تحارب الجافا وهذة حقيقة مع الاسف فيمكن تلاحظون كثيرا ان الي مركبين ويندوز اكس بي يعانون من مشكلة عدم توفر JVM على الويندوز الذي كان في نسخة السابقة توجد فيه.

طبعا صن رفعت قضية على مايكروسفت وفازت بها في المحاكم الامريكية ووعدت مايكروسفت انها في النسخ القادمة سوف توفرها في الويندوز لكي لايضطر المستخدمين الى تحميلها من الانترنت.
ولكن هناك شركات عملاقة تدعم الجافا بكل تفاني مثلا عندك الاوراكل وكلكم يعلم ان Form 9i مبني على تقنية الجافا ابليت وايضا اي بي ام (أم الشركات) ومايكروميديا وغيرها كثير.

س8:ماذا عن تقنيات الويب في الجافا ؟
في الحقيقة تكمن قوة الجافا الحقيقة في الشبكات بشكل عام وتراسل البيانات التي مع الاسف لاارى له حضور حتى على مستوى الجامعات اللهم القليل فقط.
الجافا تقدم عدد من التقنيات اهمها(طبعا في مجال الويب) :
JSP

APPLET

SERVLET

EJB

JAVABEAN

طبعا ابليت كما هو معروف هو كلاينت ساید فهو يرسل الكود من السيرفر ويعالج لدى الكلاينت اما البقية فهم سيرفر ساید حيث تتم معالجة الكود في السيرفر وترسل النتائج الى العميل.

س9: ماهي ال JSP ؟
هي لغة تمكنك من بناء صفحات انترنت ديناميكية وهي تشبة asp حقت مايكروسفت في المفهوم فقط .
طبعا JSP لغة بسيطة جدا تمكنك من دمج رموز ال HTMLمع رموز JSP بحيث يمكنك من دمج كود جافا .

س10: ماهو Servlet ؟
جواب:

باختصار ال جي اس بي هي امتداد لل سيرفلت يعني فقط تسهيلا لكتابة JSP وضعوا ال جي اس بي بالطبع هناك بعض الفروقات التي لا مجال لذكرها هنا ولكن افضل شي هو تكاملهما مع بعض .

س11: ماهو javaBeans ؟

جواب:

هي كلاسات جافا لها بعض الشروط البسيطة تخدم مع الويب وفائتها الاساسية تقليل كتابة الكود بحيث يمكن استخدامها في اكثر من صفحة.

EJP هي بالمناسبة مكونات تكون موزعة بحيث تستطيع الشركة من خلالها تكامل الانظمة والتحكم في مستوى الامن والانسيابية.

س12:ماهي سيرفرات الجافا على الويب؟

جواب:

اهمها و الكبار منها (غير مجانية)

IBM WbSphere -1وهو بالطبع بيئة إحترافية بحد ذاتها وغنية بالمزايا والان يوجد النسخة5.1

Oracle9iAs -2وهو من اسرع السيرفرات وهو من تطوير اوراكل .

Bea WebLogic -3وهو من السيرفرات المرموقة وحاليا توجد النسخة 8.1 أخرى(مجانية)

Tomcat-1وهو مجاني ومفتوح المصدر من شركة اباتشي المعروفة وهو المفضل لدى الكثيرين و النسخة الحالية هي 5.1

وغيرهم كثير .

س13:ماهي الصعوبات الي تواجة مطوري الجافا في العالم العربي ؟

جواب:

في الحقيقة دعوني أجيب على هذا السؤال من خلال تجربتي الشخصية:

كنت قد نويت أن أعمل مشروع عبارة عن شركة للسيارات من خلال الانترنت وكنت حينها لا أعلم عن الويب شي (فقط القليل من asp) قررت أن أستخدم

الجافا وخصوصا ان خبرتي في الجافا ابليكيشن والحمد لله جيدة. المهم نزلت أجوب مكتبات الرياض ومع الاسف لم أجد حتى الكتب التي أريدها لتعلم لغة JSP في حين ان رفوف الكتبة مليئة بكتب . ASP.NET دبرهتا بطريقة خاصة ومن بعض المواقع على النت الحمدلله استطعت اني انهي مشروعي ولكن السلبيات التي مريت بهاالامور :

-1حصولي على الكتب في منتهى الصعوبة .

-2مشاريع كبيرة تمت كتابتها بتقنيات مايكروسفت, حصلوا على دعم كبير بحيث ان الشركة توفر لهم الكتب المطلوبة والنصائح من المختصين لدى الشركة وهذا لم احصل علية لانني كتبت مشروعي بالجافا.

-3عدم انتشار اللغة بصورة كبيرة بحيث كل زملائي الذين اعرفهم كتبوا بلغات دوت نت لذلك لم استطع الاستفادة منهم.

-4عندما حاولت إستضافة موقعي مع الاسف لم أجد المناسب وذلك لقلة شركات استضافة الجافا(العربية).

ولكن هذا لايعني ان الجافا لغة ليست جيدة بل هي منتشرة جدا جدا في الدول المتطورة ولكن هي قليلة الانتشار في الوطن العربي , مع انني سعيد جدا لكتابة مشروعي بها فهي لغة رائعة حقا وقوية جدا.

س14: هل اتعلم الجافا او NET. ؟جواب:

إجابة هذا السؤال تعتمد عليك أنت فإذا كنت تحب البرمجة وتعشق التحدي والمغامرة والجافا تناسب إحتياجك فعليك بها(لاتريد أن تعتمد على platform معين.(

إما إذا كنت خلاف ذلك فمن رأيي أن تتوجة الى الدوت نت فهي توفر لك كامل الدعم وهي بكل تاكيد أسهل من الجافا لانني كما ذكرت هناك الكثير من الكتب والكثير من الدعم والكثير من المحررات العملاقة و.. يكفي أنها من مايكروسفت.

وعموما الخلاصة ماذا تريده أنت من اللغة هو الذي يحدد إختيارك لها!

س15:أريد أن أتعلم الجافا ولكن لا أردري أي الكتب أفضل ؟؟

بالنسبة للمبتدئين والمتوسطين أنصحهم بكتاب Java How To Programمن شركة Deitel™ فهو بحق كتاب رائع جدا ومناسب وسهل الاسلوب جدا (طبعا الكتب باللغة الانجليزية).متوفر في جرير الان(الطبعة الخامسة).
بالنسبة لل Java database فهناك كتاب لم أرى قط مثلة حتى الان وهو كتاب رائع جدا جدا ولكن لازم تكون عندك خبرة بالجافا قبل ماتقراه وهو Java Database Programmingمن Bible ورقم ال ISBN هو 0-7645-4924-3.
بالنسبة للويب :
Web Development with Java Server Pagesالطبعة الثانية
Core Servlets & Java Server Pagesمن كتب شركة صن وهو كتاب رائع.

س16:هل الجافا صعبة ؟؟
هذا سؤال في الحقيقة أيضا يعتمد عليك فأصعب مافي الجافا بدايتها فقط ,فأذا تجاوزت البداية وأحببت اللغة فيكون الباقي سهل باذن الله.

س17: ودي اتعلم جافا بس وين المواقع الجيدة ؟؟؟
جواب:
المواقع الي سوف أذكرها كلها باللغة الانجليزية.

www.java.sun.com هذا هو أهم المواقع فهذا هو الموقع الرسمي للغة الجافا وفية الكثير والكثير من الدروس وأنصحك بالمنتديات (Forums)ستجد فيها ثروة بما تعنيه هذة الكلمة.

www.javaworld.com هذا مجلة عالم الجافا المشهورة وفي ارشيف هذة المجلة العديد من الدوس والمقالات المميزة وهي تعنى بآخر أخبار مايستجد للجافا من أخبار ومنتجات جديدة.

مدخل إلى البرمجة الشيئية أو الكائنية التوجّه Object Oriented Programming

خلال الأسطر القليلة التالية، سنلقي الضوء على مفهوم البرمجة الكائنية أو الشيئة Object Oriented Programming وهي ما يطلق عليه اختصاراً OOP، ماهيتها ومميزاتها.

فكّر بالكائنات:Think about Objects

ستتعرف في هذا الجزء على أهم المصطلحات المستخدمة في الـ OOP كما ستفهم فكرة الـ OOP إن شاء الله!
لو نظرنا حولنا في عالمنا الحقيقي لوجدنا جميع ما يحيط بنا عبارة عن" كائنات :"Objects الناس، الحيوانات، النباتات، السيارات، الطائرات، البنايات، وحتى الكمبيوترات وغيرها. هذا هو معنى كلمة "كائن "Object ومن الممكن أن نطلق نفس المصطلح على أي ممثل لأي فئة، فنطلقه على الفراولة لأنها تمثل أحد الفواكة، أو نطلقه مثلاً على الطاووس لأنه يمثل أحد الطيور... وهكذا.
ويمكننا تصنيف الكائنات إلى صنفين:

- كائنات نشطة (حية) :Animate Objects وهي التي نحس فيها فنجد لها حركة ونشاط.
- كائنات غير نشطة (غير حية) :Inanimate Objects هي التي لا نلاحظ لها نشاط أو حركة أو وقع أينما وجدت.

وجميع الكائنات بصنفيها لها:

1. خصائص Attribute مثل: الحجم، اللون، الوزن، الشكل...ألخ.
2. سلوك Behavior فمثلاً: الطفل (كائن) يبكي، وينام، ويمشي، ويأكل (سلوكيات).

