

In The Name Of Allah

Book:

Lets Learn JaVa

By:

Ahmed Salah
Faculty Of Engineering
Electrical Eng. Department
Power & Machines Section

Email:

d1e.h4rd@yahoo.com

المحتويات:

- ١ - نبذة عن لغة الجافا.
- ٢ - البرامج المستخدمة وكيفية تنصيبها.
- ٣ - مكونات لغة الجافا.
- ٤ - اهم الاوامر المستخدمة .
- ٥ - تطبيقات بسيطة.

التعريف بلغة الجافا:

هي لغة حديثة الى حد ما حيث تم انتاجها عام ١٩٩٥م لتناسب التطبيقات الحديثة، وهي تناسب تطبيقات الإنترنت حيث أصبحت قلب برمجة الإنترنت بما توفره من إمكانيات .

مواصفات لغة الجافا:

- ١ - لغة برمجية تدعم مبادئ الكائنية .
- ٢ - لها بيئة تشغيل خاصة بها.
- ٣ - تقوم على لغة C و C++.
- ٤ - تعمل على معظم نظم التشغيل.

اهم مميزات لغة الجافا:

- ١ - إضافة الحركة والصوت إلى صفحات الويب.
- ٢ - كتابة الألعاب والبرامج المساعدة.
- ٣ - إنشاء برامج ذات واجهة مستخدم رسومية.

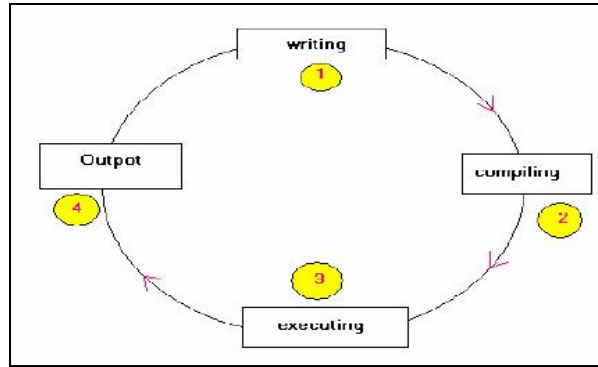
- ٤- تصميم برمجيات تستفيد من كل مميزات الانترنت.
- ٥- كتابة برامج تعليمية للإنترنت عبر برمجيات المحاكاة الحاسوبية للتجارب العلمية وبرمجيات الفصول الافتراضية للتعليم الإلكتروني والتعليم عن بعد.

ملاحظات خاصة بهذه اللغة:

- ١- اللغة حساسه للاحرف الصغيره والكبيره.
- ٢- يمكن استخدام اسطر فارغه حيث انها لا تؤثر في البرمجه.
- ٣- يجب حفظ الملف الخاص بالبرنامج بامتداد **java**.

المراحل التي يمر بها بناء اي برنامج بالجافا:

- ١- **Writing** يقصد كتابه البرنامج.
- ٢- **Compiling** يقصد بها معاينه البرنامج وتصحيح ما به من اخطاء.
- ٣- **Executing** تنفيذ البرنامج.
- ٤- **Output** اظهار المخرجات.



البرنامج المستخدم وملحقاته:

لابد من تنصيب ما يلي من برامج، وهما كالتالي:

اولا:

NetBeans IDE 6.9.1

ثانيا:

JDK

مكونات لغة الجافا:

- ١- الرموز الخاصه.
- ٢- الاختصارات المسانده.
- ٣- الكلمات المحجوزه.
- ٤- الحروف الابجديه.
- ٥- الارقام العدديه.

اولا: الحروف الابجديه:

وهي الحروف الكبيره من **A** حتى **Z**، وايضا الحروف الصغيره من **a** حتى **z**.

ثانيا: الارقام العدديه:

وهي الارقام من **0** حتى **9**.

ثالثا: الرموز الخاصه:

وهي العلامات او الرموز التي ليست بارقام او حروف كعلامه الجمع **+**، وعلامه الطرح **-**، وعلامه الضرب *****، وعلامه القسمة **/**، وعلامه المساواه **=**..... الخ.

رابعا: الاختصارات المسانده:

ونذكر منها على سبيل المثال لا الحصر:

\n سطر جديد.
\t التقدم سيع مسافات.
\b العوده مسافه للخلف.

خامسا: الكلمات المحجوزة:

ويشترط ان لا نستخدمها الا لوظيفتها المخصصة، ولا يجوز ان تكون اسماء لمتغيرات، ونذكر منها على سبيل المثال لا الحصر:

Return ، void ، for ، short ، if الخ

انواع المتغيرات:

اولا: المتغير الصحيح integer :

متغيرات عدديه صحيحة وقد تكون موجبه او سالبه ولا يحتوى على نقطه عشريه او اسيه ومثال له ، 70 ، 4 ، -6 ، الخ.

ثانيا: المتغير العشري float :

متغيرات عدديه عشرات ذات الفاصله العائمه ومثال له 5.0 ، 4.0 ، -0.1 ، -0.8 ، الخ.

ثالثا: المتغير الحرفي Char :

متغيرات عبارة عن حرف واحد فقط وبين اشارتى اقتباس مفردة ' ' ومثال له a ، b ، c ، الخ.

