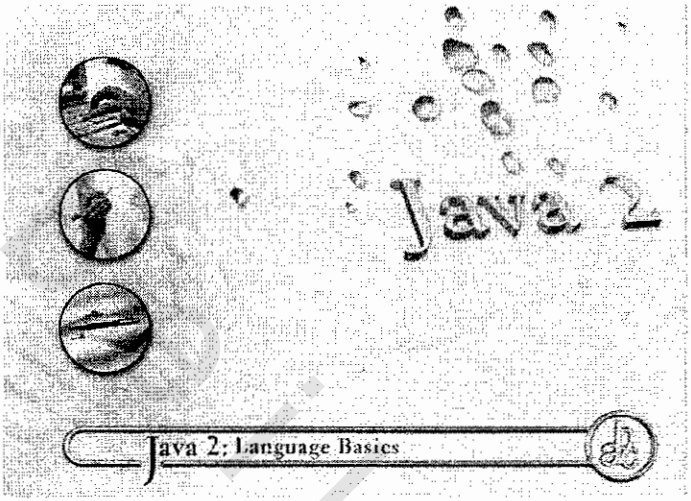
الفصل الأول



مقدمة إلى لغة جافا
Introduction to Java

فى هذا الفصل نبدأ الرحلة مع اللغة الشهيرة Java ونقدم لك التعريفات الأولية التى توفر لك المعلومات اللازمة لتبدأ المشوار وذلك من خلال النقاط التالية:

- مقدمة عن لغة Java.
- مزايا لغة Java.
- ما هى JVM.
- طرق كتابة برامج Java.
- أوامر JDK.
- خطوات إنشاء برنامج Java.
- خطوات إعداد بريمج Applet.
- عرض البريمج Applet داخل صفحة الوب Web Page.
- دورة حياة البريمج Applet.
- البرنامج JDeveloper والبرنامج JCreator والبرنامج NetBeans IDE.

obeikan.com

مقدمة:

- تعتبر لغة Java من اللغات الحديثة التي أنتجتها شركة Sun Microsystems في عام 1991.
- ولغة Java تناسب تطبيقات الإنترنت Internet بصفة خاصة بسبب العديد من المميزات التي تتميز بها لغة Java ، هذا بالإضافة إلي دعم الأنواع المختلفة من التطبيقات مثل تطبيقات تعمل على جهاز واحد Desktop Application أو تطبيقات تعمل على شبكة Client/Server Application.
- وتتصف لغة Java بالصفات التالية:

1- لغة تلتزم بقواعد البرمجة بواسطة الأهداف Object-Oriented Programming (OOP).

2- لغة لها بيئة تشغيل خاصة بها Java Virtual Machine (JVM).

3- لها مكتبة فصائل Classes Library كبيرة وغنية.

4- لغة مبنية على قواعد اللغة الشهيرة C/C++.

5- تعمل برامجها على معظم نظم التشغيل Platform Independent .

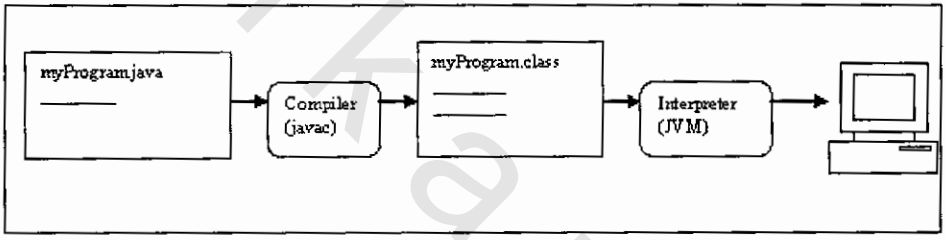
وفيما يلى شرح هذه النقاط.

لغة تلتزم بقواعد البرمجة بواسطة الأهداف OOP:

- تعتبر البرمجة بواسطة الأهداف OOP من الطرق الحديثة للبرمجة التى وفرت الكثير من الجهد الذى كان يبذله المبرمج باستخدام البرمجة التقليدية ، حيث كانت البرمجة التقليدية توفر للمبرمج مكتبة من الدوال Functions وتركيب تقليدى للبرنامج ، وعلي المبرمج أن يستعمل الدوال Functions لإنشاء التطبيقات مما يضطره لكتابة سطور كثيرة وكذلك يضطره لإعادة كتابة السطور أكثر من مرة ، فكانت وحدة بناء البرنامج هى الدالة Function. فى حين أتت البرمجة بواسطة الأهداف OOP بفكرة جديدة هى إنشاء عناصر متكاملة تحتوى على بيانات ودوال Functions ، وبالتالي أصبحت وحدة بناء البرنامج وحدة كبيرة هى الفصيلة Class مما سهل واختصر الكثير من الوقت والجهد في كتابة البرامج ، وسوف نلخص قواعد البرمجة بواسطة الأهداف OOP فيما بعد.

لغة لها بيئة تنفيذ خاصة JVM:

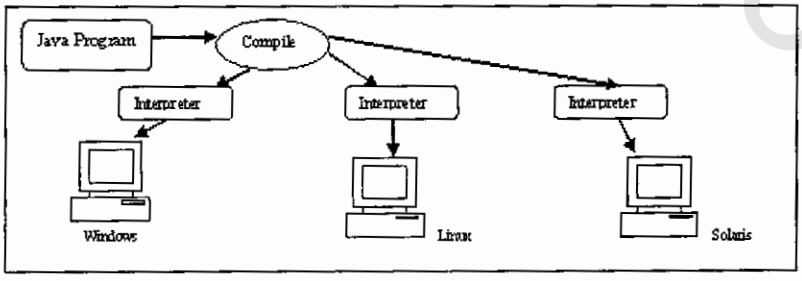
- قامت لغة Java بإعداد بيئة تشغيل للبرامج التى تبنى بهـا وتسـمى Java Virtual Machine (JVM) والتى تقوم بترجمة البرنامج للغـة وسـيطة تسـمي Byte Code ، وبالتالي فإن برامج لغة Java غير مرتبطة بنظام تشغيل Operating System معـين ، فقط يتم تشغيل نفس البرنامج على أي نظام تشـغيل Operating System أخـر مـع وجود بيئة التنفيذ JVM ، وهذا يؤدى إلى تنفيذ البرنامج بالضبط مثلما كان يعمل علـي نظام التشغيل Operating System الأول ، وهذه أحد أهم مزايا لغة Java.
- ويظهر دور JVM من الشكل (1-1).



(الشكل 1-1)

لا تعتمد على نظام التشغيل Platform Independent:

- كما أشرنا فـى الفقـرة السـابقة أن وجـود JVM لكـل نظـام تشـغيل Operating System أدى إلى إمكانيـة تنفيـذ نفـس البرنـامج علـى أكثـر مـن نظـام تشـغيل Operating System ، ويظهر ذلك من الشكل (1-2).



(الشكل 1-2)

لها مكتبة فصائل Classes Library كبيرة وغنية:

- نظراً لأن لغة Java لغة تعتمد على لبرمجة بواسطة الأهداف OOP ، فهى تحتوى على مكتبة فصائل Classes Library قوية توفر معظم الوظائف المطلوبة للعديد من التطبيقات مثل تطبيقات التعامل مع الملفات والتعامل مع قواعد البيانات Databases والتعامل مع الشبكات Networks وكذلك التعامل مع الرسومات المجسمة 3D Graphics والحركة Animation وكذلك التعامل مع الإنترنت Web Developing.

لغة مبنية على قواعد اللغة C/C++:

- عندما تم إنشاء لغة Java كان أساس بنائها لغة من أشهر وأقوى اللغات وهى C/C++ ، وبالتالى فهى لم تبدأ من حيث بدأ الأخرون بل من حيث انتهى الأخرون ، ثم تم إضافة العديد من المميزات فى لغة Java.

تعريفات وتقنيات Technologies هامة:

- هناك بعض التعريفات والتقنيات Technologies التى يجب الإلمام بها قبل دراسة لغة Java هى:

1. بيئة تشغيل لغة Java (JVM).
2. البريمج Java Applet.
3. تطبيق Java.
4. تطبيق الجهاز الخادم Servlet.
5. صفحات الخادم بلغة جافا JavaServer Pages (JSP).
6. مكتبة العلامات القياسية JSP Standard Tag Library (JSTL).
7. العلامات المخصصة Custom Tags.
8. حبيبات جافا للمشروعات Enterprise Java Beans (EJB).

بيئة تشغيل لغة Java (JVM):

- الحروف JVM اختصار للعبارة Java Virtual Machine أى ماكينة جافا التخيلية ، وهى فكرة قامت لغة Java بإنشائها لتجعل لغة Java تعمل على جميع أنظمة التشغيل Operating Systems ، وتقوم الفكرة على إنشاء طبقة وسيطة لكل

نظام تشغيل Operating System بحيث تكون هذه الطبقة الوسيطة قادرة علي فهم برامج Java لكي تقوم بتشغيلها علي نظام التشغيل Operating System ، وهذه الطبقة الوسيطة معروفة باسم ماكينة جافا التخيلية Java Virtual Machine (JVM) ، وبهذا كان من مزايا لغة Java أنها تعمل على العديد من نظم التشغيل Operating Systems الموجودة ، فلا يهم نظام التشغيل Operating System الذي تم عمل البرنامج عليه ، المهم أن البرنامج يتم كتابته ثم يتم تحميله إلى أي نظام تشغيل Operating System بشرط وجود JVM وبالتالى يعمل البرنامج.