الإنسان وخصوصاً المبرمج يتعلم عن الكائنات بمعرفة خصائصها، وملاحظة (تجربة) سلوكها، فمن الممكن أن يكون لكائنات مختلفة نفس الخصائص وسلوك متقارب.
البرمجة الشيئية Object Oriented Programming تقوم بنمذجة Modeling كائنات العالم الحقيقي في برنامج نظير software counterpart. هذا البرنامج يحمل إيجابيات العلاقات بين الفئات classes relationships حيث أن أي كائن من أي فئة يحمل جيمع مميزات وصفات characteristicsهذه الفئة أو بالأحرى يرثها لأنه ممثل لفئته. كما أن الفئات الجديدة -تسمى فئة فرعية -*subclass*ترث صفات الفئات التي أُنتجت وتكونت منها - تسمى الفئة الأم -*superclass* كما يرث الطفل جينات أبويه. وهذه الفئة الجديدة والتي تعتبر *subclass*، من الممكن أن تكون *superclass* لفئات جديدة أخرى ينشئها المبرمج.
الـ OOPكذلك تقوم باحتواء البيانات (Data (attributes والطرق (Methods (behaviorفي حزمة package هي ما نطلق عليه "كائنات"Objects ؛ حيث أن بيانات وطرق أي كائن ترتبط ببعضها ارتباط وثيق. هذا الكائن يتميّز بخاصية التخفي Information Hiding نعني بالتخفي هنا أنه بإمكان الكائنات الاتصال والتعامل مع بعضها البعض مع عدم معرفة أحدها كيف تكوّن الآخر! أي أن تفاصيل التكوين هي المخفيّة حتى عن الكائنات نفسها؛ فمن المؤكد أننا نعرف كيف نقود السيارة بكفاءة عالية دون معرفة تفاصيل هندستها. تسمى هذه الخاصية في البرمجة بـ Abstractionأي تجريد البيانات.
برامج الجافا جميعها قائمة على برمجة المبرمج لمجموعة فئات خاصة به تسمى user-defined classes باستخدام الفئات والمميزات التي توفرها اللغة ومن ثم استخدام هذه الفئات جميعها أو بعضها في برامجه(:
حيث أن كل فئة تحتوي على بيانات data ومجموعة دوال functionsتقوم بتشكيل هذه البيانات، تسمى البيانات في فئات الجافا بـ: *instance variable* أو *data member*. ويطلق علي الدوال اسم الطرق *methods*. فأي طلب لأي فئة معرّفة في اللغة كأنواع البيانات مثل int يسمى

"متغير variable"، بينما طلب أي فئة من الفئات التي عرّفها المبرمج user-defined يسمى "كائن object".

البرمجة الشيئية أو الكائنية Object Oriented Programming:

عند حديثنا عن البرمجة الشيئية، نجمل الحديث في كلمتين: الوراثة وتعدد الأشكال Inheritance & Polymorphism، وهما من التقنيات الفعّالة للتعامل مع البرمجيات المعقدّة:

- فالوراثة inheritance هي شكل للبرامج software المعدّة للاستعمال مع الفئات classes الحديثة والتي أنشئت من فئات موجودة مسبقاً وأخذت عنها خصائصها وسلوكها وأضافت إليها القدرات التي نحتاج إليها في هذه الفئة الجديدة. الوراثة ماذا تعني عملياً؟! تعني بالضبط ما الذي تم وراثته و كيف يمكن التعديل عليه وما الذي لا يمكن وراثته -يتضح ذلك بالأمثلة-. هذه الخاصية توفر الكثير من الوقت للمبرمج وتقطع عنه أشواطاً في تطوير برنامجه.
- وتعدد الأشكال polymorphism يسمح لنا بكتابة برنامجنا في صورة قابلة لتغيير واسع النطاق؛ سواء كان التغيير لفئات موجودة مسبقاً أو تغيير مستقبلي لإنتاج برامج جديدة. هذه الخاصية تسهل علينا توسيع قدرات نظامنا.

وكما ذكرنا في الأعلى أن الفئات الجديدة -تسمى فئة فرعية- *subclass* ترث صفات الفئات التي أنتجت وتكونت منها -تسمى الفئة الأم- *superclass* كما يرث الطفل جينات أبويه. وهذه الفئة الجديدة والتي تعتبر *subclass*، من الممكن أن تكون *superclass* لفئات جديدة أخرى ينشئها المبرمج. وهكذا تمتد لدينا سلسلة من الوراثة بين الفئات، extends يحكمها قانون "الوراثة المفردة Single Inheritance" حيث ينص هذا القانون على:

تنشأ أي فئة فرعية من فئة أم واحدة، فالجافا لا تدعم التوارث المتعدد multiple inheritance كالسي++ ولكنها تدعم مفهوم الواجهات Interfaces، فنظام الواجهات

يساعد الجافا على تحقيق فائدة التوارث المتعدد مع عدم وجود الأخطاء المترابطة الناتجة عن هذا التوارث المتعدد!
تذكر أن أي كائن ينتمي إلى فئة فرعية فهو ينتمي إلى الفئة الأم لهذه الفئة الفرعية ويحمل خصائصهما وسلوكهما.
وبعد هذه المقدمة وهذا التوصيف لعالم الـ OOPنلاحظ أن جُلّ التركيز في هذا النوع من البرمجة يقع على الـفئات Classes، فالمبرمج يستخدم الفئات المبنية مسبقاً في اللغة مع الفئات التي يبنيها هو كي ينتج برنامجاً بالجافا، ربما يفسر هذا الاسم(: OOP

حزم الجافا (Java Packages)

- ماهي حزم الجافا؟
- لماذا نحتاج حزم الجافا؟
- كيف نستطيع انشاء حزم الجافا؟

ماهي حزم الجافا؟

التعريف :حزم الجافا هي مجموعة من الفئات المترابطة، و كل مجموعة من الفئات تنظم تحت حزمة معينة لأجل تحديد الهوية. و الحزمة تتكون من:

- **حزم فرعية تحت الحزمة الأم .**
- **مجموعة من الفئات المتعلقة بالحزمة الأم .**

بعض الأمثلة: الحزمة Java تحتوي على حزم فرعية منها applet, io, alng, net, awt & util و لو أخذنا الحزمة الفرعية Java.awt لحصلنا على حزمة فرعية من awt مثل image و يكون الامتداد لها Java.awt.image

لماذا نحتاج حزم الجافا؟

مبرمجي الجافا يعتمدون على الحزم لتكوين فئات مترابطة داخل هذه الحزم و الأسباب هي:

- **العثور على الفئات بشكل سريع و استخدامها بالبرامج.**
- **تنحدر الفئات تحت الحزم لكي لا تتعارض اسماء الفئات مع بعضها البعض.**
- **للتحكم بالفئات بشكل كامل .**

مسميات الحزم و الحزم الفرعية و الفئات: الحزمة تتكون من حزم فرعية و فئات متفرعة، لكن لا نستطيع تسمة الحزمة أو الحزم الفرعية أو احدى الفئات باسم واحد. و

مثال على ذلك :الحزمة java.awt لديها حزمة فرعية بالاسم .image لكن لا نستطيع تسمية احدى الفئات بالاسم ،image لأن الاسم محجوز للحزمة الفرعية و العكس صحيح.

كيف نستطيع انشاء حزم الجافا؟

لنرى مجموعة من الفئات و التي نستطيع وضعها في حزمة معينة. نفترض اننا كتبنا فئات عن النقاط و الدائرة و المستطيل و المربع.

```
public class Point {
    int_x coord;
    int_y coord;

   public Point() {
      x_coord = 0;
      y_coord = 0;
   }
   public Point(int x, int y) {
      x_coord = x;
      y_coord = y;
   }
}
```

```
public class
Rectangle
extends Point
{
   double width;
   double height;

   public
Rectangle(int x,
int y, double w,
double h)
   {
      super(x, y);
      width = w;
      height = h;
   }
}
```

```
public class Circle extends
Point
{
   double radius;

   public Circle(int x, int y,
double r)
   {
      super(x, y);
```

```
public class
Square extends
Point
{
   double edge;

   public
Square(int x, int
y, double e)
```

```
    {
        edge = e;
    }
}
```

```
        radius = r;
    }
}
```

الآن نود أن نضع هذه الفئات مع بعضها البعض في حزمة لعدة أسباب:

- نستطيع نحن و المبرمجين الآخرين أن نجد هذه الفئات لأنها مترابطة.
- نستطيع نحن و المبرمجين الآخرين أن نعرف كيف نجد هذه الفئات لأنها دوال رسم مترابطة.
- اسماء الفئات السابقة لن تتعارض مع اسماء الفئات من الحزم الاخرى لأنها سوف تكون تحت حزمة جديدة من انشائك، مثال على ذلك :

```
package geometry;

public class Rectangle extends Point
{
    double width;
    double height;

    public Rectangle(int x, int y, double w, double h)
    {
        super(x, y);
        width = w;
        height = h;
    }
}
```

```
package geometry;

public class Point {
    int_x coord;
    int_y coord;

    public Point() {

x_coord = 0;

y_coord = 0;
    }
    public Point(int x, int y) {
```

	x_coord = x;
	
	y_coord = y;
	}
	}

package geometry;
 	
public class Square extends Point
{
double edge;
 	
public Square(int x, int y, double e)
{
edge = e;
}
}
	package geometry;
	
	public class Circle extends Point
	{
	double radius;
	
	public Circle(int x, int y, double r)
	{
	
	super(x, y);
	radius = r;
	}
	}

نلاحظ هنا اننا اضفنا السطر package geometry في كل الفئات) كل فئة توجد في ملف مستقل). لكن لو فرضنا اننا نريد استخدام الفئة Rectangle موجودة بالحزمة

java.awtمع الفئة الموجود بالحزمة geometry بنفس البرنامج الذي نريد كتابته، فماذا نفعل ؟

استدعاء فئتين بنفس المسمى: نستطيع ذلك باستخدام fully qualified nameو هو كتابة المسار الكامل للفئة، مثال على ذلك:

```
java.awt.Rectangle rec1 =
new java.awt.Rectangle(...);
// المسار الكامل للفئة استخدمنا

geometry.Rectangle rec2 =
new geometry.Rectangle(...);
// و هنا ايضاً
```

كيفية استدعاء فئة معينة من الحزمة الخاصة بها:

تستطيع استدعاء الفئات من الحزم عن طريق ثلاث طرق:

- **استدعائها عن طريق كتابة المسار الكامل) كما المثال السابق).**
- **استدعائها فقط عن طريق الحزمة java.awt.Rectangle**
- **استدعاء الحزمة كاملة بما فيها من فئات اخرى java.awt.*;[1].**

[1]النجمة (*) تدل على استدعاء الحزم الفرعية و الفئات الموجودة تحت هذه الحزمة.