رابعا: متغير السلسله string :

يحتوى هذا النوع من البيانات على اى عدد من الرموز او الفراغات وتكون محصوره بين علامتى تنصيص " " .

خامسا: متغير الصواب والخطأ Boolean :

ويأخذ قيمتين اما الصواب true او الخطأ false .

سادسا: المتغير المزدوج Double :

هي نفس المتغيرات العددية العشرية ولكن يمكن تمثيلها الى خمسة عشر خانه، اى ان قيم هذا النوع تكون اكبر بكثير من قيم النوع float مثلا 13.40 ، -77.987 ، 13.15 ، 55.98 ، الخ

الصيغه العامه لتعريف متغير ما :-

Datatype variable_name = value

حيث:-

Datatype نوع المتغير سواء كان int او float او char او double ..

Variable name اسم المتغير وذلك يخضع لنوع المتغير السابق ادخاله ..

Value القيمة المراد تخزينها بالمتغير..

امثله:-

Boolean x = true

long z = 1234321

float y = 1.23f

char z = 'A'

لاحظ عند تعريف متغير من النوع long قمنا بكتابه الحرف l بعد قيمته، وبالمثل عند تعريف متغير من النوع float قمنا بكتابه الحرف f بعد قيمته.

التعليقات:

اذا اردت كتابه تعليق على اى سطر من سطر البرنامج هناك طريقتين وهما كالتالى:

الاولى :

```
// Your Comment
```

والثانية:

```
/*
```

```
Your comment
```

```
*/
```

ولن يلتفت اليه البرنامج او يضعه فى حساباته.

الحلقات التكراريه والجمل الشرطيه:

الصيغه الاولى:

(1)

```
IF ( الشرط )
{
    ≡ ; code
}
```

الصيغة الثانية:

(2)

```
IF ( الشرط )
{
    ≡ ; code
}
Else
{
    ≡ ; code
}
```

الصيغة الثالثة:

(3)

```
IF ( الشرط )
{
    ≡ ; code
}
Else IF ( الشرط )
{
    ≡ ; code
}
Else
{
    ≡ ; code
}
```

الصيغة الرابعة:

```
switch ( Test )
{
case value1 :
    ≡ ; code
    break ;

case value2 :
    ≡ ; code
    break ;

⋮
default :
    ≡ ; code
    break ;
}
```

الصيغة الخامسة:

```
( مقدار الزيادة ; الشرط ; نقطة البداية )  
for  
{  
    ; code  
}
```

الصيغة السادسة:

```
( الشرط )  
while  
{  
    ; code  
    +  
    index  
}
```

الصيغة السابعة:

```
do  
{  
    ; code  
    +  
    index  
} while ( الشرط ) ;
```

=====

اوامر قد تفيدك:-

`system.out.println()`

يستخدم هذا الامر لطباعه شئ ما ، فيطبعه ثم ينزل سطر.

`system.out.print()`

يستخدم هذا الامر لطباعه شئ ما ، فيطبعه بدون ان ينزل سطر.

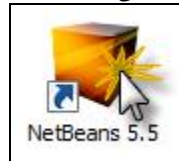
`system.exit(0)`

يستخدم هذا الامر للخروج من التطبيق.

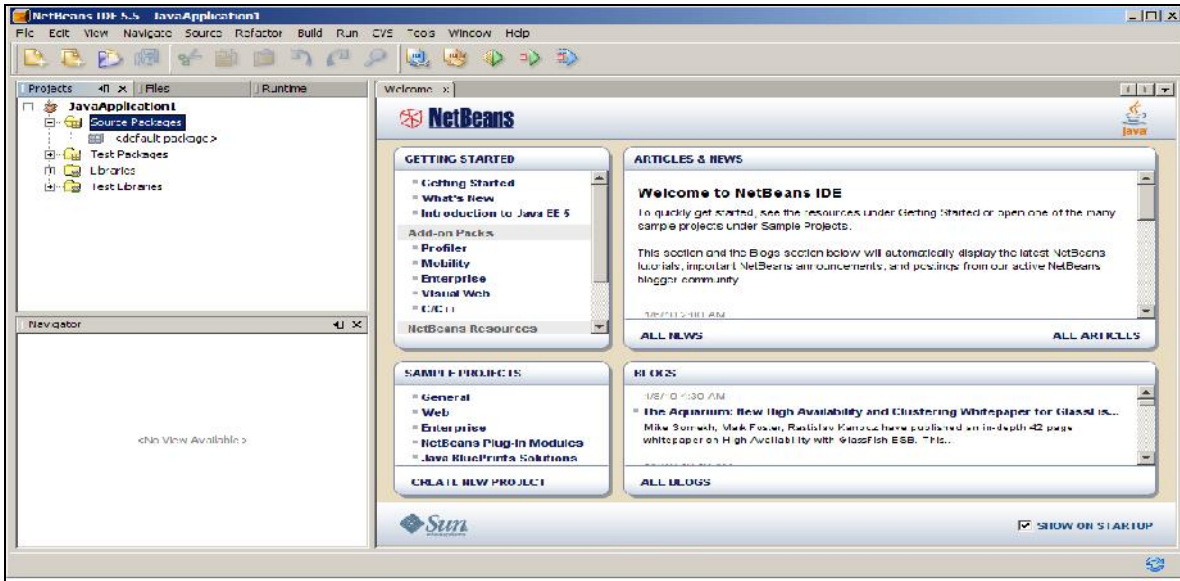
=====

طريقه فتح مشروع جديد:

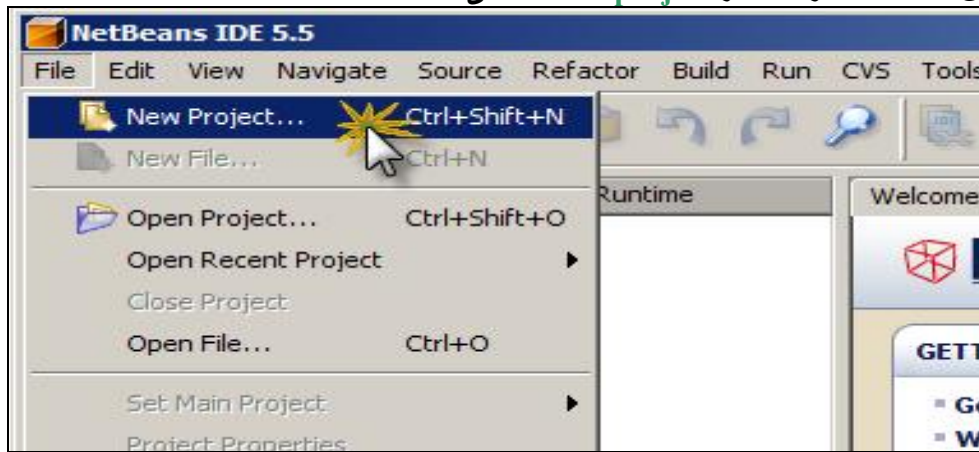
اولا: من على سطح المكتب نقوم بالضغط مرتين على ايقونه محرر لغه الجافا كالتالى:



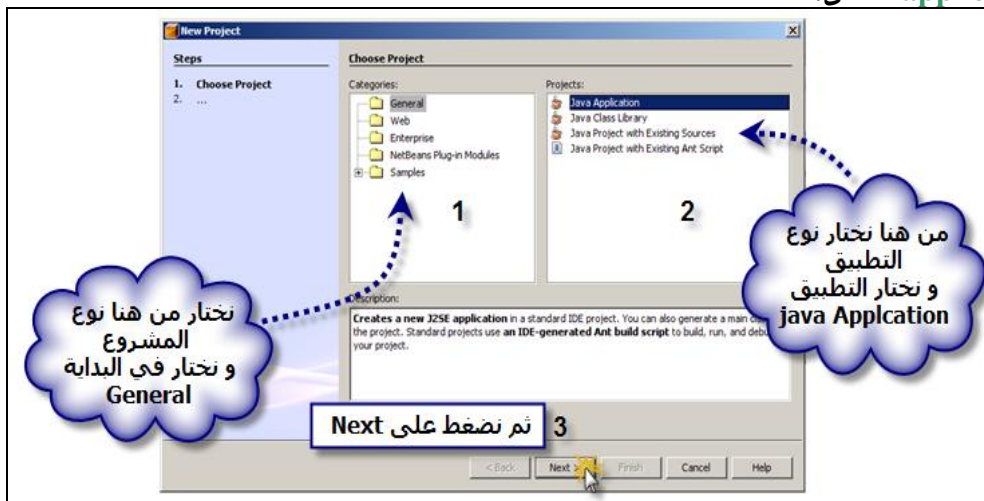
ثانيا: سوف تظهر لك النافذه التاليه للمحرر بالشكل التالي:



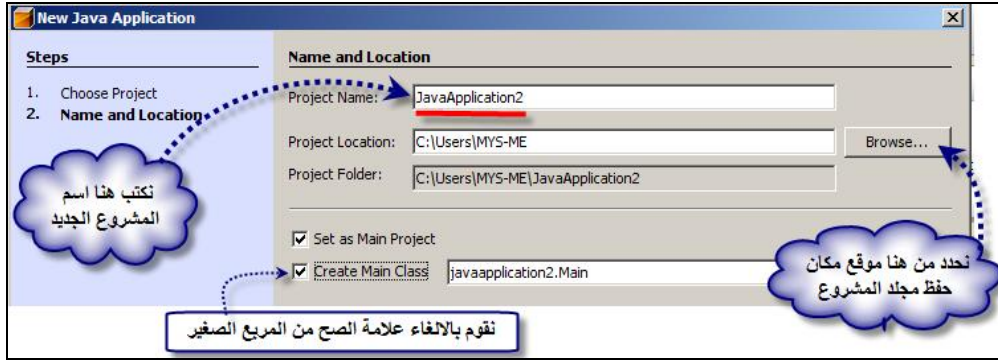
ثالثاً: من قائمة **file** سوف نختار **new project** كالتالي:



رابعاً: سوف تظهر لك نافذة جديدة، ومنها نختار نوع المشروع **General**، وكذلك نوع التطبيق **java application**، كالتالي:



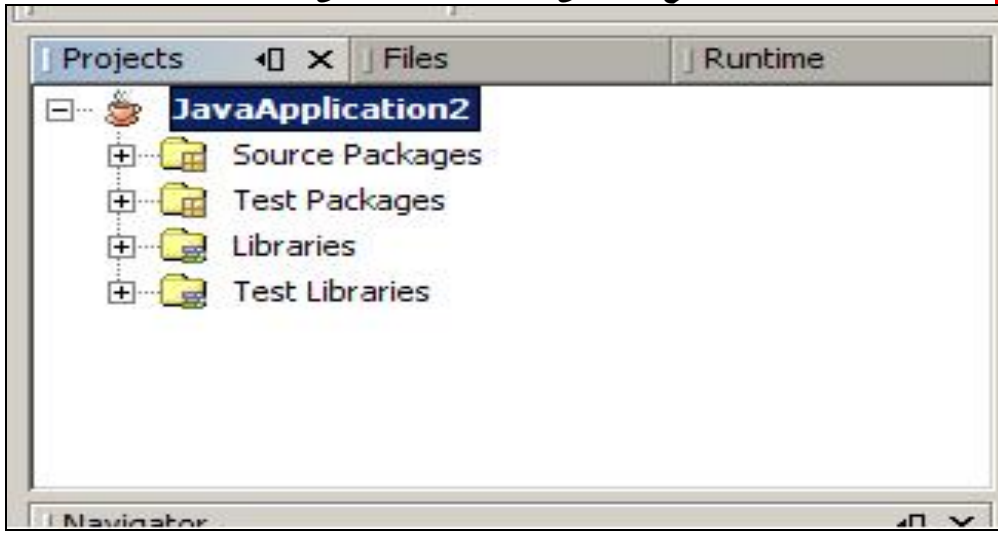
خامساً: سوف تظهر لك نافذة أخرى جديدة، ومنها نختار اسم المشروع، وموقعه، كالتالي:



سادسا: ثم الضغط على **finish**.

Finish

سابعا: سوف تظهر لك بيانات المشروع الجديد في النافذة الرئيسيه كالتالي:

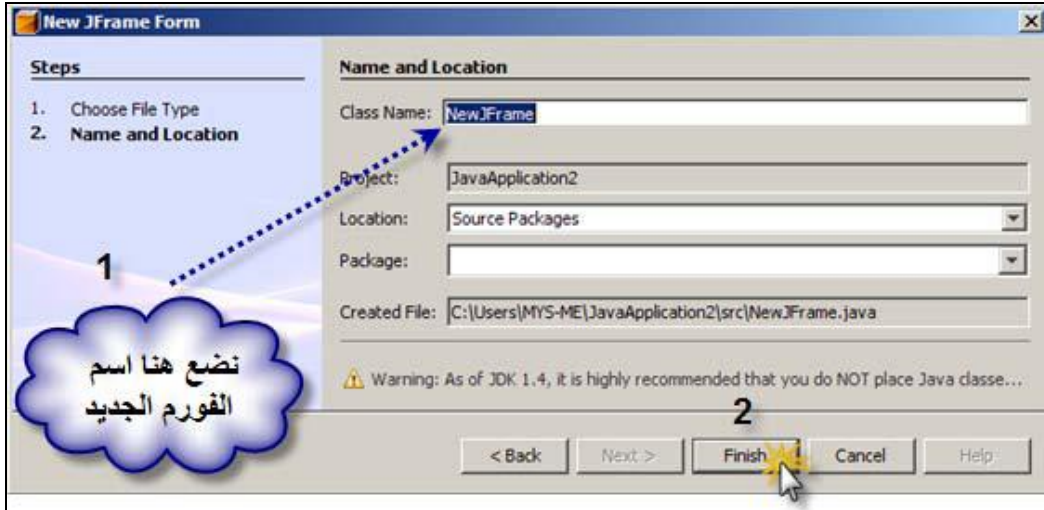


ادراج form جديد لبدأ عملية تصميم المشروع:

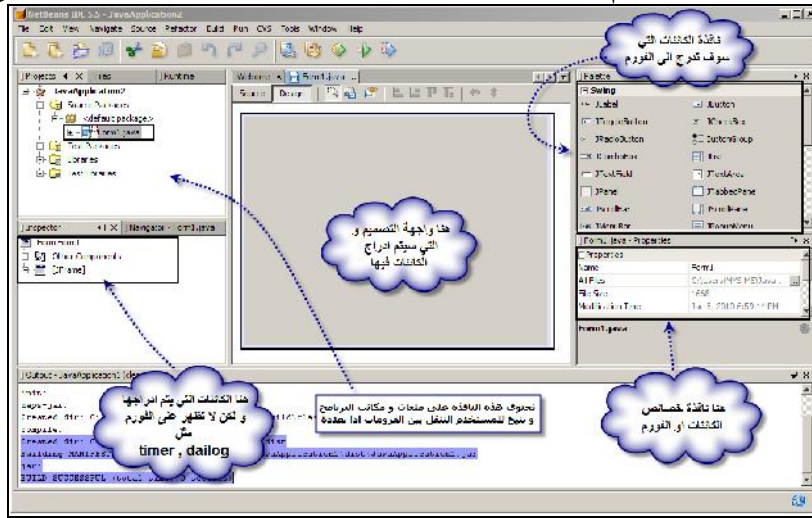
اولا:



ثانيا:



ثالثاً: سوف يظهر تبويب جديد باسم ال **form** الذى قمت بكتابته وايضا اطار العمل عليه، كالتالى:



تنفيذ البرنامج:

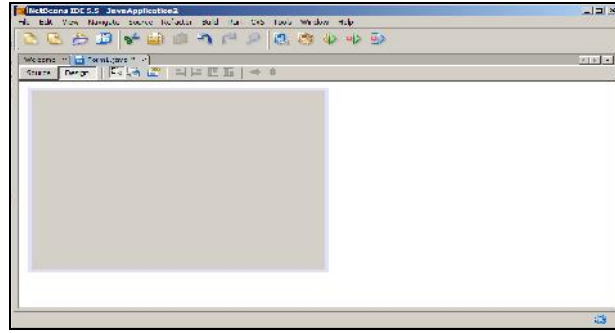
من اجل تنفيذ البرنامج الذى قمنا بتصميمه، فاننا سوف نتبع احدى الطرق التالية:
الاولى: نضغط على الزر **f5** من لوحة المفاتيح.
الثانية: نضغط على زر **run** الموجود اعلى البرنامج.

تكبير وتصغير نافذه ال form :

عند النقر المزدوج على تبويب نافذه ال **form** كالتالى:



فانه سوف يتم تكبير نافذه ال **form** و اظهارها بالكامل وذلك من اجل تحكم اكبر بشكل ال **form** ، وايضا من اجل الرجوع الى الوضعيه الاصغر يتم النقر المزدوج على التبويب نفسه كالتالى:



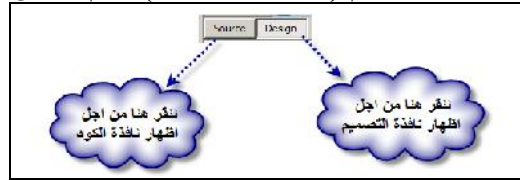
تغيير الاسم الظاهري لاي كائن:

من اجل تغيير الاسم الظاهري لاي كائن يتم النقر المزدوج عليه وتغيير اسمه كالتالي:



التنقل بين النوافذ:

من اجل التنقل بين نافذه الاكواد ونافذه التصميم (واجهه ال form) نقوم بالتالي:



اختيار حدث لاي كائن وكتابه الكود بداخله:

من اجل اختيار حدث لكائن ما وكتابه الكود بداخله نتبع ما يلي:

- ١- نحدد الكائن المراد التعديل او العمل عليه.
 - ٢- ننقر بزر الماوس الايمن عليه.
 - ٣- سوف تظهر لنا قائمة كبيرة نختار كلمة **events**.
 - ٤- سوف تظهر لنا قائمة اخرى كبيرة نختار منها تصنيف الحدث وليكن **mouse**.
 - ٥- سوف تظهر لنا قائمة اخرى كبيرة نختار منها الحدث المطلوب وليكن **mouse clicked**.
- وكل هذا موضح بالشكل التالي:



الشفرات اللازمة لبناء اي form:

عند فتح نافذه الكود فاننا نلاحظ بدايه الكود بهذه السطور:

```

public class Form1 extends javax.swing.JFrame {

    /** Creates new form Form1 */
    public Form1() {
        initComponents();
    }
}

```

تطبيقات بسيطة: 🚩

✓ التطبيق الاول: تصميم برنامج يقوم بجمع عددين واظهار النتيجة!!

الكود:

```

double No1 ;
double No2 ;
double Res ;
No1 = Double.parseDouble(jTextField1.getText() ) ;
No2 = Double.parseDouble(jTextField2.getText() ) ;
Res = No1 + No2 ;
jLabel1.setText("" + Res) ;

```

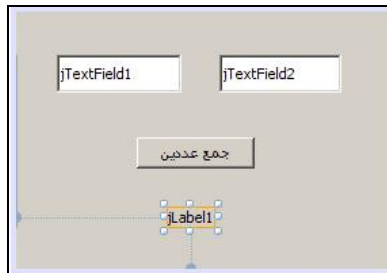
موقع الكود:

نكتب الكود السابق في الكائن (جمع عددين).

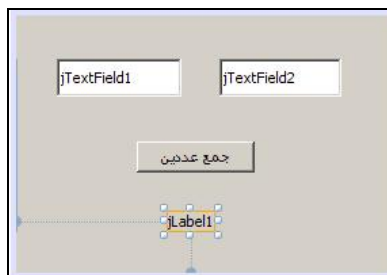
الشرح:

السطر الاول: قمنا بتعريف متغير جديد من النوع **double** يدعى **No1** وهو يمثل الرقم الاول.
السطر الثاني: قمنا بتعريف متغير جديد من النوع **double** يدعى **No2** وهو يمثل الرقم الثاني.
السطر الثالث: قمنا بتعريف متغير جديد من النوع **double** يدعى **Res** وهو يمثل الناتج.
السطر الرابع: تم اخذ النص الذي سيدخله المستخدم ل **jTextField1** وذلك باستخدام الداله **getText()**، ومن ثم تم تحويل البيانات او الارقام المدخلة من نصيه الى رقميه وذلك باستخدام الداله **parseDouble()**، ومن ثم ادراج الرقم المدخل داخل المتغير **No1**.
السطر الخامس: بالمثل كما سبق ولكن في النهايه سيتم ادراج الرقم المدخل المتغير **No2**.
السطر السادس: عمليه جمع المتغيرين وادراج القيمه الجديده في المتغير **Res**.
السطر السابع: يقوم بادراج القيمة الموجوده في المتغير **Res** الى الكائن **jLabel1**، وذلك باستخدام الداله **setText()**.

التصميم:



البرنامج:



✓ التطبيق الثاني: تصميم برنامج لاظهار نتيجة الطالب بحيث عندما يدخل المستخدم درجه الطالب يكون الخرج اما ناجح او راسب!!

الكود:

```
float m;  
m = Float.parseFloat(jTextField1.getText() );  
if (m>=0 & m<50)  
{jLabel1.setText("failed");}  
else if (m>50 & m<=100)  
{jLabel1.setText("passed");}  
else if (m==50)  
{jLabel1.setText("passed");}  
else  
{jLabel1.setText("Error!!");}
```

موقع الكود:

نكتب الكود السابق في الكائن (اظهر النتيجة).

الشرح:

فرضنا ان الدرجة النهائية للاختبار هي ١٠٠ درجة.
وعليه فاذا حصل الطالب على درجة من ٥٠ الى ١٠٠ يتعتبر ناجح!! passed!، واذا حصل الطالب على درجة من ١ الى ٤٩ يعتبر راسب!! failed!، واذا ادخل المستخدم قيم ما بخلاف ما سبق يطبع البرنامج كلمة خطأ . Error!!
وقد استخدمنا لتحقيق ما سبق من شروط جملة if الشرطيه.
السطر الاول: قمنا بتعريف متغير جديد من النوع float يدعى m وهو يمثل الدرجة التي سيدخلها المستخدم.
السطر الثاني: تم اخذ النص الذي سيدخله المستخدم ل jTextField1 وذلك باستخدام الداله `getText()` (ومن ثم تم تحويل البيانات او الارقام المدخلة من نصيه الى رقميه وذلك باستخدام الداله `parseFloat`، ومن ثم ادراج الرقم المدخل داخل المتغير m .
من السطر الثالث حتى الاخير: جملة if الشرطيه.

التصميم:



✓ **التطبيق الثالث:** تصميم برنامج يقوم بتحديد تقدير الطالب من (ممتاز) حتى (ضعيف) حسب الدرجة المدخلة اليه من قبل المستخدم!!

الكود:

```
float m;  
m = Float.parseFloat(jTextField1.getText() );  
if (m>=0 & m<50)  
{jLabel1.setText("");}  
else if (m>=50 & m<65)  
{jLabel1.setText("");}  
else if (m>=65 & m<80)  
{jLabel1.setText("");}  
else if (m>=80 & m<90)  
{jLabel1.setText("");}  
else if (m>=90 & m<=100)  
{jLabel1.setText("");}  
else
```

```
{jLabel1.setText("");}
```

موقع الكود:

نكتب الكود السابق في الكائن (اظهر النتيجة).

الشرح:

فرضنا ان الدرجة النهائية للاختبار هي ١٠٠ درجة.
وعليه:

- فإذا حصل الطالب على درجة من ٠ الى ٥٠ يكون تقديره **ضعيف** .
 - وإذا حصل الطالب على درجة من ٥٠ الى ٦٥ يكون تقديره **مقبول** .
 - وإذا حصل الطالب على درجة من ٦٥ الى ٨٠ يكون تقديره **جيد** .
 - وإذا حصل الطالب على درجة من ٨٠ الى ٩٠ يكون تقديره **جيد جدا** .
 - وإذا حصل الطالب على درجة من ٩٠ الى ١٠٠ يكون تقديره **ممتاز** .
- إذا ادخل المستخدم قيم ما بخلاف ما سبق يطبع البرنامج كلمه **خطأ** .
وقد استخدمنا لتحقيق ما سبق من شروط جملة **if** الشرطيه.

السطر الاول: قمنا بتعريف متغير جديد من النوع **float** يدعى **m** وهو يمثل الدرجة التي سيدخلها المستخدم.

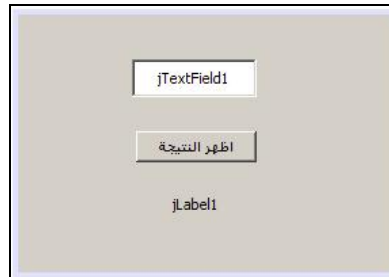
السطر الثاني: تم اخذ النص الذي سيدخله المستخدم ل **(jTextField1)** وذلك باستخدام الداله **getText()**

(،ومن ثم تم تحويل البيانات او الارقام المدخلة من نصيه الى رقميه وذلك باستخدام الداله **parseFloat**

،ومن ثم ادراج الرقم المدخل داخل المتغير **m** .

من السطر الثالث حتى الاخير: جملة **if** الشرطيه.

التصميم:



✓ **التطبيق الرابع:** تصميم برنامج لاطهار الارقام بالحروف عندما يقوم المستخدم بادخال رقم ما

مثلا:

- ١- إذا ادخل المستخدم رقم ١ يظهر لنا كلمه **one** .
 - ٢- إذا ادخل المستخدم رقم ٢ يظهر لنا كلمه **two** .
- وهكذا!! وليكن حتى رقم ٥ فقط!!

الكود:

```
int n ;
n = Integer.parseInt(jTextField1.getText()) ;
switch (n)
{
case 1 :
jLabel1.setText("one") ;
break ;
case 2 :
jLabel1.setText("two") ;
break ;
case 3 :
jLabel1.setText("three") ;
break ;
case 4 :
jLabel1.setText("four") ;
break ;
```

```

case 5 :
jLabel1.setText("five") ;
break ;
default:
jLabel1.setText("enter the correct number!!") ;
break ;
}

```

موقع الكود:

نكتب الكود السابق في الكائن (أظهر الرقم كتابة).

الشرح:

السطر الاول: قمنا بتعريف متغير جديد من النوع **int** يدعى **n** وهو يمثل الدرجة التي سيدخلها المستخدم.
السطر الثاني: تم اخذ النص الذي سيدخله المستخدم ل **jTextField1** وذلك باستخدام الدالة **getText()**، ومن ثم تم تحويل البيانات او الارقام المدخلة من نصيه الى رقميه وذلك باستخدام الدالة **parseInt**، ومن ثم ادراج الرقم المدخل داخل المتغير **n**.
من السطر الثالث حتى الاخير: جملة **switch** الشرطيه.

التصميم:



البرنامج:



✓ **التطبيق الخامس:** تصميم برنامج لاضافه الاعداد من واحد الى عشره الى كائن من نوع **jComboBox** !!

الكود:

```

jComboBox1.removeAllItems() ;
for (int i=1 ; i<=10 ; i++)
{
jComboBox1.addItem(i) ;
}

```

موقع الكود:

نكتب الكود السابق في الكائن (الدرج الارقام).

الشرح:

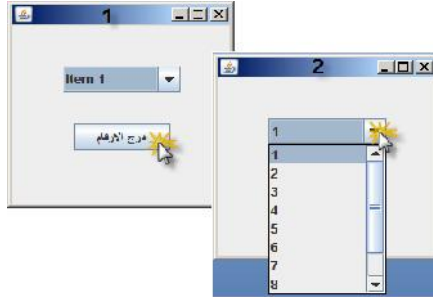
السطر الاول: يقوم بتفريغ الكائن **jComboBox1** من جميع العناصر وقد تم ذلك باستخدام الدالة **removeAllItems()**.
السطر الثاني: الحلقة التكراريه **for** والتي تحتوى على ثلاث اجزاء:
الجزء الاول: ينفذ قبل اول لفة فقط وهو **int i=1**.
الجزء الثاني: ينفذ قبل كل لفة ويسمى الشرط وهو **i<=10**.

الجزء الثالث: ينفذ بعد كل لفة ويسمى الخطوه وهو `+++i` .
السطر الرابع: يقوم باضافه العناصر الى الكائن `JComboBox1` وقد تم ذلك باستخدام الداله `.addItem()` .

التصميم:



البرنامج:



✓ التطبيق السادس: تصميم برنامج يقوم بضرب عددين و اظهار النتيجة!!

الكود:

```
double n1 ;  
double n2 ;  
double res ;  
n1 = Double.parseDouble(jTextField1.getText() ) ;  
n2 = Double.parseDouble(jTextField2.getText() ) ;  
res = n1 * n2 ;  
jLabel1.setText("" + res) ;
```

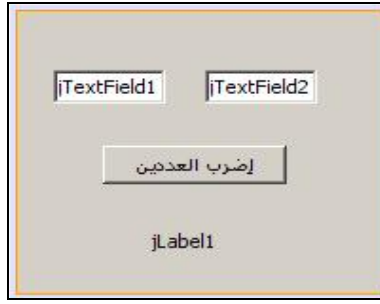
موقع الكود:

نكتب الكود السابق في الكائن(ضرب عددين).

الشرح:

السطر الاول: قمنا بتعريف متغير جديد من النوع `double` يدعى `n1` وهو يمثل الرقم الاول.
السطر الثاني: قمنا بتعريف متغير جديد من النوع `double` يدعى `n2` وهو يمثل الرقم الثاني.
السطر الثالث: قمنا بتعريف متغير جديد من النوع `double` يدعى `res` وهو يمثل الناتج.
السطر الرابع: تم اخذ النص الذي سيدخله المستخدم ل `JTextField1` وذلك باستخدام الداله `getText()` (ومن ثم تم تحويل البيانات او الارقام المدخلة من نصيه الى رقميه وذلك باستخدام الداله `parseDouble` ،ومن ثم ادراج الرقم المدخل داخل المتغير `n1` .
السطر الخامس: بالمثل كما سبق ولكن في النهايه سيتم ادراج الرقم المدخل المتغير `n2` .
السطر السادس: عمليه ضرب المتغيرين وادراج القيمه الجديده في المتغير `res` .
السطر السابع: يقوم بادراج القيمه الموجوده في المتغير `res` الى الكائن `JLabel1` ، وذلك باستخدام الداله `.setText` .

التصميم:



✓ **التطبيق السابع:** تصميم برنامج يقوم بادراج الاعداد من ٢٠ حتى ١٠ اى تنازليا من الاكبر الى الاصغر واضافتها فى كائن من النوع JComboBox !!

الكود:

```
jComboBox1.removeAllItems() ;
int x ;
x = 20 ;
while (x >= 10)
{
jComboBox1.addItem(x) ;
x-- ;
}
```

موقع الكود:

نكتب الكود السابق فى الكائن (الدرج الارقام).

الشرح:

السطر الاول: يقوم بتفريغ الكائن JComboBox1 من جميع العناصر وقد تم ذلك باستخدام الداله `.removeAllItems()`

السطر الثانى: تعريف متغير جديد من النوع `int` يدعى `x` .

السطر الثالث: تهيئه المتغير السابق تعريفه بقيمه قدرها 20 .

من السطر الرابع حتى الاخير: حلقه `while` التكراريه، حيث اذا كانت قيمه `x` اكبر من او تساوى ١٠ يتم تنفيذ ما بداخل الحلقه والرجوع مره اخرى وتكرارها حتى لا يتحقق هذا الشرط!!
وجسم تلك الحلقه يحتوى على امرين:

الامر الاول: اضافه العناصر الى الكائن JComboBox1 وقد تم ذلك باستخدام الداله `.addItem()` .

الامر الثانى: لتكن قيمه `x` الجديده هي `1 + x` .

التصميم:



✓ **التطبيق الثامن:** تصميم برنامج يقوم باضافه الاعداد الفرديه من ٢٠ حتى ١ وذلك الى كائن من النوع JComboBox !!

الكود:

```
jComboBox1.removeAllItems() ;
int x ;
x = 19 ;
do
{
jComboBox1.addItem(x) ;
```

```
x = x - 2 ;
}
while ( x >= 1 ) ;
```

موقع الكود:

نكتب الكود السابق فى الكائن (ادرج الارقام).

الشرح:

السطر الاول: يقوم بتفريغ الكائن jComboBox1 من جميع العناصر وقد تم ذلك باستخدام الداله **removeAllItems()**.

السطر الثانى: تعريف متغير جديد من النوع **int** يدعى **x**.

السطر الثالث: تهيئه المتغير السابق تعريفه بقيمه قدرها 19.

من السطر الرابع حتى الاخير: حلقه **do** ثم **while** التكراريه، حيث نلاحظ ان هذه الحلقه سوف تنفذ فيها الاوامر التى تلى كلمه **do** وهما:

```
jComboBox1.addItem(x) ;
x = x - 2 ;
```

ثم نرى الشرط الذى يلى كلمه **while** وهو:

```
( x >= 1 ) ;
```

فاذا تحقق نعود ننفذ ما سبق من اوامر مره اخرى، واذا لم يتحقق نخرج من الحلقه، وبالتالي فان هذه الحلقه سوف تنفذ مره واحده على الاقل.

التصميم:



✓ **التطبيق التاسع:** تصميم برنامج يتضمن كود برمجى يقوم باظهار رساله للمستخدم تخبره بانه لم يدخل القيمه فى كائن من نوع **jTextField** !!

الكود:

```
if (jTextField1.getText().length() == 0)
{
JOptionPane.showMessageDialog(null, "please insert the data ?", "worrying", 2);
}
```

موقع الكود:

نكتب الكود السابق فى الكائن (تحقق من الادخال).

الشرح:

الداله length() تم استخدامها لعملية التحقق من وجود نص ام لا داخل الحقل، حيث ان البرنامج يتعرف عليها كمسافه وليس كفراغ.

JOptionPane: مكتبه استدعاء دوال صناديق الحوار.

null: تشير الى عدم وجود اضافات متقدمه.

الداله showMessageDialog(): تستخدم فى اظهار الرساله.

بالنسبه للرساله التى سوف يتم اظهارها:

١- محتواها: **please insert the data ?**

٢- عنوانها: **worrying**

٣- شكل الايقونه: لتحديد الازرار التى نريد ظهورها فى الرساله ولتكن 2 وتشير الى من 0 حتى 3.

البرنامج:



✓ **التطبيق العاشر:** تصميم برنامج يتضمن كود برمجي يقوم بعملية مسح نص مكتوب داخل كائن من نوع JTextField بشرط عند تنفيذ امر المسح تظهر رساله تأكد من عملية المسح او الغاءه.

الكود:

```
int sign ;
sign = JOptionPane.showConfirmDialog(null, "do you want to delete ?",
"delete", 0,3) ;
if (sign == 0)
{
jTextField1.setText("");
}
```

الشرح:

السطر الاول: متغير جديد **sign** تم تعريفه من النوع **int**، يقوم باستقبال القيمة الممرره والتي تمثل ما قام المستخدم بضغطه من الازرار **yes** او **no** او **cancel** .

جمله if: تحتوى على عملية المقارنه بين القيمة الممرره والقيمة المار التعامل معها كالتالى:

if (sign == 0)

القيم الممرره وما ترمز له:

Yes refer to 0

No refer to 1

Cancel refer to 2

JOptionPane: مكتبة استدعاء دوال صناديق الحوار.

null: تشير الى عدم وجود اضافات متقدمه.

الداله () showConfirmDialog: تستخدم فى اظهار الرساله.

بالنسبه للرساله التي سوف يتم اظهارها:

١- **محتواها:** **do you want to delete ?**

٢- **عنوانها:** **delete**

٣- **شكل الايقونه:** لتحديد الازرار التي نريد ظهورها فى الرساله ولتكن من **0** حتى **2** .

البرنامج:



تم بحمد الله

الكتاب قيد الاستكمال