البريمج Java Applet:

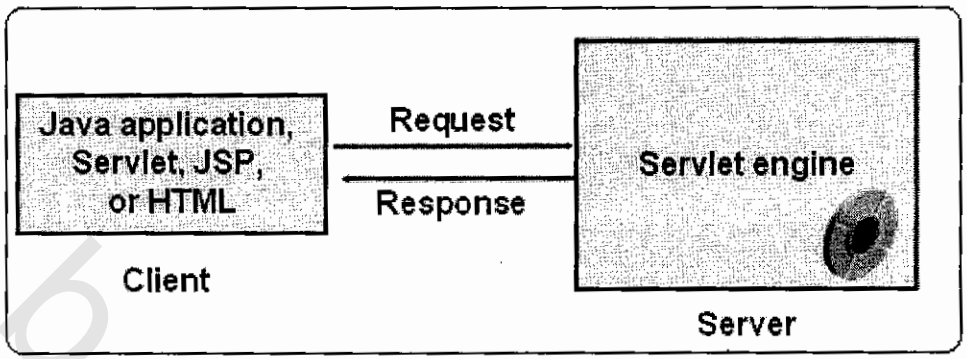
- البريمج Applet (بضم الباء وفتح الراء وسكون الياء وكسر الميم) -أي تصغير كلمة برنامج- هو نوع من أنواع التطبيقات الذى صمم خصيصاً للإنترنت Internet حيث يقوم المبرمج بإعداد بريمج Applet ثم يتم استدعاؤها من خلال ملف HTML فيتم عرض هذا التطبيق من خلال صفحة الإنترنت Internet ، وسوف نتعرض فى الفقرات التالية لطريقة إنشاء بريمج Applet وطريقة استدعائها من داخل ملف HTML.

تطبيق Java:

- هو تطبيق يتشابه مع التطبيقات التى يتم إنشاؤها بجميع لغات البرمجة الأخرى ، وبالرغم من أن المشهور عن لغة Java أنها تعد برامج للإنترنت Internet ، إلا أنها توفر الكثير من نقاط القوة فى إعداد تطبيق علي جهاز واحد Desktop Application أو تطبيق خاص بالشبكات Client/Server Application ، مما شجع شركة أوراكل Oracle على تبنى اللغة واستعمالها فى إعداد التطبيقات من خلال الأداة JDeveloper وقاعدة بيانات أوراكل Oracle.

تطبيق الجهاز الخادم Servlet:

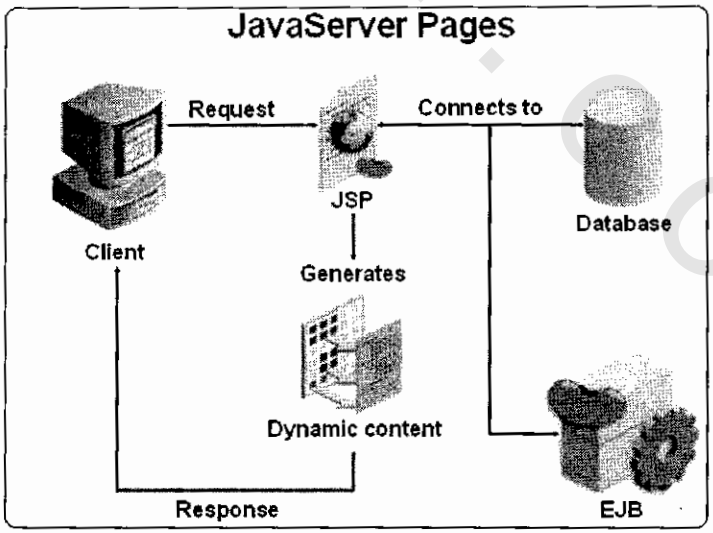
- هى صورة من صور فصائل Classes لغة Java ، وهى صممت خصيصاً لإعداد تطبيقات على شبكة الإنترنت Internet ، والشكل (1-3) يوضح كيف تعمل فصائل تطبيق الجهاز الخادم Servlet .



(الشكل 1-3)

صفحات الخادم بلغة جافا JavaServer Pages (JSP):

- الحروف JSP اختصار للعبارة JavaServer Pages وهى أحد صور لغة المسودة Script التى يطلق عليها اسم لغة المسودة للهاز الخادم Server Side Scripts ، وبالطبع تستخدم لإعداد صفحات الإنترنت Internet ذات المحتويات المتغيرة Dynamic Contents مثل عمليات قواعد البيانات Databases وغيرها ، ويتم كتابة أكوادها داخل صفحات HTML مثلها مثل أي لغة مسودة Script.
- والشكل (1-4) يوضح كيف تعمل أكواد JSP.



(الشكل 1-4)

مكتبة العلامات القياسية JSP Standard Tag Library (JSTL):

- هى مجموعة من العلامات Tags الجاهزة قامت شركة Sun بإعدادها فى مكتبات لغة Java لتسهيل كتابة الأوامر داخل صفحة JSP.

العلامات المخصصة Custom Tags :

- هى طريقة من الطرق المتقدمة لإنشاء علامات Tags تخص المبرمج لتحقيق أغراض معينة يستعملها وقت ما يشاء مما يسهل إعادة بناء التطبيقات ويحقق نظرية إعادة الاستخدام Reusability.

حبيبات جافا للمشروعات Enterprise Java Beans (EJB):

- هي طريقة لإعداد وحدات Components تسهل التعامل مع قواعد البيانات Databases وذلك من خلال وحدات Components تتصل بقاعدة البيانات Databases وتوفر دوال التعامل معها.

طرق كتابة برامج Java:

توجد أكثر من طريقة لكتابة برامج Java وترجمتها منها:

1. استعمال المكتبة JDK مباشرة مع استعمال أى محرر نصوص Text Editor.
2. استعمال برامج وسيطة مثل JCreator أو Kawa.
3. استعمال البرامج المعدة للغة Java مثل JBuilder, JDeveloper, NetBeans.

وفيما يلى توضيح هذه الطرق.

استعمال المكتبة JDK مباشرة:

وهى الطريقة التقليدية وذلك بإعداد مكتبة Java على جهازك Java Development Kit (JDK) والتى توفرها شركة Sun مجاناً على موقعها -كما سنشرح لاحقاً- مع كتابة البرنامج باستعمال أي محرر نصوص Text Editor.

سوف تحتاج ما يلى:

1. محرر نصوص Text Editor:

وليكن برنامج المفكرة Notepad الموجود مع برنامج النوافذ Windows.

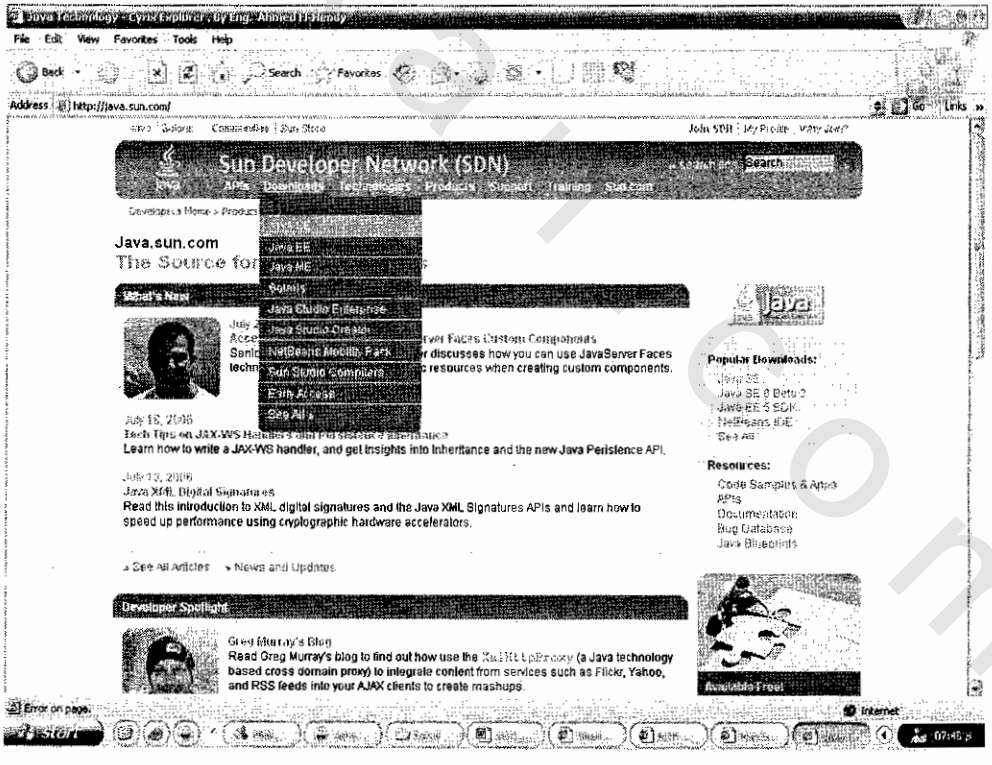
2. مكتبة JDK:

ويمكنك الحصول على مكتبة JDK بأكثر من طريقة ، إما بنسخها على اسطوانة وإعدادها على الجهاز أو بإنزالها (Download) من موقع الشركة بالإنترنت Internet ، ولتحقيق ذلك اتبع ما يلى:

- قم بتشغيل الجهاز واتصل بشبكة الإنترنت Internet كما هو معتاد.
- ادخل موقع شركة Sun بكتابة العنوان التالى:

http://java.sun.com

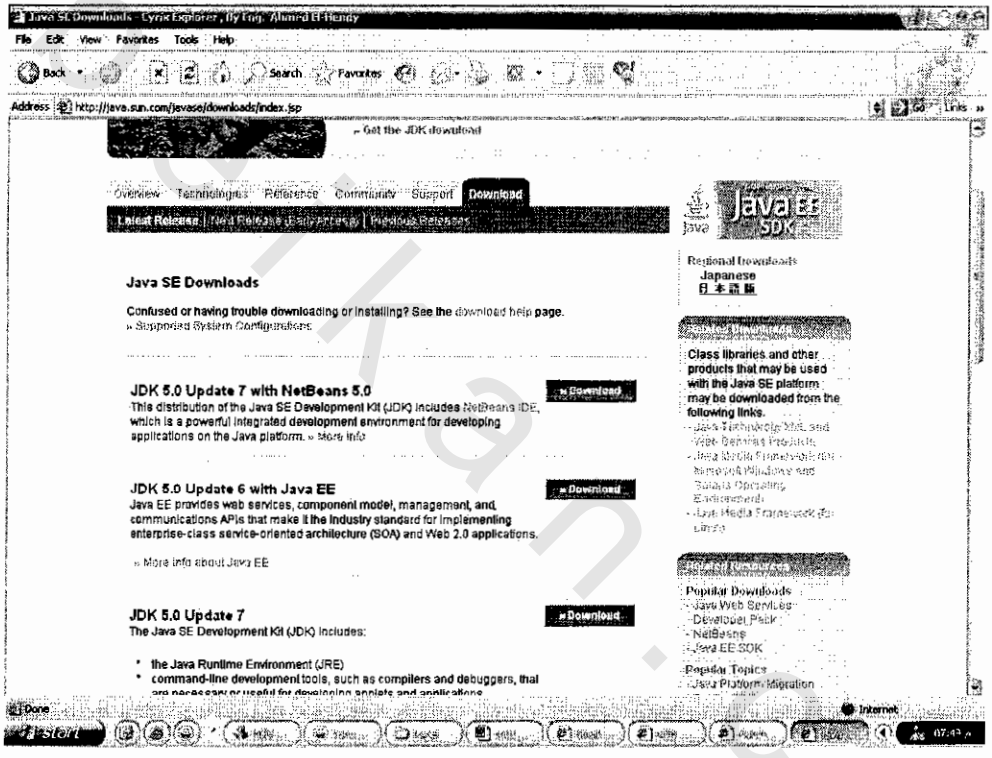
لتحصل على موقع الشركة كما فى الشكل (1-5).



(الشكل 1-5)

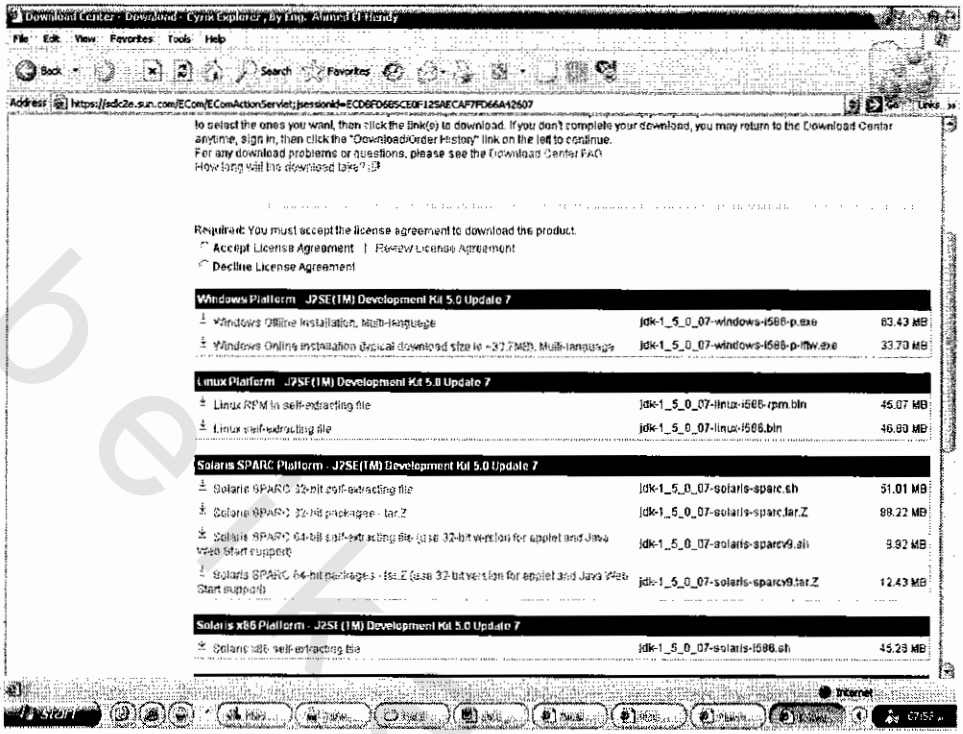
بسهولة اضغط الاختيار Download واختر منها Java SE بمعنى Java Standard Edition.

تحصل على صفحة إنزال البرامج كما فى الشكل (1-6).



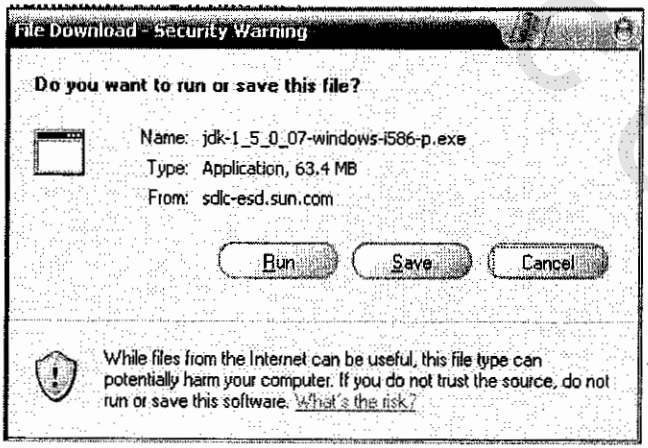
(الشكل 1-6)

اضغط الزر Download الموجود بجانب عبارة JDK 5.0 Update 7 لتنتقل إلى صفحة تنزيل مجموعة البرامج الخاصة بتشغيل برامج Java كما فىالشكل (1-7).



(الشكل 1-7)

فى هذا الشكل اختر أولاً Accept License Agreement ثم اضغط الاختيار Windows Offline Installation, Multi-language ليظهر مربع حوار حفظ الملف كما فى الشكل (1-8).



(الشكل 1-8)

قم بحفظ الملف وتابع حتى يتم إنزال JDK على القرص الصلب Hard Disk ، قم بإعدادها كما تعد أي برنامج ، وسوف تجد الخطوات الكاملة للإعداد فى نهاية الكتاب فى جزء الملاحق.

تم وضع نسخة من الأمثلة على موقع المؤلفين
(انظر مقدمة الكتاب).

استعمال برامج وسيطة:

الطريقة الثانية هى استعمال بعض البرامج البسيطة والمعدة لتسهيل كتابة برنامج لغة Java ولكنها ليست بيئة متكاملة Integrated Environment ، حيث توفر كتابة البرنامج وتنفيذه من خلال بيئة رسومية Graphical User Interface بشرط وجود مكتبة ومترجم Java مسبقاً على الجهاز (JDK) ، وتوجد العديد من هذه البرامج مثل برنامج JCreator وكذلك Kawa وسوف نشرح كيفية استعمال هذه البرامج فيما بعد.

استعمال البرامج المعدة للغة Java:

وهى أفضل وأقوى الطرق لكتابة البرامج بلغة Java حيث نقوم باستعمال أحد البرامج التى يطلق عليها اسم البرامج المرئية Visual ، حيث توفر جميع متطلبات إعداد تطبيق Java مثل المكتبة والمترجم Compiler ونستعملها كأنك تستعمل أحد اللغات المشهورة السهلة مثل لغة Visual Basic. ومن هذه البرامج برنامج Forte وهو من إنتاج شركة Sun وكذلك البرنامج Visual Cafe والبرنامج JBuilder والبرنامج JDeveloper ، وسوف نشرح كيفية استعمال بعض من هذه البرامج لاحقاً.

أوامر JDK:

فيما يلي أهم الأوامر التي تستخدم لترجمة Compile وتشغيل برامج لغة Java.

- الأمر javac: هو الأمر المستعمل فى ترجمة Compile الملف المصدرSource Code.
- الأمر java: هو الأمر المستعمل في تنفيذ برامج Java.
- الأمر appletviewer: يستخدم لعرض تطبيقات البريمج Applet.