المصفوفة المتناثرة أو مصفوفة الأصفار (Sparse Matrix) :

لو ألقينا نظرة على المصفوفة التالية التي تحوي 6 صفوف و6 أعمدة وتتكون من : 6 x 6 = 36 عنصر :

	col0	col1	col2	col3	col4	col5
row0	15	0	0	22	0	-15
row1	0	11	3	0	0	0
row2	0	0	0	-6	0	0
row3	0	0	0	0	0	0
row4	91	0	0	0	0	0
row5	0	0	28	0	0	0

Figure1:Matrix

c4arab.com

سيتضح لنا من الوهلة الأولى أن أكثر عناصر هذه المصفوفة عبارة عن " أصفار " ؛ تسمى المصفوفة التي أكثر عناصرها أصفار بـمصفوفة الأصفار أو المصفوفة المتناثرة " Sparse Matrix " . ومن الصعب علينا تحديد ما إذا كانت المصفوفة عبارة عن Sparse Matrix أو لا .. ولكن يتضح لنا ذلك عن طريق النظر؛ ففي المصفوفة السابقة يوجد فقط 8 عناصر لا تساوي الصفر من أصل 36 عنصر ،بينما البقية كلها أصفار.
نعالج الـ Sparse Matrix بالتكنيك الذي سنشرحه في درسنا لتوفير المساحة في الذاكرة حيث نستطيع تخزين العناصر الغير مساوية للصفر فيها فقط ؛ وذلك من خلال استخدام مصفوفة وحيدة لكل عنصر من عناصرها يوجد 3 صفات هي : الصف والعمود والقيمة الخاصة به؛ ويتم ذلك

عن طريق استخدامنا لـ Classكالتالي :

```
package Sparse_Matrix;

public class SM extends Object {
    private int row , col , val ;

//------------<< SM constructor >>----------
--------------
    public SM (int row1 ,int col1 ,int val1
) {
        setrow(row1) ;
        setcol(col1) ;
        setval(val1) ;
        }

//------------<< Method to perform row >>---
--------------
    public int setrow( int row1 ){
        return row = row1 ;
    }
//------------<< Method to perform column
>>--------------
    public int setcol( int col1 ){
        return col = col1 ;
    }
//------------<< Method to perform value >>-
--------------
    public int setval( int val1 ){
        return val = val1 ;
    }
//------------<< Method to Print >>---------
--------------
    public String Print() {
        return ( row +"\t"+ col +"\t"+ val
+"\n") ;
```

```
        }
}
```

ولكن يجب أن نراعي هنا أن ترتيب العناصر في هذه المصفوفة الوحيدة سيكون تابع لأحد هذه الصفات وهو الصف " row " و لابد من أن يكون تصاعدياً ..

إذن .. يتضح لدينا أن تعريف الـ Sparse Matrix كما هو موجود في قاموسنا كالتالي :
مصفوفة ذات بعد واحد تحوي الكثير من العناصر المتشابهة ، والتي غالباً ما تساوي صفر، لكل عنصر فيها ثلاث صفات : الصف ، العمود ، والقيمة التي تسند إليه .

*** Sparse_Matrix : a set of triples < row , col , value > , where row & column are integers & from a unique combination , & value comes from the set item .**
*** Sparse_Matrix Create(*max_row , max_col*) ::= return a Sparse_Matrix that can hold up to *max_item = max_row X max_col* ,& whose maximum row size is *max_row* , & whose maximum column size is *max_col*.**

ومن هنا نستطيع إعادة رسم الـ Sparse Matrix كما في الشكل التالي :

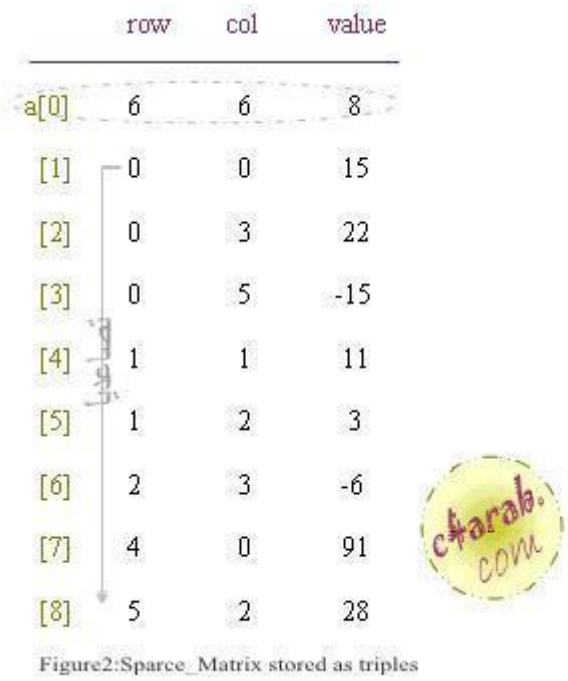


Figure2:Sparce_Matrix stored as triples

حيث أن a[0].row تحتوي عدد الأسطر الكلي للمصفوفة الأصلية (في هذا المثال = 6) , كذلك a[0].col فهي تحوي عدد الأعمدة الكلي للمصفوفة الأصلية (في هذا المثال = 6) , وأيضاً a[0].value عدد العناصر الغير مساوية للصفر فقط (في هذا المثال = 8) .
ولكي نعرف رقم الصف لأي عنصر ننظر لـ Field *row* , وبالمثل إذا أردنا أن نعرف رقم العمود فننظر لـ Field *col* وستكون قيمة هذا العنصر مخزنة في Field *value* . والثلاثي < row , col , value > سيكون مرتب في المصفوفة على حسب الصفوف " تصاعدياً " كما ذكرنا سابقاً ..
ولكن كيف نستطيع كتابة شيفرة لإنشاء هذه المصفوفة بلغة الجافا ؟ هذا ما سنعرفه ان شاء الله خلال الأسطر التالية :

(1في البداية نقوم بعمل ملف ونسمّيه: Sparse_Matrix مثلاً كما اعتمدنا في هذا المثال.
(2ثم نقوم بعمل كلاس نسميّه SM ونخزّنه في الملف السابق الذكر بعد أن نعمل فيه package Sparse_Matrix ونكتب في هذا الكلاس الكود السابق.

(3ثم نقوم بعمل Application ونضع فيه إستدعاء للكلاس السابق أي .. import Sparse_Matrix.SM :
ونكتب فيه شيفرة لمعالجة المصفوفة المتناثرة بالتسلسل التالي:

- **سنفترض في مثالنا الحالي أن المصفوفة مكوّنة من 3 صفوف و3 أعمدة ..**
- **وبعد ذلك نسمح للمستخدم بإدخال العناصر كمصفوفة عادية شريطة أن تكون أكثرها مساوية للصفر ونطبع المصفوفة بالطريقة التقليدية العادية .**
- **ثم نقوم بإنشاء الـ Sparse Matrix ؛ نخزّن في البداية عدد الصفوف الكلي وعدد الأعمدة الكلي في كل من a[0].row و .. a[0].col ثم نضع عداداً = 1 كفهرس لكي نبدأ التخزين ..**
- **نقوم بعمل Loop يمّر على كل عناصر المصفوفة العادية ويسأل ما إذا كان هذا العنصر مساوياً للصفر أما لا ؟**
إذا اتضح أن العنصر لا يساوي الصفر .. نقوم بتخزين رقم الصف الموجود فيه هذا العنصر وكذلك رقم العمود ثم نخزّن قيمة العنصر باستخدام العداد الذي جعلنا قيمته =1 كفهرس لأول عنصر يقابلنا غير مساوي للصفر .. ثم نزيد قيمة العداد بواحد لكي يفهرس العنصر المخزن الجديد .. وهكذا إلى أن ننتهي من جميع عناصر المصفوفة الأصلية .
- **الآن قمنا بتخزين جميع القيم الغير مساوية للصفر في الـ Sparse Matrix ولكن يتبقى Field واحد لم نخزّن به شئ .. أتعلمون ما هو ؟**
إنه الـ Field الخاص بعدد العناصر الغير مساوية للصفر في المصفوفة الأصلية .. (a[0].val)
ونستطيع معرفة عدد العناصر الغير مساوية للصفر من خلال العداد الذي فهرس العناصر..
ولكن نلاحظ هنا أن هذا العداد داخل Loop عندما انتهى التخزين قد زادت قيمته بواحد على عدد العناصر الغير مساوية للصفر؛ فيجب علينا أن نقوم بانقاص قيمته بمقدار واحد ثم نخزنها في a[0].val ...

- الآن نستطيع نقوم بطباعة المصفوفة المتناثرة الناتجة لدينا.

وأليك الشيفرة كاملة:

```
import Sparse_Matrix.SM;
import javax.swing.*;

public class Sparse_MatrixO {
    public static void main (String args[])
    {
    int Matrix [][] ;
    int i , j , q ,count = 1 ;
    SM SparseMatrix ;
    SparseMatrix= new SM(0,0,0);
    String x,output="";
    output+="The Normal Matrix :\n";
    JTextArea outputarea = new
JTextArea(10,20);
    Matrix = new int[3][3];
    //-----------<< To Read & Print Normal
Matrix >>--------------------
    for(i=0 ; i<3 ; i++){
        for(j=0 ; j<3 ; j++){

x=JOptionPane.showInputDialog("plz. Enter
the value of element \n");
            q = Integer.parseInt(x) ;
            Matrix[i][j] = q ;
            output += Matrix[i][j] +"\t" ;
            if ( Matrix[i][j] != 0 )
count++ ;
        } // end of j Loop
```

```
            output += "\n" ;
        } // end of i Loop
        //-----------<< To Make & Print
Sparse_Matrix >>-------------------
        output +="\n The Sparce_Matrix
:\nrow\tcol\tvalue\n-----------------------
---------------------------\n" ;
        SparseMatrix.setcol(3);
        SparseMatrix.setrow(3);
        SparseMatrix.setval(count-1);
        output += SparseMatrix.Print() ;
        for(i=0 ; i<3 ; i++){
            for(j=0 ; j<3 ; j++){
                if ( Matrix[i][j] != 0 ) {
                    SparseMatrix.setcol(j);
                    SparseMatrix.setrow(i);

SparseMatrix.setval(Matrix[i][j]);
                    output +=
SparseMatrix.Print() ;
                } // end of if
            } // end of j Loop
        } // end of i Loop

        outputarea.setText(output);

JOptionPane.showMessageDialog(null,outputar
ea,"Sparse_Matrix",
JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
        }
}
```

طبعاً، في برنامجنا المتواضع أدخلنا العناصر عنصراً عنصراً، وهي طريقة غير عملية بتاتاً مع التطبيقات

الكبيرة، حيث أننا في التطبيقات الكبيرة نخزن المصفوفة في ملف ونقوم بقراءة عناصرها منه.

طريقة البحث الثنائي:Binary Search

يصادف المبرمج دوماً العمل مع كمية بيانات كبيرة مخزنة في مصفوفة، ومن الضروري أن يستخدم تكنيك معين يحدد له ما إذا كان العنصر الذي يبحث عنه key ينتمي إلى هذه المصفوفة أم لا! هذا التكنيك يطلق عليه "البحث" وله عدة أنواع، من أشهرها وأكثرها فاعلية طريقة البحث الثنائي.

ولكي نطبق أحد خوارزميات الـ Binary Searchعلى مصفوفة ما نتبع الخطوات البسيطة التالية:

1. الخطوة الأولى والأهم والتي لا يمكن تطبيق الـ Binary Searchلولاها هي:
   ترتيب المصفوفة تصاعدياً أو تنازلياً أو أبجدياً على حسب نوع البيانات المخزنة فيها!
2. تحديد أول عنصر في المصفوفة ولنسمه i، وآخر عنصر فيها ولنسمه مثلاً j.
3. تحديد العنصر الذي يقع في منتصف هذه المصفوفة ولنسمه k.

بعد ذلك يمكننا تطبيق تكنيك البحث الثنائي على مصفوفتنا، وهناك عدة خوارزميات للبحث الثنائي، سأشرح أحدها في هذا الدرس على مصفوفة ذات بيانات رقمية إن شاء الله. وفي الدرس الثاني سنتعرف على المكتبة الجاهزة والتي توفرها الجافا لتطبيق الـ Binary Searchعلى مصفوفة ذات بيانات حرفية strings بإذن الله.

خوارزم البحث الثنائي:Binary Search Algorithm

تقوم فكرة البحث الثنائي على تقسيم المصفوفة إلى نصفين واستبعاد النصف الذي لا ينتمي إليه المفتاح key الذي نبحث عنه، كيف ذلك؟
عن طريق تحديد العنصر الذي يقع في منتصف هذه المصفوفة، ثم نقارن هذا العنصر مع المفتاح الذي نبحث عنه كالتالي) تذكر أن مصفوفتنا مرتبة تصاعدياً أو تنازلياً):

1. إذا كان يساويه نكون قد وجدنا العنصر الذي نبحث عنه.
2. إذا كانت قيمة المفتاح أقل من قيمة العنصر الأوسط في المصفوفة، إذن نحتاج أن نبحث فقط في نصف المصفوفة الأول ونستبعد البحث في نصفها الثاني.
3. وفيما عدا ذلك: إذا كانت قيمة المفتاح أكبر من قيمة العنصر الأوسط في المصفوفة، إذن نحتاج أن نبحث فقط في نصف المصفوفة الثاني ونستبعد البحث في نصفها الأول.
4. بعد ذلك: نطبق نفس الخطوات من 1 إلى 3 في النصف الجديد الذي نبحث فيه، فنقوم بتقسيمه إلى قسمين، ونقارن المفتاح مع العنصر الأوسط الجديد، بنفس الترتيب الذي ذكر في الخطوات 1 إلى3 السابقة.

سيساعدك المثال التالي على فهم الطريقة إن شاء الله:
نفرض أننا نبحث عن عناصر مختلفة في هذه المصفوفة:
Array[]={0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28}
تابع في هذا الفلاش التسلسل في البحث عن عناصر مختلفة:
والسؤال الآن: كيف نكتب code يمثل هذا الخوارزم بالجافا أو السي؟!
وللإجابة على هذا السؤال سيصادفنا تساؤل آخر: كيف نرتب المصفوفة تصاعدياً او تنازلياً؟!
والإجابة:
لترتيب المصفوفة فهناك عدة خوارزميات للترتيب منها على سبيل المثال (Bubble sort, sorting by Selection, sorting by Insertion, Shell sort, & Quick sort).
ولا مجال لذكرها الآن، حيث سنعتمد في الـ codeعلى ترتيبنا نحن للمصفوفة بشكل صحيح.
والآن، لنستعرض معاً code يطبق تكنيك binary search على مصفوفة ذات عناصر رقمية بلغة الجافا، حيث أن : j=high، i=low, & k=middle

```
// Binary search of an array
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.text.*;

public class BinarySearch extends JApplet
implements ActionListener {
JLabel enterLabel, resultLabel;
JTextField enter, result;
JTextArea output;

int a[];
String display = "";

public void init()
{
Container c = getContentPane();
c.setLayout( new FlowLayout() );

enterLabel = new JLabel( "Enter key" );
c.add( enterLabel );

enter = new JTextField( 5 );
enter.addActionListener( this );
c.add( enter );

resultLabel = new JLabel( "Result" );
c.add( resultLabel );

result = new JTextField( 22 );
result.setEditable( false );
c.add( result );

output = new JTextArea( 6, 60 );
output.setFont(
```

```
new Font( "Courier", Font.PLAIN, 12 ) );
c.add( output );

// create array and fill with even integers 0 to 28
a = new int[ 15 ];

for ( int i = 0; i < a.length; i++ )
a[ i ] = 2 * i;
}

public void actionPerformed( ActionEvent e )
{
String searchKey = e.getActionCommand();

// initialize display string for the new search
display = "Portions of array searched\n";

// perform the binary search
int element = binarySearch( a, Integer.parseInt(
searchKey ) );

output.setText( display );

if ( element != -1 )
result.setText("Found value in element " + element
);
else
result.setText( "Value not found" );
}

// Binary search
public int binarySearch( int array[], int key )
{
int low = 0; // low subscript
int high = array.length - 1; // high subscript
int middle; // middle subscript
```

```
while ( low <= high ) {
middle = ( low + high ) / 2;

// The following line is used to display the part
// of the array currently being manipulated during
// each iteration of the binary search loop.
buildOutput( low, middle, high );

if ( key == array[ middle ] ) // match
return middle;
else if ( key < array[ middle ] )
high = middle - 1; // search low end of array
else
low = middle + 1; // search high end of array
}

return -1; // searchKey not found
}

// Build one row of output showing the current
// part of the array being processed.
void buildOutput( int low, int mid, int high )
{
DecimalFormat twoDigits = new DecimalFormat(
"00" );

for ( int i = 0; i < a.length; i++ ) {
if ( i < low || i > high )
display += " ";
else if ( i == mid ) // mark middle element in
output
display += twoDigits.format( a[ i ] ) + "* ";
else
display += twoDigits.format( a[ i ] ) + " ";
}
```

```
display += "\n";
}
}
```

وإذا أحببت أن تطلع على الكود بلغة السي، تفضل بزيارة الوصلة
التالية
:http://www.c4arab.com/showlesson.php?lesid=1496
أما إذا أحببت أن تطلع على خوارزم آخر للـ Binary search، تفضل بزيارة هذه
الوصلة
:http://www.c4arab.com/showlesson.php?lesid=1498

عدد مرات البحث في أي مصفوفة عن عنصر محدد باستخدام الـBinary Search:

لو تسائلنا عن أقصى عدد من مرات البحث باستخدام الـ Binary Searchفي أي مصفوفة، لوجدنا أنه يُعطى من إيجاد القوة التي يرفع إليها رقم 2 كي يطعينا العدد الذي يزيد عن عناصر المصفوفة بواحد. أي أنه أول قوة لـ2 والتي تُعطي رقم أكبر من عدد عناصر المصفوفة بواحد.
ففي مثالنا: استخدمنا مصفوفة من 15 عنصر، نلاحظ ان العدد الذي يزيد على عدد عناصر المصفوفة بواحد، أي العدد 16 ينتج من القوة الرابعة لرقم2 (2^4=16) وذلك يعني اننا نحتاج على الأكثر لأربع مرات مقارنة في الـ Binary Searchحتى نجد العنصر الذي نبحث عنه! فمن الممكن أن نجده من أول مرة في المقارنة، ومن الممكن أن نجده في ثاني مرة، أو ثالث مرة أو رابع مرة.. أو أن يكون غير موجود في المصفوفة!
وفي مثال آخر: لو بحثنا في مصفوفة تحوي 1024 عنصر، سنحتاج إلى 10 مرات للمقارنة كحد أقصى، ونعرف ذلك بتكرار قسمة عدد العناصر على رقم 2 إلى أن نصل إلى العدد واحد في خارج القسمة (وسبب ذلك هو أننا بعد كل مقارنة نقوم بإلغاء نصف عناصر المصفوفة من الاعتبار)، فبتكرار قسمة 1024 على رقم 2 نحصل على القيم

التالية على الترتيب: 512، 256، 128، 64، 32، 16، 8، 4، 2، ورقم .1 نلاحظ أن العدد 1024 (2^10) قسم على رقم 2 عشر مرات حتى حصلنا على العدد.1
نستنتج من ذلك، أن القسمة على اثنين تقابل مرة واحدة من المقارنة في الـ .Binary Search Algorithm
فمصفوفة بـ 1048576 (2^20) عنصر تستلزم على الأكثر 20 مرة من المقارنة حتى نجد العنصر الذي نبحث عنه، ومصفوفة تحوي بليون عنصر، تستلزم على الأكثر إلى 30 مرة من المقارنة حتى نجد العنصر المطلوب فيها!
ترى، كم يوفر لنا هذا التكنيك من الوقت في البحث؟..
فقط 30 مرة من البحث بين بليون عنصر لنجد ضالتنا!!..
إنه تكنيك عبقري فعلاً :)

البحث الثنائي (Binary Search2)

جميع المكتبات التي سنستخدمها هنا وفرتها الـ Java Collections Framework، وسنستخدم مكتبتين:

- Collections.binarySearch: والتي تأخذ list كـ argument أول لها، وتأخذ Object كـ argument ثاني. فتقوم بالبحث عن الـ Object داخل الـ List باستخدام تكنيك الـ binary search والذي شرحناه في الدرس الأول.
- Collections.sort: والتي تأخذ List كـ argument وحيد لها، وتقوم بترتيبها أبجدياً.

ولكي نستطيع استخدام هذه المكتبات، لابد لنا من أن نضع جميع العناصر الحرفية في list (سواء اخترنا ArrayList, LinkedList or Vector) وسنختار هنا الـ ArrayList. ويوضح الـ code التالي كيفية تعريف هذا النوع من الـ list وكيفية تطبيق هذه المكتبات عليه، حيث أن الـ Method binarySearch التي سنستخدمها ستعطينا في المخرجات:

- موضع المفتاح (العنصر الذي نبحث عنه) في المصفوفة إذا وجد فيها. أي أننا سنحصل هنا على قيمة مساوية للصفر أو أكبرمنه.
- قيمة سالبة إذا لم يكن المفتاح (العنصر الذي نبحث عنه) ينتمي إلى المصفوفة.