مثال (1): خطوات إنشاء برنامج Java :

فيما يلي خطوات إعداد برنامجJava :

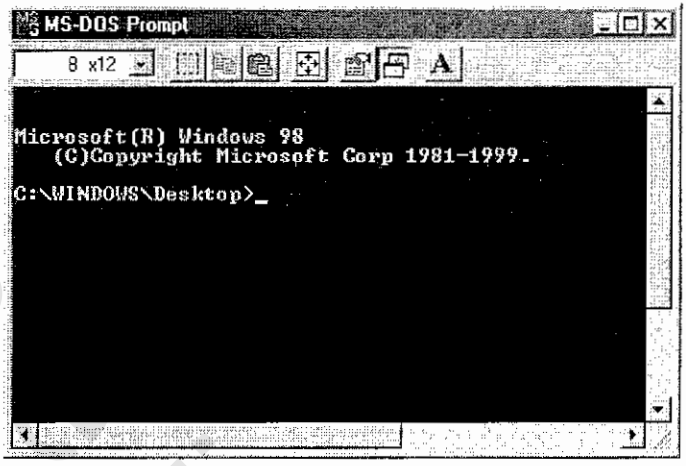
1. كتابة البرنامج باستخدام محرر نصوص Text Editor.
2. ترجمة Compile البرنامج.
3. تنفيذ Execute البرنامج.

- سوف نبدأ بفتح برنامج المفكرة Notepad ثم نكتب البرنامج كما هو واضح في الشكل (1-9).
- هذا هو الشكل الأساسي لبرنامج Java وسوف يتم توضيح جميع أجزاء البرنامج فى الفصول التالية.

```
1  package mypackage3;
2  public class Ex1
3  {
4    public Ex1()
5    {
6    }
7  public static void main(String[] args)
8  {
9  new Ex1();
10   System.out.println("Welcome With First Java Program");
11 }
12 }
13
```

(الشكل 1-9)

- احفظ هذا البرنامج بالاسم Ex1.java (لاحظ ضرورة تحديد الامتداد Extension ليكون java) ثم قم بالضغط علي زر البداية Start ثم اختيار Run ثم كتابة الأمر command (في نظام النوافذ Windows 98 و Windows ME) أو الأمر cmd (في نظام النوافذ Windows 2000 و Windows XP و Windows 2003) لتحصل على شاشة كتابة أوامر Dos كما فى الشكل (1-10).



(الشكل 1-10)

- قم بتحويل المسار إلي نفس المسار الذي قمت بحفظ الملف Ex1.java فيه (من خلال أمر cd) ، ثم اكتب أمر الترجمة Compile كما فى السطر التالى:

```
javac Ex1.java
```

- يتم ترجمة Compile البرنامج ، وفي حالة وجود أخطاء ستحصل على رسالة تفيد ذلك وتبين نوع الخطأ ، أما في حالة عدم وجود أخطاء فلن تجد أي رسائل وستحصل على ملف بالاسم

```
Ex1.class
```

- يمكنك التأكد من إنشاء هذا الملف في نفس مسار الملف Ex1.java.
- نفذ البرنامج باستعمال الأمر java كما فى السطر التالى:

```
java Ex1
```

- تحصل على الرسالة التالية وهى نتيجة تنفيذ البرنامج:

```
Welcome With First Java Program
```

- تم ذكر هذه الخطوات فقط لتأخذ فكرة عن البرمجة بلغة Java ، وسوف نتناول كل هذه الخطوات بالتفصيل في الفصول القادمة.

مثال (2): خطوات إعداد بريمج Applet :

فيما يلى نتناول خطوات إعداد بريمج Applet وهي:

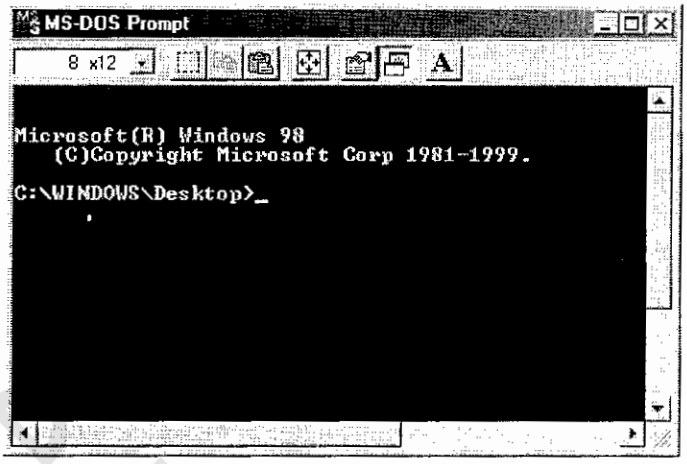
1. كتابة البريمج Applet باستخدام محرر نصوص Text Editor.
2. ترجمة Compile البرنامج.
3. تنفيذ Execute البرنامج.

افتح برنامج المفكرة Notepad واكتب به السطور التالية (لا تكتب أرقام السطور):

```
1:  import java.awt.Graphics;
2:  import java.awt.Font;
3:  import java.awt.Color;
4:
5:  public class HelloAgainApplet extends java.applet.Applet
6:  {
7:    Font f = new Font("TimesRoman", Font.BOLD, 36);
8:
9:    public void paint(Graphics g)
10:   {
11:     g.setFont(f);
12:     g.setColor(Color.red);
13:     g.drawString("Hello again!", 5, 40);
14:   }
15: }
```

احفظ هذا الملف باسم HelloAgainApplet.java (لاحظ ضرورة تحديد الامتداد Extension ليكون java).

قم بتشغيل أوامر DOS بأى طريقة ولتكن Start → Run → Command لتحصل على شاشة كتابة أوامر DOS كما فى الشكل (1-11).



(الشكل 1-11)

- قم بتحويل المسار إلي نفس المسار الذي قمت بحفظ الملف Ex1.java فيه (من خلال أمر cd) ، ثم اكتب أمر الترجمة Compile كما فى السطر التالى:

```
javac HelloAgainApplet.java
```

- اضغط الزر Enter وستلاحظ ترجمة Compile البرنامج ، فإذا لم يكن بالبرنامج أخطاء تحصل على الملف HelloAgainApplet.class وهو نتيجة الترجمة كما أشرنا من قبل.
- هناك أكثر من طريقة لتنفيذ البريمج Applet حيث يمكننا استعمال الأمر appletviewer وذلك عن طريق كتابة الأمر كما يلي:

```
appletviewer HelloAgainApplet
```

- لاحظ تنفيذ البريمج Applet وظهور النتيجة عبارة عن الرسالة Hello Again!.

ملحوظة:

تستخدم هذه الطريقة للاختبار فقط حيث أنه من غير المنطقي أن يقوم المستخدم بكتابة هذه الأوامر لتشغيل البريمج Applet ، أما الأفضل فهو استخدام الطريقة الثانية لتشغيل البريمج Applet كما يلي.

مثال (3): عرض البريمج Applet داخل صفحة الوب Web Page:

وهذه هى الطريقة الشائعة والمشهورة لأن البريمج Applet يتم عرضه من خلال صفحة

وب Web Page ، وتتم عملية إدخال البريمج Applet داخل صفحة الوب Web Page باستعمال علامات Tags لغة HTML وهي العلامة <Applet> ، ويظهر ذلك من نص ملف HTML التالي:

```
1: <HTML>
2: <HEAD>
3: <TITLE>This page has an applet on it</TITLE>
4: </HEAD>
5: <BODY>
6: <p>My second Java applet says:
7: <br><APPLET CODE="HelloAgainApplet"
   WIDTH="200" HEIGHT="50">
8: Hello Again!
9: </APPLET>
10:</BODY>
11:</HTML>
```

- تم استعمال العلامة <Applet> ثم الصفة CODE لتحديد اسم البريمج Applet المطلوب تحميله ثم تم تحديد عرض وارتفاع نافذة البريمج Applet ثم يتم إغلاق علامة <Applet> بالعلامة </Applet> التى تعنى انتهاء التعامل معها.

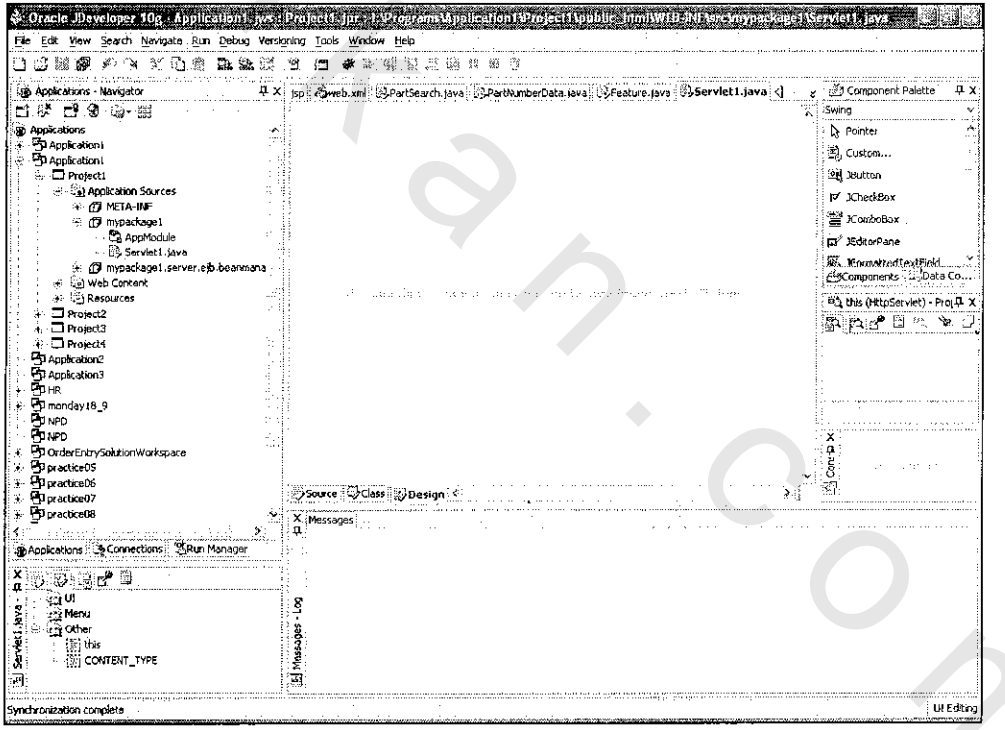
دورة حياة البريمج Applet:

- الصورة السابقة للبريمج Applet الذى أنشأناه هى أبسط صورة للبريمج Applet ولكن هناك الصور الأكثر شمولية التى تظهر فيها دوال Methods البريمج Applet وهي:
  - الدالة ()init التى تنفذ مرة واحدة فقط عند تحميل البريمج Applet.
  - الدالة ()paint التى تنفذ عند الدخول فى البرنامج ورسم محتوياته.
  - الدالة ()stop وتنفذ عند الانتقال من الصفحة التى تحتوى على البريمج Applet إلى صفحة أخرى.
  - الدالة ()repaint وتنفذ عند الرجوع إلى الصفحة التى تحتوى على البرنامج من صفحة أخرى.

- الدالة ()destroy وتنفذ عند الانتهاء من استعمال البريمج Applet.

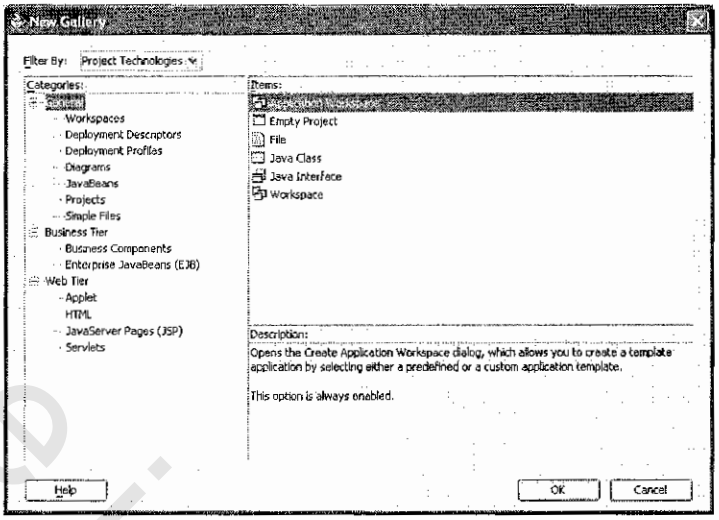
البرنامج JDeveloper:

- كما أشرنا توجد برامج جاهزة تستعمل لإعداد برامج وتطبيقات لغة Java بسهولة ، ومن هذه البرامج Oracle JDeveloper وهو منتج قوى يمكنك إنزال نسخة مجانية منه من موقع الشركة www.oracle.com وأخر إصداراته حتى كتابة هذا الكتاب الإصدار g10.
- وبعد إنزال وتشغيل البرنامج تحصل على الشاشة الرئيسية للبرنامج كما فى الشكل (1-12).



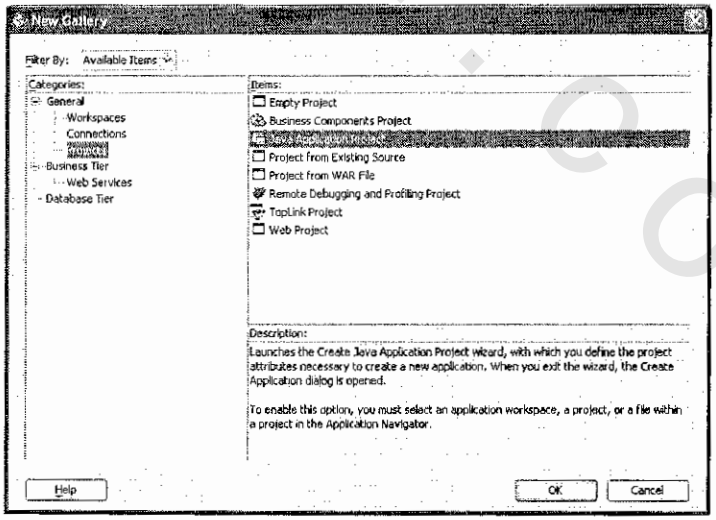
(الشكل 1-12)

- سنقوم باستعمال البرنامج لكتابة برامج Java كما يلى:
- من القائمة الرئيسية اختر File ثم New ثم Workspace كما فى الشكل (1-13).



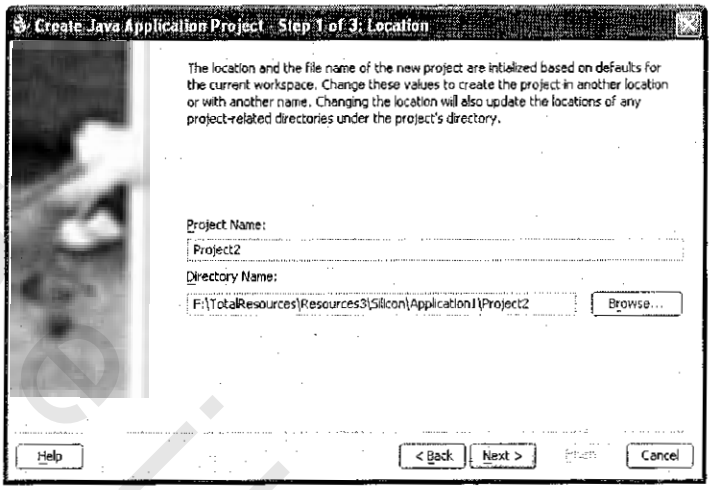
(الشكل 1-13)

- اضغط الزر OK لتحصل على مربع حوار New Workspace لتحديد الاسم. اترك كل شئ كما هو واضغط الزر Ok تحصل على مربع حوار New Project لتحديد اسم ومسار التطبيق. اضغط الزر Ok لتعود الى شاشة التصميم.
- اختر File ثم New لتظهر الشاشة كما فى الشكل (1-14) حيث نختار Projects من الناحية اليسري و Java Application Project من الناحية اليمني.



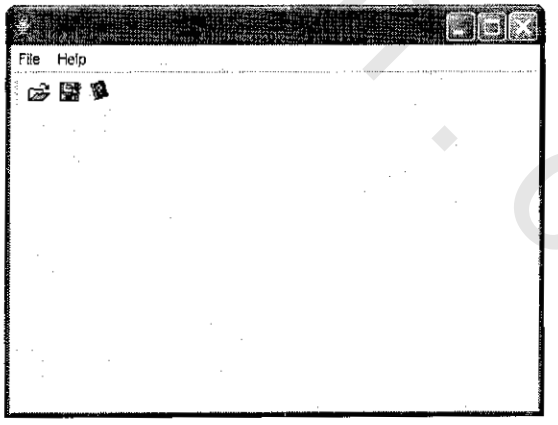
(الشكل 1-14)

- اضغط الزر Ok لتحصل على الشاشة كما فى الشكل (1-15).



(الشكل 1-15)

- اضغط الزر Next حتى تنتهى خطوات الإعداد وتعود إلى بيئة التصميم.
- نفذ البرنامج بالضغط على الزر F11 لتحصل على الشكل الأساسي للبرنامج كما فى الشكل (1-16).

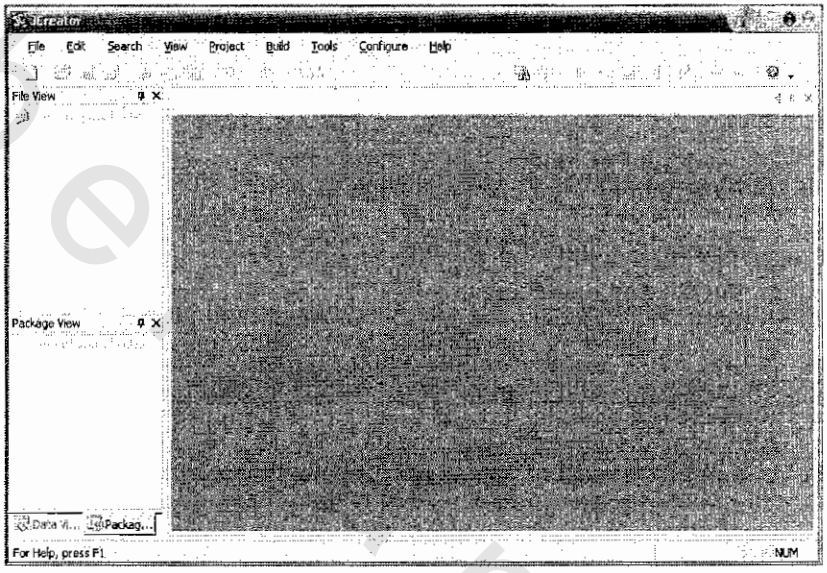


(الشكل 1-16)

البرنامج JCreator:

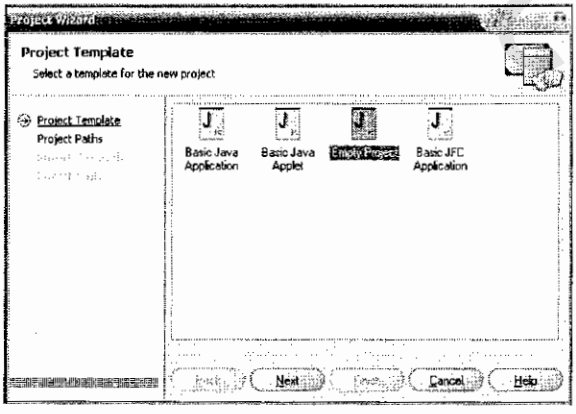
- من البرامج المشهورة جداً والمستعملة بكثرة فى التدريب على لغة Java البرنامج JCreator ، ويتم شرح كيفية تنزيل وإعداد البرنامج JCreator فى الملحق جـ.