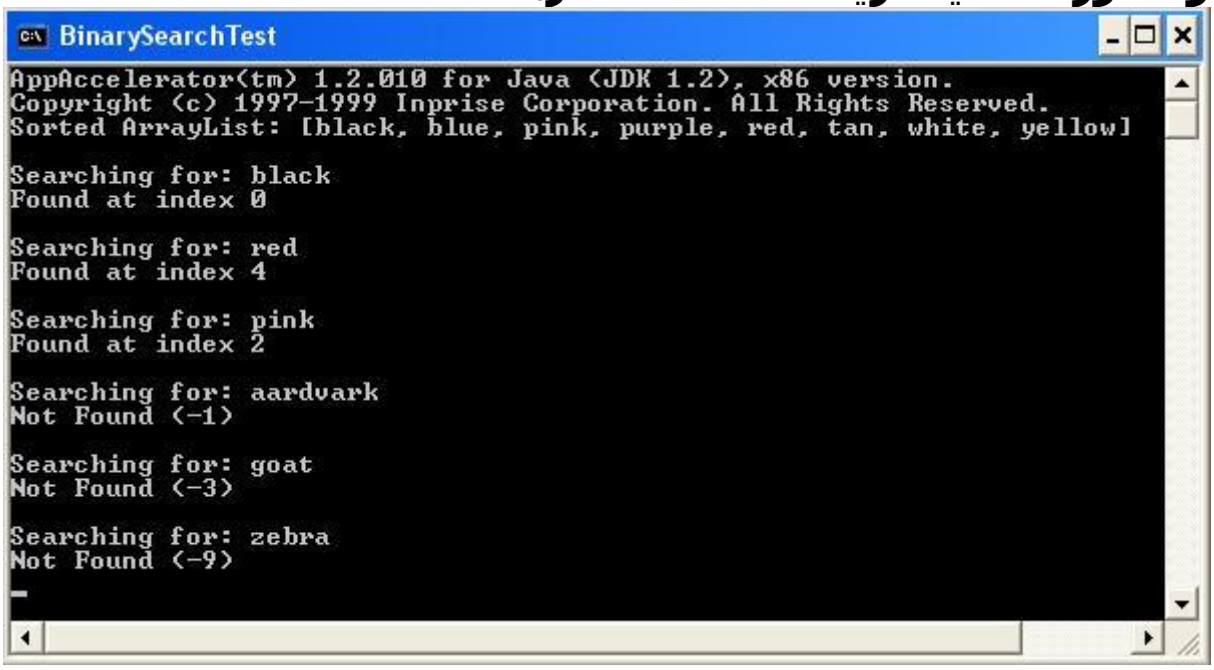
وإليك الشيفرة كاملة:

```
// Using algorithm binarySearch
import java.util.*;

public class BinarySearchTest {
private String colors[] = { "red", "white",
"blue","black", "yellow", "purple", "tan", "pink" };
```

```
private ArrayList aList; // ArrayList reference

public BinarySearchTest()
{
aList = new ArrayList( Arrays.asList( colors ) );
Collections.sort( aList ); // sort the ArrayList
System.out.println( "Sorted ArrayList: " + aList );
}

public void printSearchResults()
{
printSearchResultsHelper( colors[ 3 ] ); // first
item
printSearchResultsHelper( colors[ 0 ] ); // middle
item
printSearchResultsHelper( colors[ 7 ] ); // last
item
printSearchResultsHelper( "aardvark" ); // below
lowest
printSearchResultsHelper( "goat" ); // doesnt exist
printSearchResultsHelper( "zebra" ); // doesnt
exist
}

private void printSearchResultsHelper( String key )
{
int result = 0;

System.out.println( "\nSearching for: " + key );
result = Collections.binarySearch( aList, key );
System.out.println( ( result >= 0 ? "Found at index
" + result : "Not Found (" + result + ")" ) );
}

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

```
new BinarySearchTest().printSearchResults();
}
}
```

والصورة التالية تريك شاشة المخرجات:



وبذلك نكون قد غطينا موضوع الـ binary searchفي الجافا من جميع الجوانب ولله الحمد(:

كيفية استخدام UndoManager بالبرامج

اي شخص يحتاج الى Undo و Redo لاحد برامجه،، يستطيع من خلال هذا الدرس تطبيق الطريقه و استخدامها.

بالبداية: اضف الباكيج للبرنامج:

```
import javax.swing.undo.*;
```

ثانياً: نقوم بنسخ الكود التالي للبرنامج:

```
//UNDO AND REDOACTION CLASSES
//THIS PASRT OF CODE WAS TAKEN FROM THE NOTEPAD DEMO FOUND IN THE JDK1.4.1 DEMO DIRECTORY
class UndoAction extends AbstractAction{
public UndoAction(){
super("Undo", new
ImageIcon("images/undo.gif"));
setEnabled(false);
}
public void actionPerformed(ActionEvent e){
try{
undo.undo();
}
catch (CannotUndoException ex){
System.out.println("Unable to undo: " + ex);
ex.printStackTrace();
}
update();
redoAction.update();
}
protected void update(){
```

```
if(undo.canUndo()){
setEnabled(true);
putValue("Undo",
undo.getUndoPresentationName());
}
else{
setEnabled(false);
putValue(Action.NAME, "Undo");
}
}
}
class RedoAction extends AbstractAction{
public RedoAction(){
super("Redo", new
ImageIcon("images/redo.gif"));
setEnabled(false);
}
public void actionPerformed(ActionEvent e){
try{
undo.redo();
}
catch (CannotRedoException ex){
System.out.println("Unable to redo: " + ex);
ex.printStackTrace();
}
update();
undoAction.update();
}
protected void update(){
if(undo.canRedo()){
setEnabled(true);
putValue("Redo",
undo.getRedoPresentationName());
}
else{
setEnabled(false);
```

```
putValue(Action.NAME, "Redo");
}
}
}
```

ثالثاً: نقوم بانشاء الاوبجكت للبرنامج:

```
//for using undo & redo
UndoManager undo = new UndoManager();
UndoAction undoAction = new UndoAction();
RedoAction redoAction = new RedoAction();
```

اخيراً: نقوم اضافة الـ UndoAction و الـ redoAction للـ Menu كأنه MenuItem و سوف تستطيع استخدامها بشكل تلقائي:

```
MenuName.add(undoAction);
ManuName.add(redoAction);
```

****للتوضيح .. اذا كان المنيو مافي اي صور .. نستبدل هذا الكود:**

```
super("Undo", new
ImageIcon("images/undo.gif"));
```

بهذا الكود:

```
super("Undo");
```

و نفس الطريقه مع كلاس.. RedoAction

استخدام الهتمل (HTML) بمكونات الـSwing

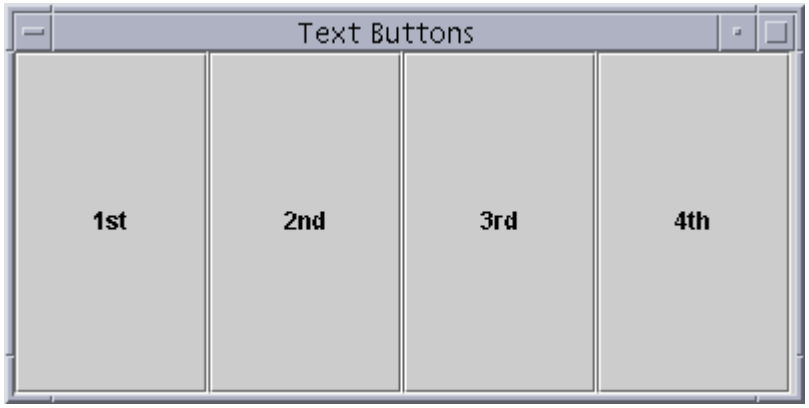
تستطيع استخدام و عرض الهتمل (HTML) في معظم أدوات الـ Swing، سوف نقوم بعرض الطريقة عن كيفية عرض الهتمل (HTML) في بعض الأدوات مثل الـ JLabel و الـJButton.

لنبدأ بمثال بسيط بدون استخدام الهتمل (HTML):

الكود:

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.GridLayout;

public class TextButtons extends JFrame {
    TextButtons() {
        super("Text Buttons");
        setSize(400,200);
        getContentPane().setLayout(new GridLayout(1,4));
        getContentPane().add(new JButton("1st"));
        getContentPane().add(new JButton("2nd"));
        getContentPane().add(new JButton("3rd"));
        getContentPane().add(new JButton("4th"));
    }
    public static void main(String[] args) {
        new TextButtons().setVisible(true);
    }
}
```

و هنا تستطيع رؤية نتيجة البرنامج

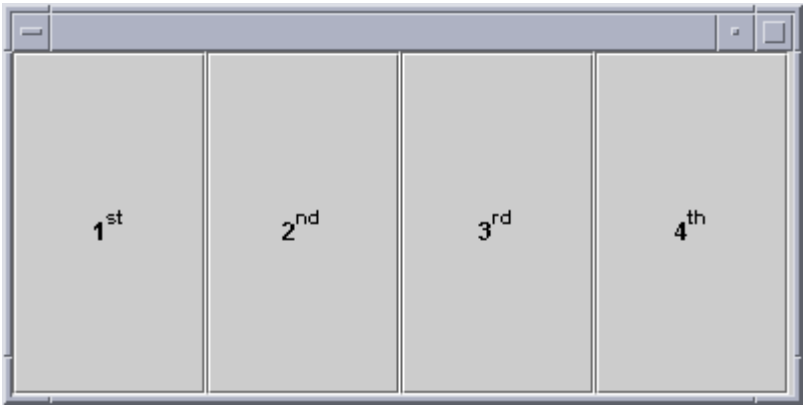


سوف تلاحظ أن البرنامج سليم و قام بالمطلوب لكن هل تريد عرض الـ st, nd, rd & th على أعلى الرقم مثل .1^{st} تستطيع عمل ذلك من خلال وسم الهتمل (HTML).

الكود:

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.GridLayout;

public class TextButtons extends JFrame {
    HTMLButtons() {
        setSize(400,200);
        getContentPane().setLayout(new GridLayout(1,4));
        getContentPane().add(new JButton(
        "<html><p>1<sup>st</sup></p></html>"));
        getContentPane().add(new JButton(
        "<html><p>2<sup>nd</sup></p></html>"));
        getContentPane().add(new JButton(
        "<html><p>3<sup>rd</sup></p></html>"));
        getContentPane().add(new JButton(
        "<html><p>4<sup>th</sup></p></html>"));
    }
    public static void main(String[] args) {
        new HTMLButtons().setVisible(true);
    }
}
```

و هذه هي النتيجة:



تستطيع عمل المزيد من خلال الهتمل (HTML)، (ككتابة النصوص الطويلة بعدة سطور و اضافة الخطوط و الخ.

بالمثال التالي سوف نقوم بعرض مثال عن أهمية استخدام الهتمل (HTML) (مع مكونات الـSwing.

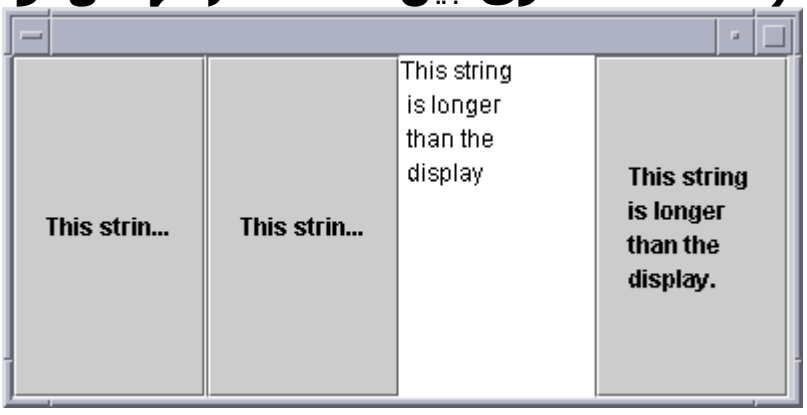
سوف نستخدم النصوص الطويلة لإظهار النتائج بالشكل المطلوب:

- الزر الأول بنص طويل.
- الزر الثاني بنص طويل و تقسيم النص على عدة سطور باستخدام .\n
- منطقة لكتابة النصوص الطويلة و باستخدام .\n
- الزر الثالث و سوف نقوم باستخدام الهتمل (HTML).

الكود:

```
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JTextArea;
import java.awt.GridLayout;

public class LongNames extends JFrame {
    LongNames(){
        setSize(400,200);
        getContentPane().setLayout(new GridLayout(1,4));
        getContentPane().add(new JButton(
            "This string is longer than the display."));
        getContentPane().add(new JButton(
            "This string \n is longer \n than the" +
                                    "\n display"));
        getContentPane().add(new JTextArea(
            "This string \n is longer \n than the" +
                                    "\n display"));
        getContentPane().add(new JButton(
            "<html> This string is longer than the " +
                                "display. </html>"));
    }
    public static void main(String[] args) {
        new LongNames().setVisible(true);
    }
}
```

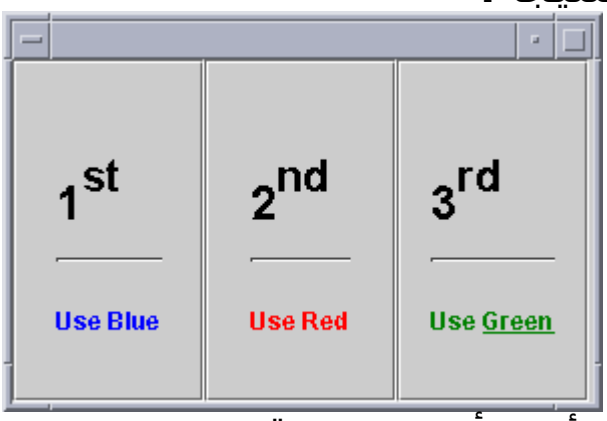
و هنا تستطيع مشاهدة النتيجة، فإذا قمت بتكبير الاطار ستلاحظ الفرق بين استخدام الهتمل او بدونه. (Frame)



تستطيع عمل المزيد و المزيد باستخدام الهتمل (HTML).
سوف تشاهد كيفية كتابة النصوص بجميع التأثيرات عليها.
الكود:

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.GridLayout;

public class MoreHTMLButtons extends JFrame {
    MoreHTMLButtons() {
        setSize(300,200);
        getContentPane().setLayout(new GridLayout(1,3));
        getContentPane().add(new JButton(
            "<html><h1>1<sup>st</sup></h1>" +
            "<hr> <p color=blue> Use Blue</p></html>"));
        getContentPane().add(new JButton(
            "<html><h1>2<sup>nd</sup></h1>" +
            "<hr> <p color=red>Use <b>Red</b></p></html>"));
        getContentPane().add(new JButton(
            "<html><h1>3<sup>rd</sup></h1><hr>" +
            "<p color=green> Use <u>Green</u></p></html>"));
    }
    public static void main(String[] args) {
        new MoreHTMLButtons().setVisible(true);
    }
}
```

و هذه هي النتيجة:



سوف تصبح الأمور أسهل اذا قمت باستخدام لوحات الأسلوب الساقط (Cascading Style Sheets (مع الهتمل (HTML).(مثال ذلك: تستطيع انشاء style sheet باسم jbutton.css.
الكود:

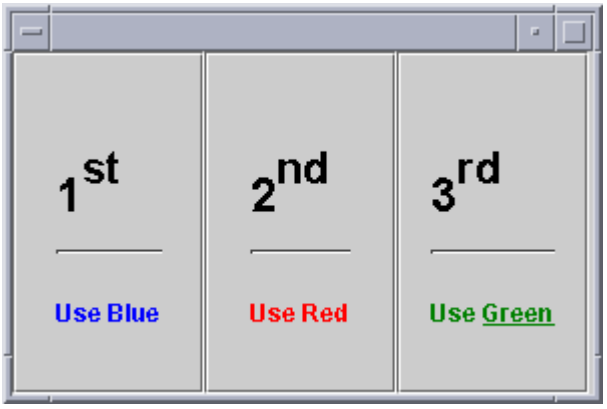
```
p {
   color: red;
}
```

و بعد ذلك نقوم بكتابة كود الجافا -يجب عليك تحديد اسم و مكان الملف باستخدام: -()getResource

الكود:

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.GridLayout;

public class StyleSheets extends JFrame {
    private static final String HTML_HEAD = "<head>" +
        "<link rel=STYLESHEET TYPE=\"text/css\" HREF=\"" +
        StyleSheets.class.getResource("jbutton.css") +
        "\"></head>";

    StyleSheets() {
        setSize(300, 200);
        getContentPane().setLayout(new GridLayout(1, 3));
        getContentPane().add(new JButton(
            "<html><h1>1<sup>st</sup></h1>" +
            "<hr> <p color=blue> Use Blue</p></html>"));

        getContentPane().add(new JButton( "<html>" +
            HTML_HEAD +       // reference the style sheet
            "<h1>2<sup>nd</sup></h1>" +
            "<hr> <p>Use <b>Red</b></p></html>"));