- ولتوضيح كيفية استعماله تابع الخطوات التالية:
- قم بتشغيل البرنامج JCreator لتحصل على الشاشة الرئيسية للبرنامج كما فى الشكل (1-17).



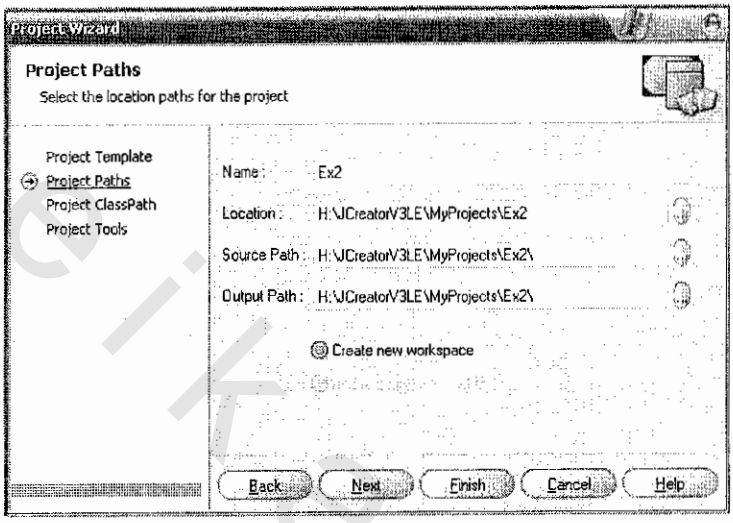
(الشكل 1-17)

- اختر New من القائمة File لتحصل على الشاشة كما فى الشكل (1-18).



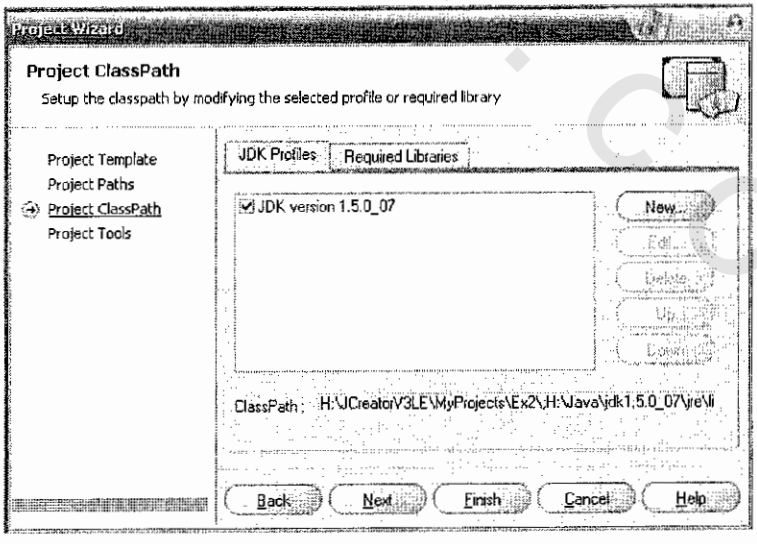
(الشكل 1-18)

- اختر Empty Project ثم اضغط الزر Next لتحصل على شاشة تحديد مواصفات التطبيق.
- اكتب اسم التطبيق Ex2 كما فى الشكل (1-19).



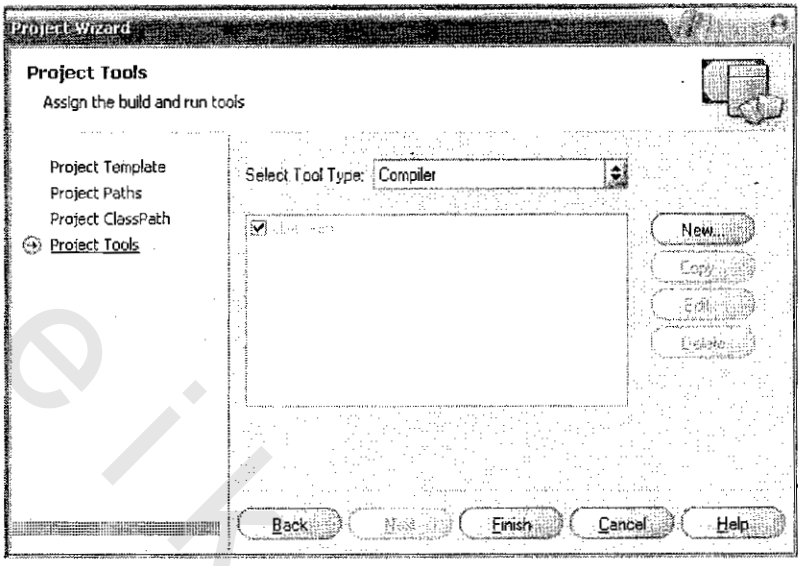
(الشكل 1-19)

- اضغط الزر Next لتحصل على الشاشة كما فى الشكل (1-20).



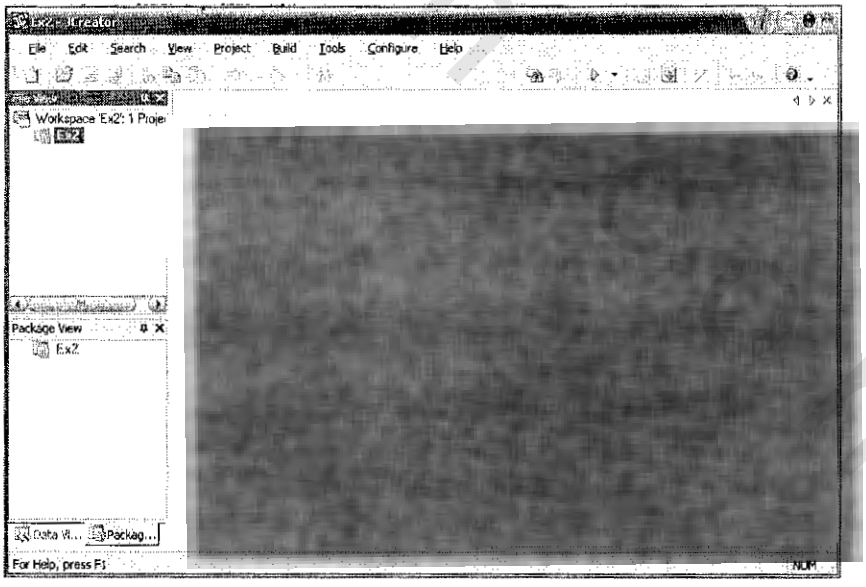
(الشكل 1-20)

اضغط الزر Next تحصل على أخر مربع حوار كما فى الشكل (1-21).



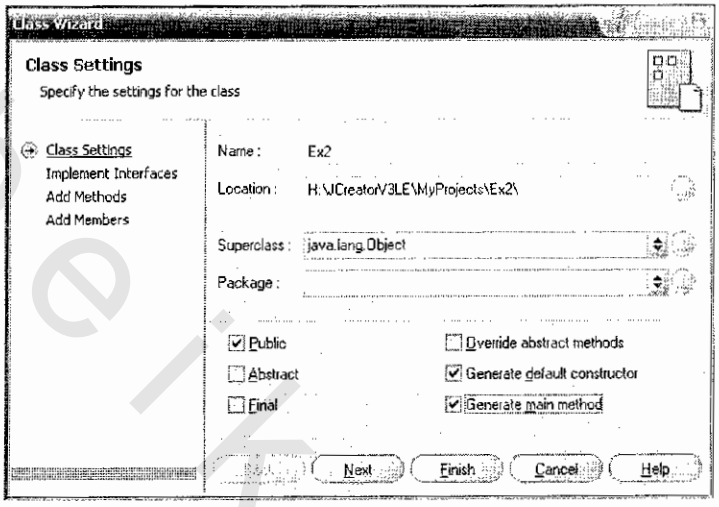
(الشكل 1-21)

اضغط الزر Finish تحصل على الشاشة الرئيسية كما فى الشكل (1-22).



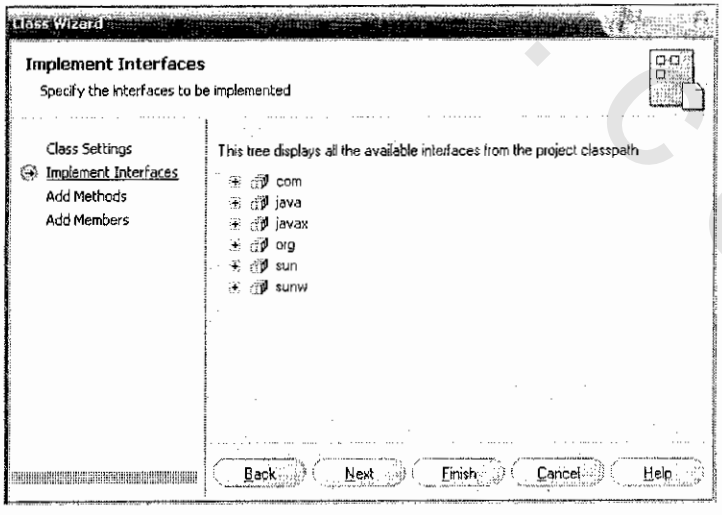
(الشكل 1-22)

- اضغط اسم التطبيق Ex2 بالزر الأيمن للفارة Mouse ثم اختر Add ومنها اختر class تحصل على الشاشة كما فى الشكل (1-23).



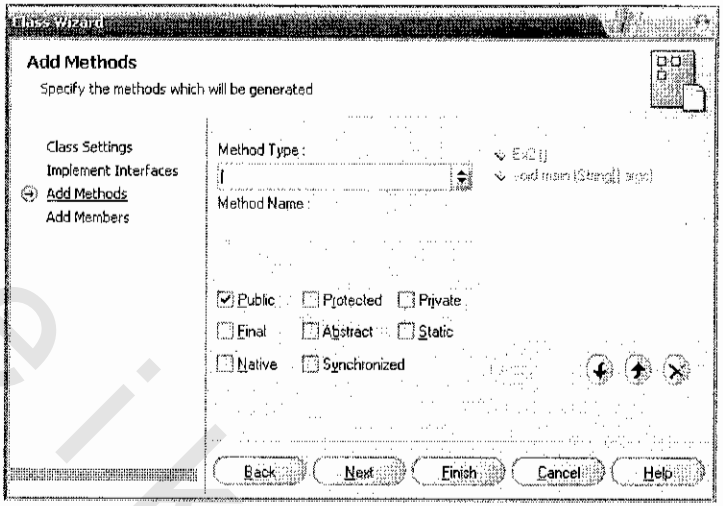
(الشكل 1-23)

- فى هذا الشكل تلاحظ قيام المعالج Wizard بإضافة المواصفات الرئيسية للبرنامج.
- اضغط الزر Next لتحصل على الشاشة كما فى الشكل (1-24).



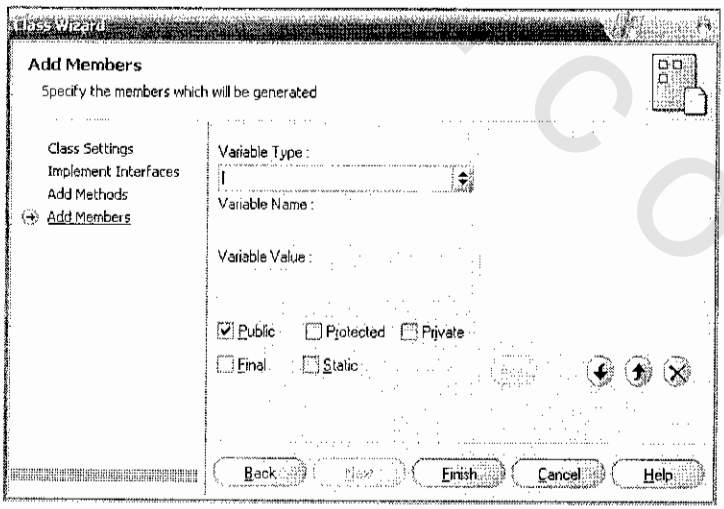
(الشكل 1-24)

- اضغط الزر Next لتحصل على الشاشة كما فى الشكل (1-25).



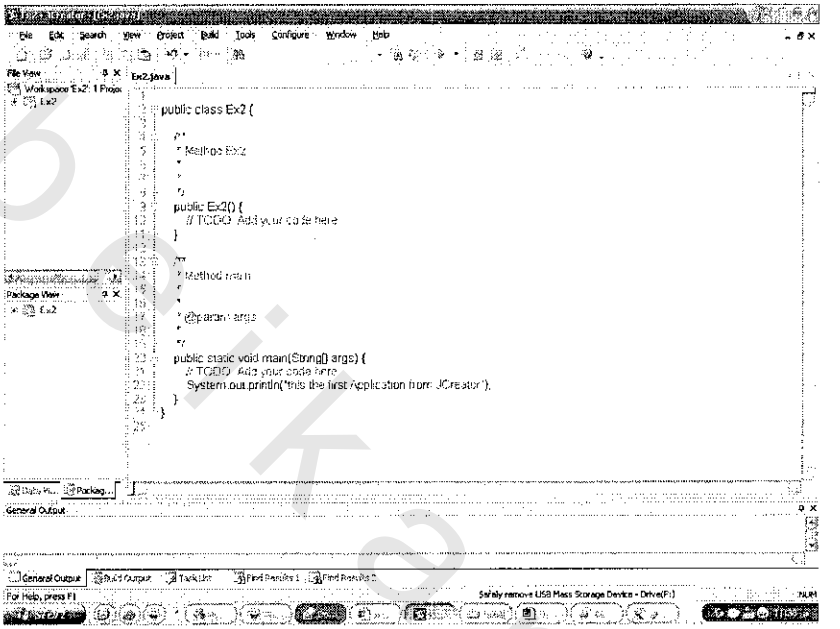
(الشكل 1-25)

- فى هذا الشكل تلاحظ إمكانية إضافة عناصر جديدة للبرنامج وسوف نتناولها فى الفصول التالية.
- اضغط الزر Next لتحصل على الشاشة كما فى الشكل (1-26).



(الشكل 1-26)

- اضغط الزر Finish لتحصل على شاشة الأوامر.
- اكتب سطور البرنامج التالية كما فى الشكل (1-27).



(الشكل 1-27)

شرح السطور:

- تم إنشاء التركيب الأساسى لبرنامج Java كما سبق شرحه ، كما تم إضافة سطر طباعة نص في السطر رقم 22.
- اضغط الزر F7 لترجمة البرنامج ومراجعة أخطاء البرنامج وستلاحظ ظهور رسائل تفيد ذلك.
- اضغط الزر F5 لتنفيذ البرنامج لتحصل على نتيجة التنفيذ كما فى الشكل (1-28).

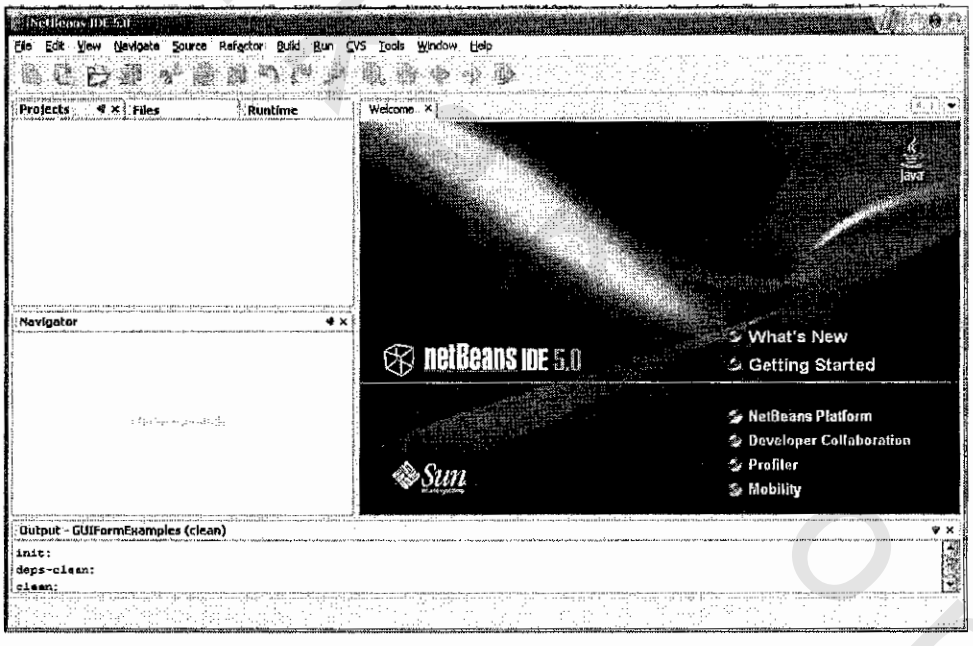
(الشكل 1-28)

البرنامج NetBeans:

من الأدوات الممتازة لكتابة وتطوير تطبيقات Java البرنامج NetBeans ، ويتميز بالسهولة فى التطوير ، وقد تم عرض شرح كامل لخطوات إعداد هذا البرنامج فى ملاحق الكتاب فى الملحق هـ.

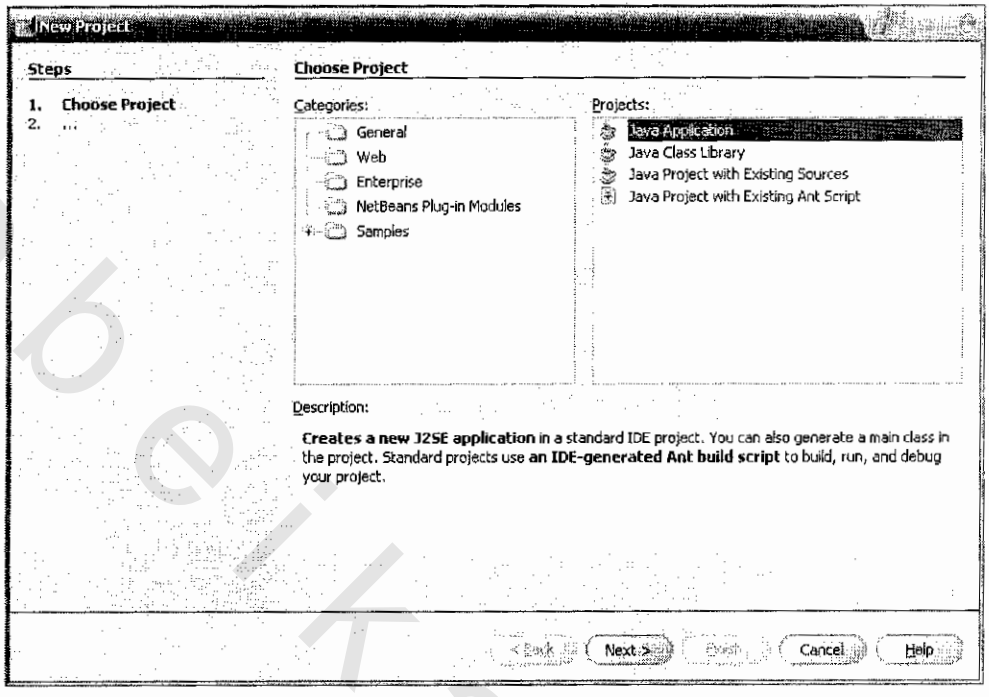
خطوات استعمال البرنامج NetBeans IDE:

- قم بإعداد البرنامج NetBeans IDE باستعمال الخطوات الموجودة فى الملاحق ثم تابع معنا خطوات استعمال البرنامج فى تطوير تطبيقات Java كما يلى:
- قم بتشغيل البرنامج NetBeans IDE لتحصل على الشاشة الرئيسية كما فى الشكل (1-29).



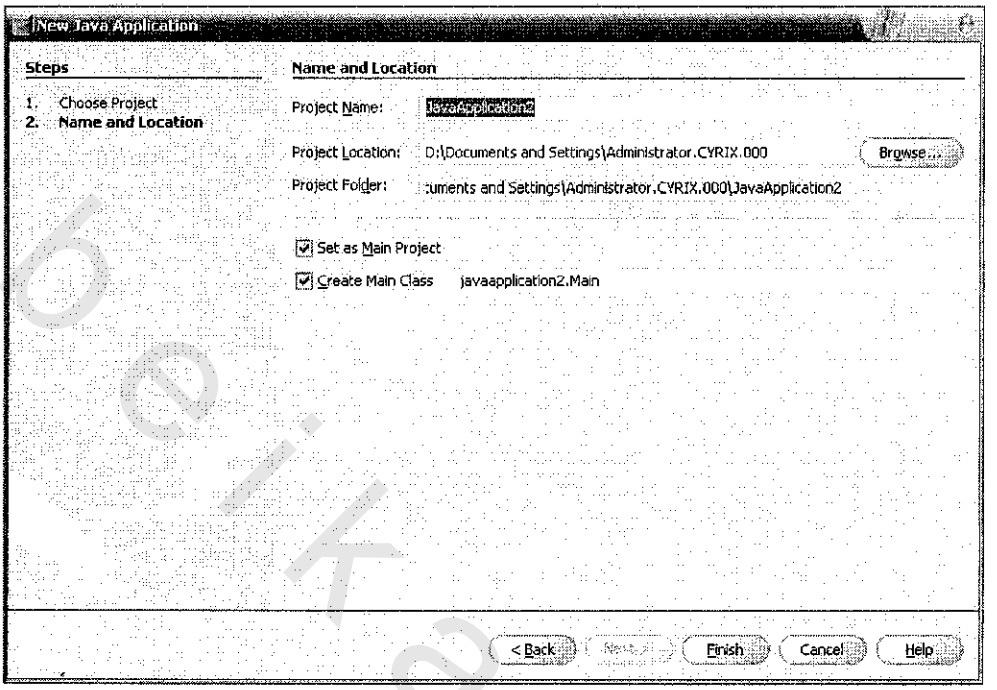
(الشكل 1-29)

- من القائمة الرئيسية للبرنامج اختر File ثم New Project تحصل على مربع حوار New Project الذى يعرض لك أنواع التطبيقات التى يمكن إنشاؤها. اختر General ثم اختر Java Application كما فى الشكل (1-30).



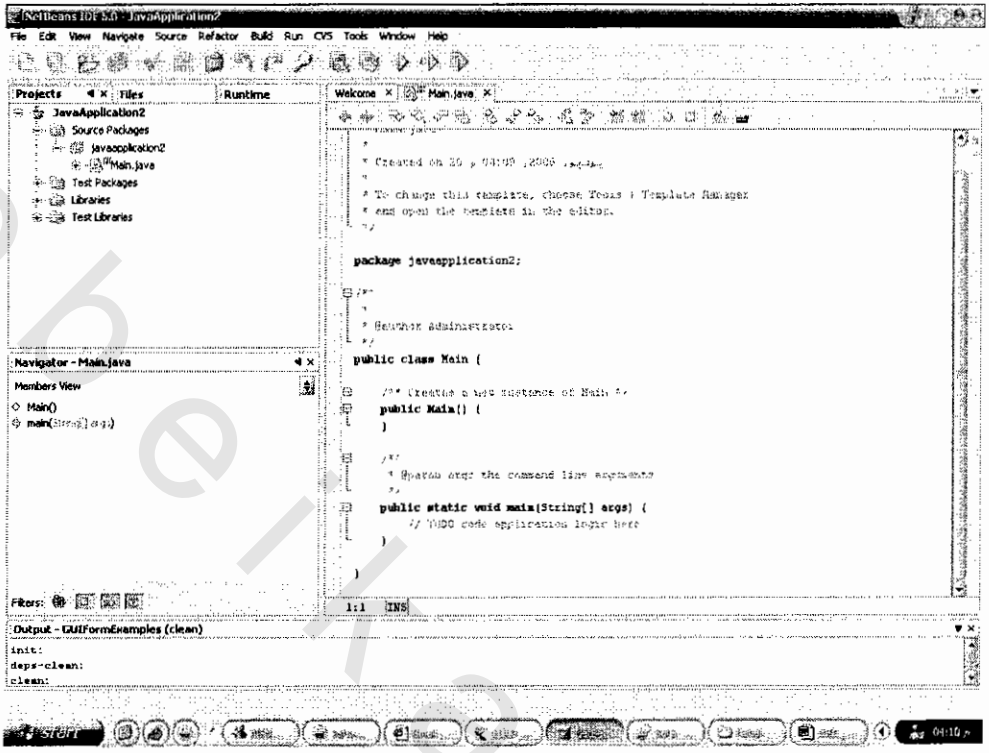
(الشكل 1-30)

بعد تحديد General من الناحية اليسري اختر Java Application من الناحية اليمني ثم اضغط الزر Next تحصل على مربع حوار تحديد اسم هذا التطبيق ومسار حفظ الملفات كما فى الشكل (1-31).



(الشكل 1-31)

- اترك الاختيارات الافتراضية كما هى ثم اضغط الزر Finish ليتم إنشاء التطبيق وتحصل على صفحة أكواد البرنامج تلقائياً كما فى الشكل (1-32).



(الشكل 1-32)

- في هذا الشكل تلاحظ قيام برنامج NetBeans بإنشاء جميع أجزاء برنامج Java حيث أنشأ فصيلة class باسم Main وبداخلها الدالة الرئيسية ()main وبالتالى وفر عليك إنشاء ذلك.
- قم بكتابة السطر التالي داخل سطور الدالة ()main وذلك كما فى الشكل (1-33).

```
System.out.println("hello from first NetBeans IDE Application");
```

```
/*
 * Main.java
 *
 * Created on 20 ,يوليو ,2006 04:09 م
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication2;

/**
 *
 * @author Administrator
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }
       /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("hello from first NetBeans IDE Application");
    }

}
```

(الشكل 1-33)

نفذ البرنامج بالضغط على الزر F6 لتحصل على الشكل (1-34) في أسفل شاشة البرنامج.

```
Output - JavaApplication2 (run)
init:
deps-jar:
compile:
run:
hello from first NetBeans IDE Application
BUILD SUCCESSFUL (total time: 5 seconds)
Finished building JavaApplication2 (run).
```

Output

(الشكل 1-34)

- مما سبق تلاحظ مدى سهولة استعمال البرنامج NetBeans IDE فى تطوير تطبيقات Java وبالطبع ما زال يحتوى على الكثير وهذا ما سنتناوله فى أماكن لاحقة من الكتاب.

ملخص الفصل:

- تعرضنا في هذا الفصل لشرح أساسيات لغة Java وتعرفنا علي أهم تقنيات لغة Java بالإضافة إلي شرح العديد من الطرق المستخدمة لكتابة وتنفيذ برامج Java.
- في الفصل القادم سوف نتعرف - بإذن الله - علي أنواع البيانات Data Types في لغة Java ، فتابع معنا الفصل القادم.

obeikan.com