        getContentPane().add(new JButton(
            "<html><h1>3<sup>rd</sup></h1><hr>" +
            "<p color=green> Use <u>Green</u></p></html>"));
    }
    public static void main(String[] args) {
        new StyleSheets().setVisible(true);
    }
}
```

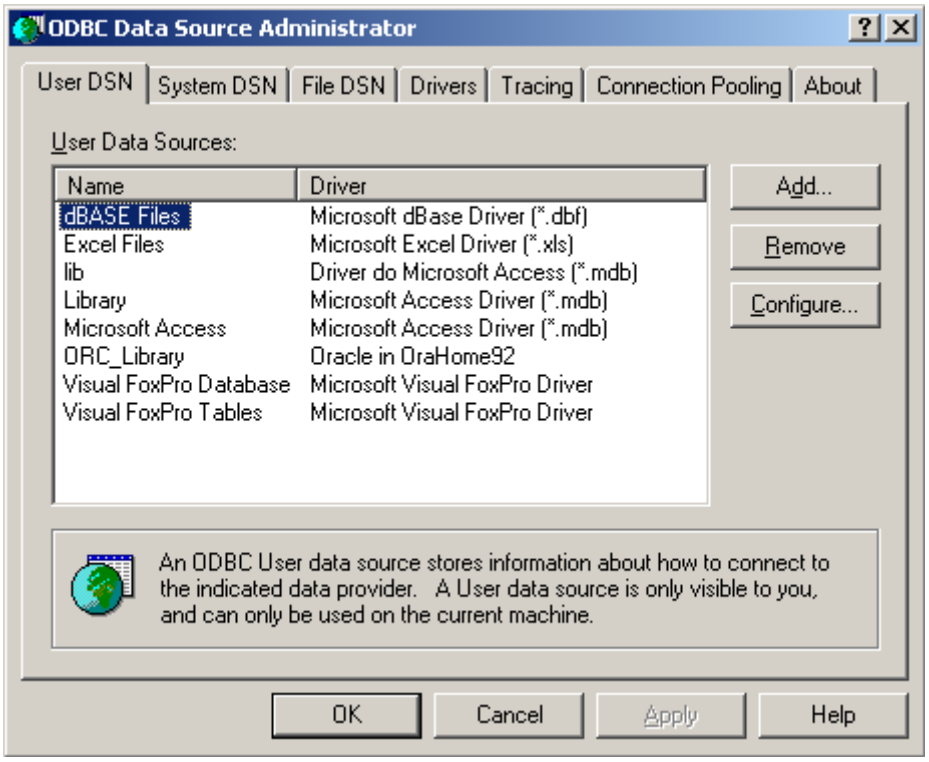
النتيجة:



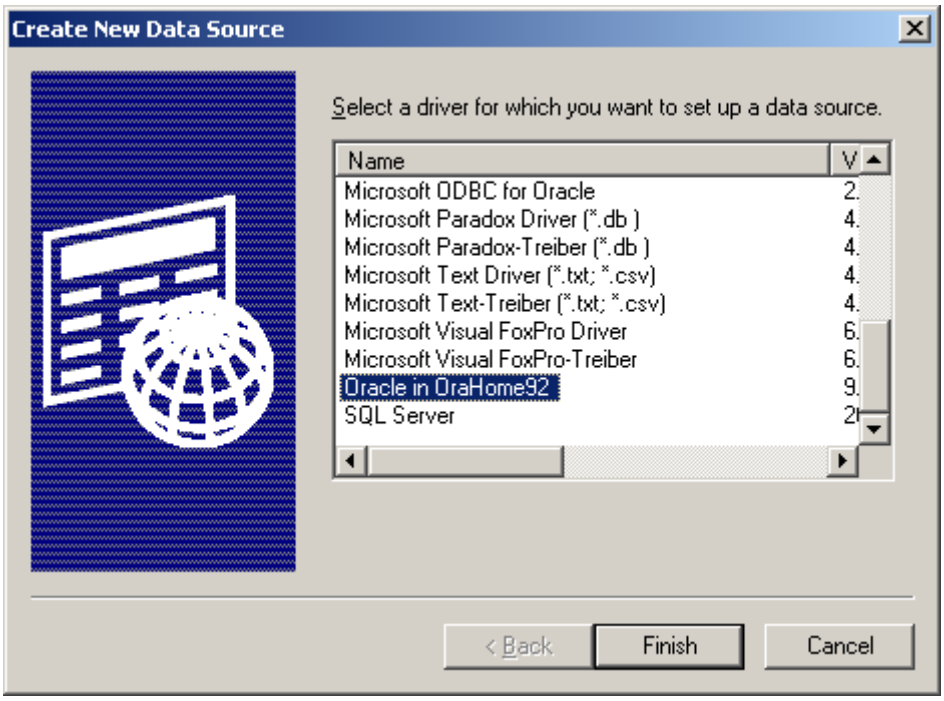
ربط قاعدة بيانات اوراكل بالجافا

سوف نقوم بعملية ربط الجافا بقاعدة البيانات اوراكل..
سوف نستخدم Windows XP و Oracle9i database
بالدرس،

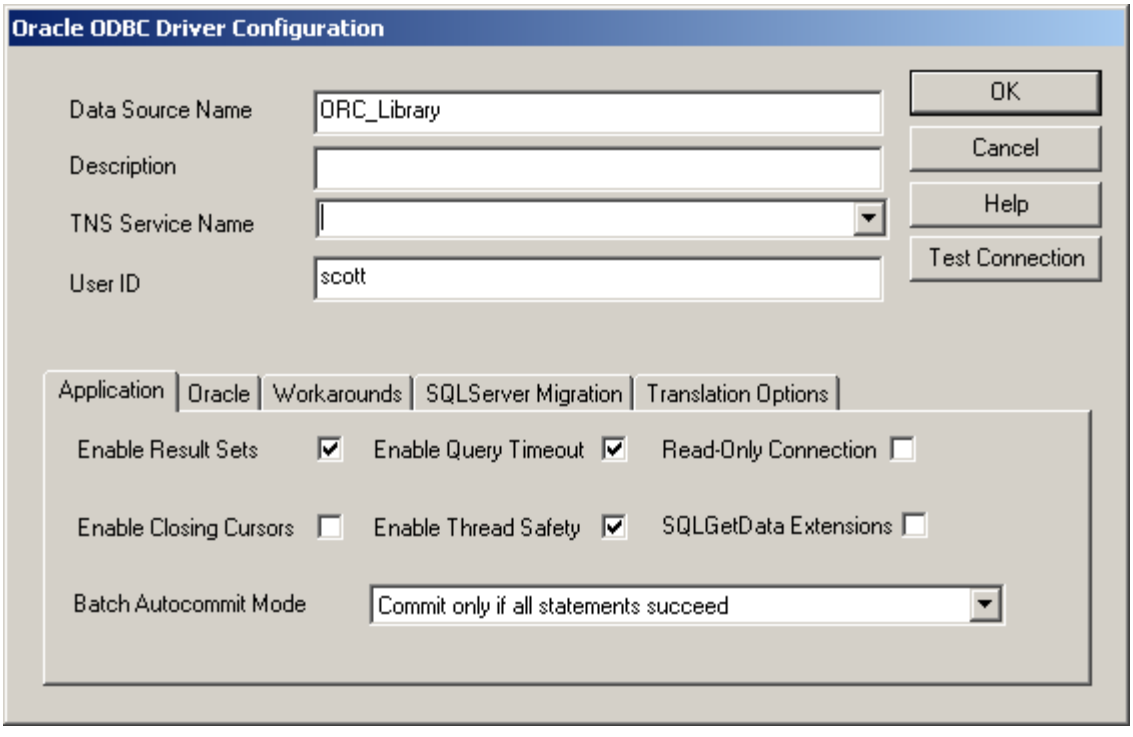
اولاً: نقوم بفتح ODBC Data Source Administrator من لوحة التحكم



ثانياً: نقوم بإضافة User Data Sources عن طريق الزر Add، سوف تجد لائحة و بنهايتها سوف تجد Oracle in OraHome92 و هو الدرايفر المطلوب. قم بالضغط على Finish.



ثالثاً: سوف تظهر نافذة الـ Oracle ODBC Driver Configuration و منها نقوم بكتابة الـ Data Source Name و اسم المستخدم User ID، و قم بعمل Test Connection للتأكد من الاعدادات المطلوبة. و اخيراً قم بالضغط على OK للموافقة على الدرايفر.



و هذا مثال تطبيقي

```
import java.sql.*;
public class Oracle{
 public static void main(String[] args){
  Connection connection;
  Statement statement;
  ResultSet result;
  try{
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
  }
  catch(Exception e){}
  try{
   connection =
DriverManager.getConnection("jdbc:odbc:ORC_Lib
rary","scott","tiger");
   statement  = connection.createStatement();
   result  = statement.executeQuery("select * from
members");
   while(result.next()){
    System.out.println(result.getString("firstname")
+ " " +                result.getString("lastname"));
   }
   result.close();
   statement.close();
   connection.close();
  }
  catch(SQLException e){
   System.out.println(e.toString());
  }
 }
```

}

بالسطر التالي:

```
connection =
DriverManager.getConnection("jdbc:odbc:ORC_Lib
rary","scott","tiger");
```

ORC_Library: يجب يكون الاسم المكتوب في Data Source Name.
scott,tiger هم اسم و كلمة مرور المستخدم لقاعدة البيانات.

بالسطر التالي:

```
result  = statement.executeQuery("select * from
members");
```

هنا تكتب استعلام الـ SQL

بالسطر التالي:

```
System.out.println(result.getString("firstname") +
" " + result.getString("lastname"));
```

هنا تطبع النتائج الموجودة في العمود firstname و العمود lsatname. (يجب اختيار جدول معين لديك فرضاً EMP Table).

اظهار الجداول من الجهه اليمنى(JArabicTable)

في هذا الدرس سنتعلم ان شاء الله كيفية اظهار الجداول من الجهة اليمنى.

لعمل ذلك نقوم بوراثة الكائن JTable ثم ننشأ الرسام الخاص بنا الذي يطبق دوال TableCellRenderer

الفئة JArabicTable تقوم بتحديد الرسام الخاص بها وهو ArabicCellRenderer: في هذه الحالة

```
public class JArabicTable extends JTable {
 public TableCellRenderer getCellRenderer(int row,
int column) {
  return new ArabicCellRenderer();
 }
}
```

الان نقوم بتطبيق الرسام وما يهمنا هنا هو ان نحدد اتجاه الخانات:

```
class ArabicCellRenderer implements
TableCellRenderer {
 public Component
getTableCellRendererComponent(JTable
table,Objectvalue,boolean isSelected,boolean
hasFocus,int row,int column) {
  final JLabel l = new JLabel();
  l.setText(value.toString());
  if(isSelected) {
   l.setBackground(new Color(68,134,250));
  }
  else {
   l.setBackground(Color.white);
  }

l.setComponentOrientation(ComponentOrientation.
```

```
RIGHT_TO_LEFT);
  return l;
 }
}
```

الان قم باستعمال JArabicTable بدل JTable بالبرامج.

تخزين البيانات الضخمة بقاعدة البيانات اوراكل بواسطة الجافا

هل لديك مشروع و تريد تخزين و استرجاع البيانات من صور و مقاطع صوتية و مرئية بقاعدة بيانات Oracle؟

هذا الدرس سوف تقوم من خلاله بعمل المطلوب. و سوف نقوم باستعراض الدرس بالشفرات البرمجية المطلوبة لفهم الموضوع على أكمل وجه (و الكمال لله تعالى).

الـ BLOB و الـ CLOB: و هي اختصار لـ Binary Large Object و اختصار لـ Character Large Object. من ناحية أخرى، هذان النوعان صمما لحفظ و استرجاع البيانات الضخمة بقاعدة البيانات. و التي تصل سعة التخزين إلى أربع جيجابايت (4 GB) من البيانات. و هما من الأنواع التي توجد بالـ SQL3، و التي هي مدعومة من الـ JDBC 2.0. الـ JDBC 2.0 تخدم نوعان جديدان و هما الـ BLOB و الـ CLOB للتعامل مع الأنواع الجديدة من الـ SQL3. و بهذا الدرس سوف نستخدم الـ BLOB لتخزين و استرجاع الصور من و إلى قاعدة البيانات.

أولاً سوف نقوم بإنشاء جدول جديد باستخدام SQL Plus أو بأي طريقة أخرى و نسميه demo:

```
CREATE TABLE demo (id NUMBER(6), image
BLOB);
```

كما تلاحظ يوجد لدينا عمودان. الأول و هو مخصص لإضافة رقم خاص لكل صورة و هو من نوع NUMBER، و العمود الثاني و هو لتخزين البيانات (مثل: الصور و المقاطع الصوتية و المرئية) و هو من نوع BLOB.

سوف نقوم باستعراض أربع أمور:

- تخزين البيانات بقاعدة البيانات.
- استرجاع البيانات من قاعدة البيانات و حفظها على هيئة ملفات.
- حذف البيانات من قاعدة البيانات.
- إظهار البيانات و استخدامها ببرامج الجافا.

تخزين البيانات بقاعدة البيانات:
الآن سوف نقوم بكتابة شفرة برمجية بلغة الجافا لتخزين الصور بقاعدة بيانات Oracle.

الشفرة البرمجية:

```
import java.io.*;
import java.sql.*;
import oracle.sql.BLOB;
import oracle.jdbc.driver.*;

public class WriteBlob {
     private Connection con;
     private String url =
"jdbc:oracle:thin:@127.0.0.1:1521:ORCL2";
     private String user = "scott";
     private String password = "tiger";

     public WriteBlob() {
          try {
            DriverManager.registerDriver(new
oracle.jdbc.driver.OracleDriver());
                con =
DriverManager.getConnection(url, user,
password);
           }
           catch (SQLException e) {
```

```
System.err.println(e.getMessage());
                    e.printStackTrace();
               }
          }
          public static void main(String[] args)
throws Exception, IOException {
               new WriteBlob().process();
          }
          public void process() throws
IOException, SQLException {
               int rows = 0;
               FileInputStream fin = null;
               OutputStream out = null;
               ResultSet rslt = null;
               Statement stmt = null;
               BLOB photo = null; // NOTE:
oracle.sql.BLOB!!!
               long person_id = 0;

               try {
                    con.setAutoCommit(false);
                    stmt =
con.createStatement();
               rslt = stmt.executeQuery("SELECT
               image FROM demo WHERE id = 1 FOR
               UPDATE NOWAIT");
                    while (rslt.next()) {
                         rows++;
                         photo =
(BLOB)rslt.getBlob("image");
                    }
                    rslt.close();
                    rslt = null;
                    /**
                  * If it doesn't exist, then
          insert a row in the information table
```

```
                    * This creates the LOB
locators
                    */
                   if (rows == 0) {
               rows =
               stmt.executeUpdate("INSERT INTO
               demo VALUES (1, empty_blob())");
                        System.out.println(rows
+ " rows inserted");
                        // Retrieve the locator
                        rows = 0;
               rslt = stmt.executeQuery("SELECT
               image FROM demo WHERE id = 1 FOR
               UPDATE NOWAIT");
                        rslt.next();
                        photo =
((OracleResultSet)rslt).getBLOB("image");
                        rslt.close();
                        rslt = null;
                   }
                   stmt.close();
                   stmt = null;
                   // Now that we have the
locator, lets store the photo
                   File binaryFile = new
File("jfc.gif");
                   fin = new
FileInputStream(binaryFile);
                   out =
photo.getBinaryOutputStream();
                   // Get the optimal buffer
size from the BLOB
                   byte[] buffer = new
byte[photo.getBufferSize()];
                   int length = 0;
                   while ((length =
fin.read(buffer)) != -1) {
```

```
                    out.write(buffer, 0,
length);
                }
                // you need to close the
output stream before
                // you commit, or the
changes are lost!
                out.close();
                out = null;
                fin.close();
                fin = null;
                con.commit();
            }
            catch (SQLException e) {
                System.err.println("SQL
Error: " + e.getMessage());
            }
            catch (IOException e) {
                System.err.println("IO
Error: " + e.getMessage());
            }
            finally {
                if (rslt != null) {
                    try {
                        rslt.close();
                    }
                    catch (SQLException
ignore) {}
                }
                if (stmt != null) {
                    try {
                        stmt.close();
                    }
                    catch (SQLException
ignore) {}
                }
                if (out != null) {
                    try {
```

```
                              out.close();
                        }
                        catch (IOException
ignore) {}
                  }
                  if (fin != null) {
                        try {
                              fin.close();
                        }
                        catch (IOException
ignore) {}
                  }
            }
      }
      protected void finalize() throws
Throwable {
            if (con != null) {
                  try {
                        con.close();
                  }
                  catch (SQLException ignore)
{}
            }
            super.finalize( );
      }
}
```

و الآن سوف نقوم بشرح مبسط للشفرة البرمجية السابقة:

```
private String url =
"jdbc:oracle:thin:@127.0.0.1:1521:ORCL2";
private String user = "scott";
private String password = "tiger";
```

هنا سوف نقوم بتحديد نوع الـ Driver و بالمثال السابق استخدمنا thin driver و قمنا بتحديد مكان قاعدة البيانات و هي موجودة بالجهاز المحلي و تحديد المنفذ و هو

1521 و حددنا اسم الـ SID لقاعدة البيانات و هـي ORCL2 (يجـب تغييـر المتغيـرات عنـد الحاجـة لـذلك). و قمنـا أيضـاً بتحديد اسم المستخدم و كلمة المرور له.

```
rows = stmt.executeUpdate("INSERT INTO demo
VALUES (1, empty_blob())");
System.out.println(rows + " rows
inserted");
// Retrieve the locator
rows = 0;
rslt = stmt.executeQuery("SELECT image FROM
demo WHERE id = 1 FOR UPDATE NOWAIT");
rslt.next();
photo =
((OracleResultSet)rslt).getBLOB("image");
rslt.close();
rslt = null;
```

بعد ذلك قمنـا بالاتصـال بقاعـدة البيانـات (تسـتطيع قـراءة الـدرس السـابق عـن الاتصـال بقاعـدة البيانـات بموقـع الموسوعة العربية للكمبيوتر و الانترنت أو الفريق العربـي للتكنولوجيا). و من ثم قمنا بإضافة BLOB فـارغ (يجـب أولاً إضافة empty_blob() لقاعـدة البيانـات و اسـترجاعها مـن قاعـدة البيانـات، و بعـد ذلـك نقـوم بإرسـال البيانـات إلـى قاعدة البيانات).

```
// Now that we have the locator, lets store
the photo
File binaryFile = new File("jfc.gif");
fin = new FileInputStream(binaryFile);
out = photo.getBinaryOutputStream();
// Get the optimal buffer size from the
BLOB
byte[] buffer = new
byte[photo.getBufferSize()];
int length = 0;
while ((length = fin.read(buffer)) != -1) {
```

```
    out.write(buffer, 0, length);
}
// you need to close the output stream
before
// you commit, or the changes are lost!
out.close();
out = null;
fin.close();
fin = null;
con.commit();
```

بالجزء السابق من الشفرة البرمجية نحدد اسم الملف (على سبيل المثال: jfc.gif)، و من ثم نقوم بتخزينه بقاعدة البيانات. و بعد ذلك قفل الملف و تنفيذ المطلوب بقاعدة البيانات لكي لا نخسر ما قمنا به.

استرجاع البيانات من قاعدة البيانات و حفظها على هيئة ملفات:
الآن سوف نقوم باسترجاع البيانات (على سبيل المثال: الصورة السابقة) على الجهاز من قاعدة البيانات.

الشفرة البرمجية:

```
import java.io.*;
import java.sql.*;
import oracle.sql.*;
import oracle.jdbc.*;

public class ReadBlob {
     private Connection con = null;
     private PreparedStatement pstmt =
null;
     private ResultSet rs = null;
     private BLOB blob = null;
```

```
	private String selectSQL = "SELECT
image FROM demo WHERE id = ?";
	private String url =
"jdbc:oracle:thin:@127.0.0.1:1521:ORCL2";
	private String user = "scott";
	private String password = "tiger";

	public ReadBlob() {
		try {

Class.forName("oracle.jdbc.driver.OracleDri
ver");
		}
		catch(ClassNotFoundException e) {
			e.printStackTrace();
		}
		try {
			con =
DriverManager.getConnection(url, user,
password);
			pstmt =
con.prepareStatement(selectSQL);
			pstmt.setInt(1, 1); //for
set the ID (line 13)
			rs = pstmt.executeQuery();
			while(rs.next()) {
				blob =
((OracleResultSet)rs).getBLOB("image");
			}
			InputStream blobStream =
blob.getBinaryStream();
			FileOutputStream
fileOutStream = new
FileOutputStream("abc.gif");
			byte[] buffer = new
byte[10];
			int nbytes = 0;
```

```
                while ((nbytes =
blobStream.read(buffer))!= -1) {

fileOutStream.write(buffer,0,nbytes);
                }
                fileOutStream.flush();
                fileOutStream.close();
                blobStream.close();
                rs.close();
                pstmt.close();
                con.close();
           }
           catch(IOException ex) {
                ex.printStackTrace();
           }
           catch(Exception ex) {
                ex.printStackTrace();
           }
     }
     public static void main(String[] args)
throws Exception {
          new ReadBlob();
     }
}
```

سوف نقوم بشرح جزأين من الشفرة البرمجية:

```
con = DriverManager.getConnection(url,
user, password);
pstmt = con.prepareStatement(selectSQL);
pstmt.setInt(1, 1); //for set the ID (line
13)
rs = pstmt.executeQuery();
while(rs.next()) {
     blob =
((OracleResultSet)rs).getBLOB("image");
}
```

نقـوم باسـترجاع الصـورة التـي تحمـل الـرقم المخصـص (على سبيل المثال الرقم: 1).

```
InputStream blobStream =
blob.getBinaryStream();
FileOutputStream fileOutStream = new
FileOutputStream("abc.gif");
byte[] buffer = new byte[10];
int nbytes = 0;
while ((nbytes = blobStream.read(buffer))!=
-1) {
     fileOutStream.write(buffer,0,nbytes);
}
```

باستخدام getBinaryStream() نقـوم باسـترجاع البيانـات كاملة و حفظها على هيئة ملف gif.

حذف البيانات من قاعدة البيانات:
لحذف البيانات بشكل صحيح يجـب علـى المبـرمج إسـناد القيمة التالية:

```
UPDATE demo SET image = empty_blob() WHERE
id = ?
```

إذا قمنا بإسناد NULL لعمود الصور عنـد الحـذف، فسـوف نحصل على NullPointerException عنـد تخـزين البيانـات لنفس العمود مرة أخرى أو نقوم بإضافة empty_blob() و بعد ذلك تخزين البيانات.

إظهار البيانات و استخدامها ببرامج الجافا:
و أخيراً سوف نقوم بعرض البيانات بـالبرامج (علـى سـبيل المثال: عرض الصورة السابقة من قاعدة البيانات).

الشفرة البرمجية:

```
import java.sql.*;
import oracle.sql.*;
import oracle.jdbc.*;

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Main extends JFrame {
     private Connection con = null;
     private PreparedStatement pstmt =
null;
     private ResultSet rs = null;
     private BLOB blob = null;

     private String selectSQL = "SELECT
image FROM demo WHERE id = ?";
     private String url =
"jdbc:oracle:thin:@127.0.0.1:1521:ORCL2";
     private String user = "scott";
     private String password = "tiger";

     private JPanel panel = new JPanel();
     private ImageIcon icon;
     private JLabel imageLabel;

     public Main() {
          super("Main");
          try {

Class.forName("oracle.jdbc.driver.OracleDri
ver");
          }
          catch(ClassNotFoundException e) {
               e.printStackTrace();
```

```
            }
            try {
                  con =
DriverManager.getConnection(url, user,
password);
                  pstmt =
con.prepareStatement(selectSQL);
                  pstmt.setInt(1, 1); //for
set the ID (line 14)
                  rs = pstmt.executeQuery();
                  rs.next();

                  blob =
((OracleResultSet)rs).getBLOB("image");
                  icon = new
ImageIcon(blob.getBytes(1,
(int)blob.length()));

                  rs.close();
                  pstmt.close();
                  con.close();

                  imageLabel = new
JLabel(icon);
            }
            catch(Exception ex) {
                  ex.printStackTrace();
            }

            Container cp = getContentPane();
            panel.add(imageLabel);
            cp.add("Center", panel);
            pack();
            setVisible(true);

            addWindowListener(new
WindowAdapter() {
```

```
                    public void
windowClosing(WindowEvent e) {
                              System.exit(0);
                    }
              });
        }
        public static void main(String args[])
{
              new Main();
        }
}
```

و هنا صورة للبرنامج السابق: