

تم تحميل الملف من موقع
البوصلة التقنية
www.boosla.com

الكتاب الثاني من سلسلة كتب

الجرافيكس باستخدام لغة الجافا

ب عنوان

مقدمة في البرمجة الشيئية (OOP).

ماهر محمد محمد فايز

طالب بالفرقة الرابعة - كلية الهندسة - جامعة المنصورة - مصر

قسم الحاسبات والنظم

30 من مارس 2006م

بسم الله الرحمن الرحيم



- مقدمة-

- كلمة عن الكتاب الأول من السلسلة-

- الرد على بعض التعليقات-

- أرسل تعليقك لى-

- تعريف بالسلسلة-

مقدمة

بسم الله الرحمن الرحيم الحمد لله رب العالمين، وأشهد أن لا إله إلا الله وحده لا شريك له، وأشهد أن محمدًا عبده ورسوله،
صلى الله عليه وعلى آله وسلم تسليماً.

الكتاب الثاني من سلسلة كتب الجرافيكس باستخدام لغة الجافا

وقال النبي صلى الله عليه وسلم: (من عرف نفسه عرف ربه) ولسنا نقول: إن الله يعرف بالمخلوقات، بل المخلوقات كلها تعرف بالله، لكن معرفته تزيد بالنظر في مخلوقات الله.

جزء من مقدمة كتاب **توحيد الربوبية** لشيخ الإسلام ابن تيمية- برنامج مجموعة فتاوى ابن تيمية- إنتاج موقع روح الإسلام

الرد على بعض التعليقات عن الكتاب الأول

أود أولاً أن أشكر كل من قرأ الكتاب وأرسل تعليقا عليه ...

الكتاب الثاني من سلسلة كتب الجرافيكس باستخدام لغة الجافا

كيف أحصل على لغة الجافا ؟ ما هي الشركة التي تملكها الآن ؟ كيف أحصل على معلومات عن أدوات اللغة؟ هل هذه اللغة مجانية ؟

أسهل الطرق من موقع www.Java.sun.com قم بتحميل ما يسمى SDK وأيضا IE والشركة التي تملكها الآن هي شركة SUN وتستطيع الحصول على معلومات عن اللغة من الموقع ذاته.

كيف أحصل على برنامج محرر نصوص ذكي عن محرر النصوص العادي وكيف أتعامل معه؟

يمكنك تحميل برنامج KAWA أو JAVA creator أو النسخة الحديثة من TEXT bad.

هل يوجد مسابقات لهذه اللغة ؟

تعاونت شركة Sun مع شركة siemens على عمل مسابقة Java masters للطلبة والمحترفين هذا العام بدأت منذ فبراير وتنتهي في مايو المقبل من عام 2006 وتقريبا تدور المسابقة على عمل برنامج يتم تثبيته على جهاز جوال (محمول) وتستطيع الحصول على معلومات والاشتراك في هذه المسابقة من على موقع www.javamasters.org ويتم اختيار الفائز عن طريق استفتاءات الزائرين وليس تقييما من قبل الشركة ولذلك إن لم تكن مشاركا فلنكن مصوتا والله الموفق للجميع ..

لإرسال التعليقات

البريد الالكتروني : islam Maher4@yahoo.com

تعريف بالسلسلة

هي سلسلة كتب الجرافيكس باستخدام لغة الجافا وتتضمن السلسلة الكتب التالية:-

1- مقدمة في البرمجة باستخدام لغة الجافا.

2- مقدمة في البرمجة الشيئية (OOP).

3- جمل الإدخال في لغة الجافا.

4- عمل جافا أبلت Applets.

5- أمثلة على الرسم .

6- الحركة.

7- عرض الشرائح.

8- لغة Html .

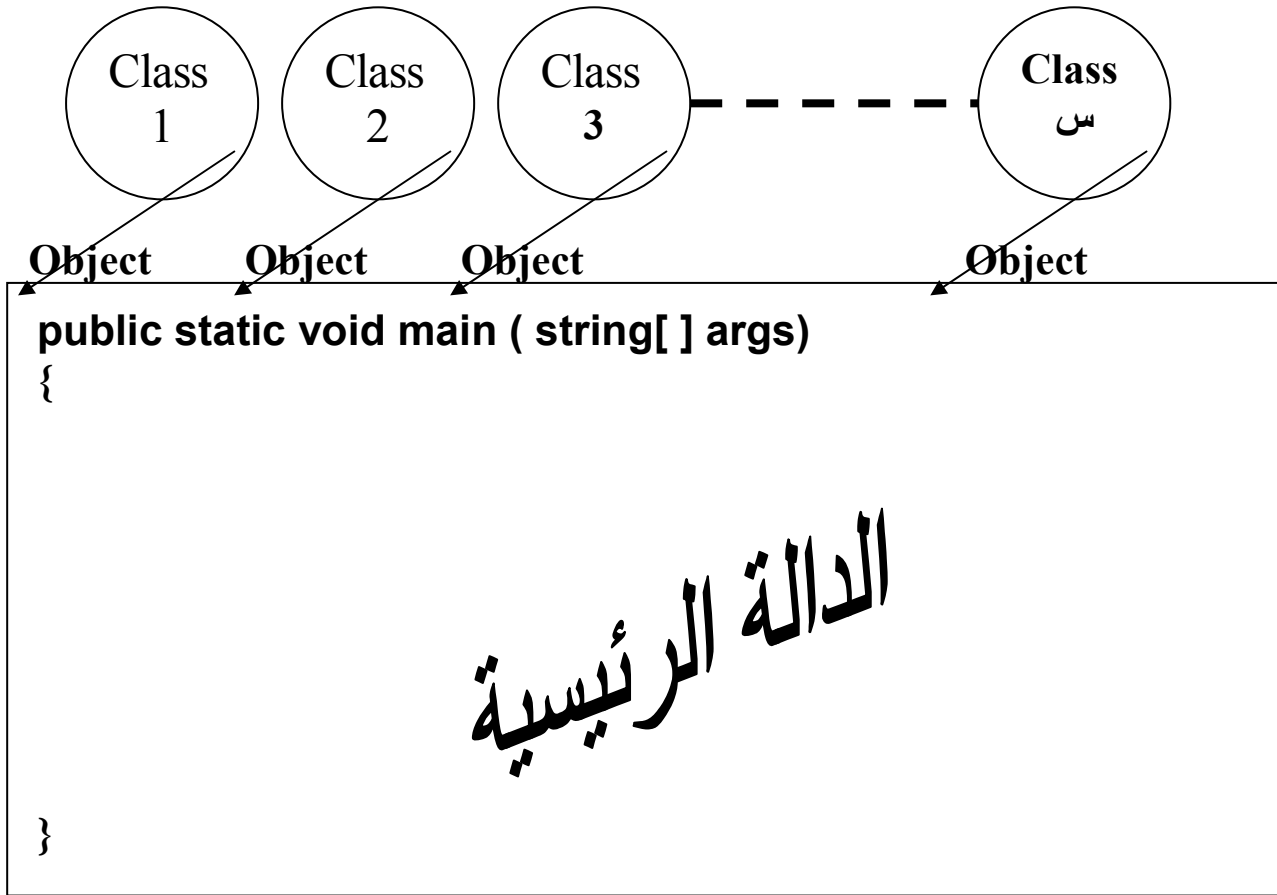
الكتاب الثاني : مقدمة في البرمجة الشيئية (OOP)

- (1) ماذا تعنى OOP.
- (2) ما هو الCLASS.
- (3) أنواع الCLASS.
- (4) بنية الCLASS.
- (5) كيفية كتابة كود للCLASS.
- (6) إنشاء Objects.

(1) ماذا تعنى OOP

ال OOP هى البرمجة الشيئية وهى اختصار لـ Object Oriented Programming وهو الفكر الحديث للبرمجة وهى طريقة تعتمد على كتابة كود برمجى داخل مجموعة من ال (Classes) بحيث يحتوى كل Class على جزء من الكود المطلوب إدراجه داخل البرنامج ويتم إدراج هذه الاكواد المبعثرة

داخل ال **Classes** عن طريق أخذ **Object** منها داخل الدالة الرئيسية دال ال **Main** كما أشرنا فى نهاية الكتاب الأول (شكل 1. 2)



(شكل 1. 2)

ملحوظة

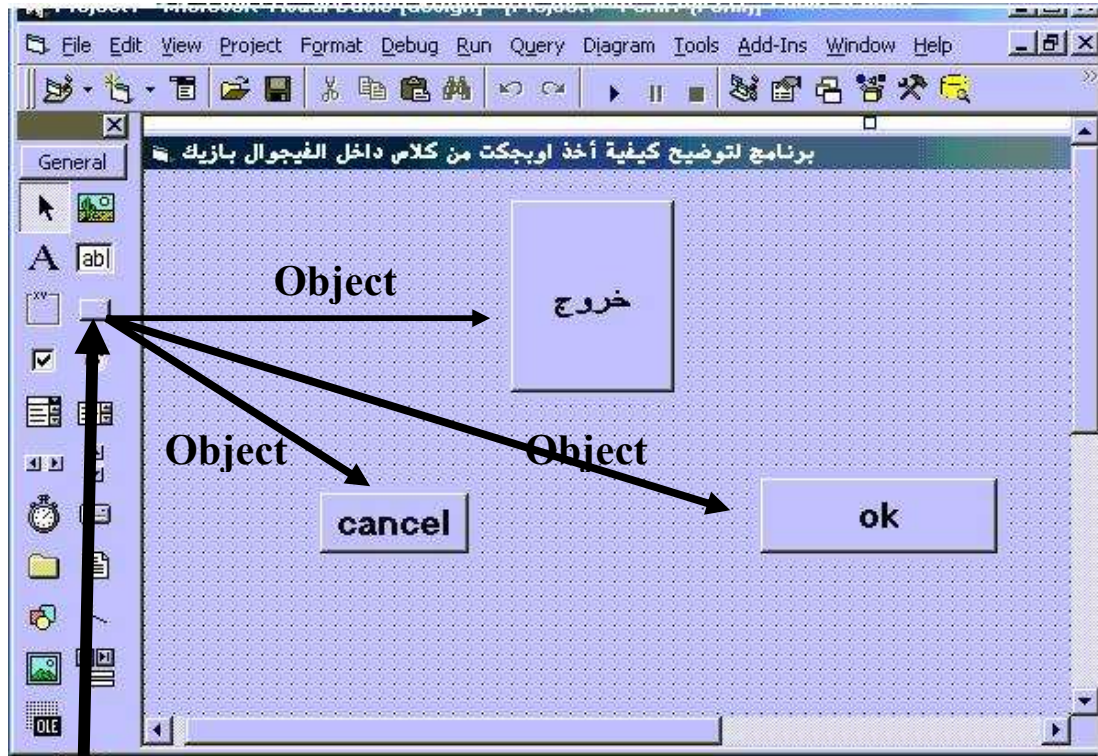
- لا يمكنك أن تستخدم الكود المكتوب داخل ال **Class** الا باستخدام الطريقة السابقة وهى بناء **Object** منها...
- يمكنك بناء أكثر من **Object** من **Class** واحد ...
- قد يختلف خصائص **Object** عن آخر مأخوذين من نفس ال **Class** حسب احتياجات المبرمج .
- مما سبق يطرح تساؤل لماذا نستخدم البرمجة الشيئية ونترك برمجة الأحداث السابقة كما فى فيجوال بازيك 6 مثلا؟؟ الاجابة هى انه فى برمجة الأحداث كان يستخدم نفس فلسفة البرمجة الشيئية من قبل مصممي اللغة ويحجب ذلك عن المستخدمين ولما تحايل المستخدمين على برمجة الأحداث للحصول على بعض ميزات لم تقدمها لهم . دفع ذلك مصممي اللغة إنشاء برمجة شيئية توفى باحتياجات المبرمجين ولغة الجافا من أوائل اللغات التى تبنت هذا الفكر البرمجي وبالمثال تتضح المقال :-

مثال :-

داخل الفيجوال بازيك مثلا يمكنك وضع أكثر من مفتاح على ال **Form** أو النموذج . حقيقة هذا المفتاح هو **Object** تم بناءه عن طريق استخدام **Class** تسمى مثلا **Button** موجودة داخل الفيجوال بازيك وتحتوى هذه ال **Class** على الكود الذى يقوم ببناء المفتاح . والا كيف تم إنشاء هذه المفاتيح ؟وبذلك قد تم حجب هذه الميزة عن المبرمج واقتصرت على مصمم اللغة .

ملاحظة :

جميع المفاتيح يتم بناؤها من **Class** واحد إلا أنها تختلف عن بعضها في العديد من الخصائص مثل الطول والعرض. (شكل 2. 2)



Class button

```
{  
هنا مكتوب كود اللازم لعمل مفتاح يمكن  
التغيير في خصائصه  
}
```

(شكل 2. 2)

(2) ما هو الـ CLASS

لننا تطرقنا إلى ذلك في النقطة السابقة المهم أن الـ **Class** عبارة عن وعاء كبير يحتوى على كود مخصص لعمل وظائف حسب رغبة المبرمج وعند احتياج المبرمج إليه ما عليه إلا اخذ **Object** من هذا الـ **Class** ليستطيع الاستفادة من هذا الـ **Class**.

تستطيع معرفة أسماء ال **Classes** ووظائفها التي داخل لغة الجافا من داخل موقع الشركة www.java.sun.com وبتحميل 40 ميجا من JDK التي داخل الموقع تستطيع تصفحها من على جهازك.

ملاحظة : في بعض الأحيان يكون السيرفر مغلق للدول العربية لزيادة الضغط عليه لذا عليك تكرار المحاولة في وقت آخر..

CLASS أنواع ال (3)

هناك نوعين رئيسيين

- (1) **جاهز** في اللغة تستطيع الاستفادة منه مباشرة بأخذ **object** و أكثر منه مباشرة ويطلق عليه **(Build-in Classes)**.
- (2) أنت الذي تقوم بعمله حسب رغباتك ويطلق عليه **(User-Defined-Classes)** ويجب عليك أولاً عمل **Class** ثم بعد ذلك الاستفادة منه عن طريق أخذ **Object** أو أكثر منه .

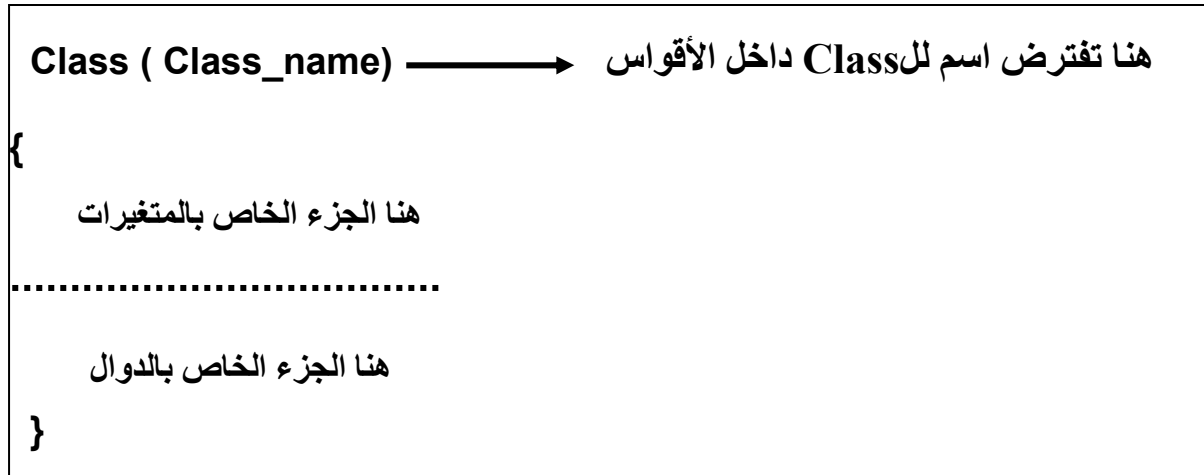
CLASS بنية ال (4)

إن شاء الله تعالى من بداية هذه النقطة وحتى آخر السلسلة هي الجزء الأكثر جدية للتعامل مع لغة الجافا فكل ما سبق ما هو إلى مقدمه لأي لغة تتعامل مع البرمجة الشيئية وليست الجافا وحدها..

- أي Class عبارة عن جزأين رئيسيين :-

(1) **Variables** : جزء خاص بالمتغيرات (تكلما عنها في الجزء الأول من السلسلة) وعادة يكون في أول الكود الخاص بال Class.

(2) **Methods** : جزء خاص بمجموعة دوال التي يحتويها هذا ال Class وهي الجزء هو المميز لل Class. (شكل 2. 3)



(شكل 2. 3)

ولنأخذ كل جزء بشكل من التفصيل :

(1) المتغيرات (Variables):

وهو الجزء الأول من أي Class كما تعرفنا وفائدة هذه المتغيرات هو عمل اختلاف ما بين ال Object في بعض الخواص . معنى ذلك أن كل Object يتم أخذه من أي Class له قيم للمتغيرات تميزه عن غيره.

مثال :-

- لو قمنا بعمل Class وظيفته هي رسم مستطيل .. نريد رسم مستطيل طوله 5 ووحده طول وعرضه 10 ووحده طول - نريد أيضا رسم مستطيل طوله 8 ووحده طول وعرضه 9 ووحده طول فماذا نفعل؟؟

الحل :-

- (1) نقوم بإنشاء Class يحتوى على متغيرين أحدهم خاص بالطول والآخر خاص بالعرض.
- (2) نقوم بأخذ Object من هذا الClass ويتم إسناد قيمه لمتغير الطول ولتكن 5 والعرض 10.
- (3) نقوم بأخذ Object آخر من هذا الClass ويتم إسناد قيمه لمتغير الطول ولتكن 8 والعرض 9.

الصيغة العامة للمتغيرات داخل الClass

اسم المتغير Δ نوع المتغير

أمثله:

```
Int x ;  
Float y;  
String z;
```

(1) الدوال (Methods):

وهو الجزء الثانى من بنية ال Class والتي عادة يلى المتغيرات

والصيغة العامة لأى داله كما يلى..

(قيم يتم إعطائها إلى الدالة إن لزم الأمر) اسم للدالة Δ نوع القيمة الذى يعود بها الدالة
{

كود الدالة

القيمة الذى يعود بها الدالة Δ Return
}

أظن الآن أن الكل متعطش للأمثلة :-

(1) داله لحساب مجموع رقمين:

Ex1 :

نوع القيمة التي تعود بها الدالة
قيم يتم إعطاؤها إلى الدالة
اسم الدالة

```
Float Sum( Float x , Float y )  
{  
Float z ;  
  
Z= x + y ;  
Return z ;  
}
```

توضيح:

- هذه الدالة تعود بقيمه لمتغير z من نوع Float ولذلك وضعنا نوع القيمة التي تعود بها الدالة بـ Float.

مثال آخر:

Ex2 :

```
String easy(  
{  
  
return "هذه الدالة لا تأخذ قيم وتعود بجمله حرفيه فقط"  
}
```

لا يوجد متغيرات

ملاحظات :-

- المثال السابق مختلف عن ما قبله في انه لا يمرر متغيرات إلى الدالة وذلك لعدم احتياج الدالة لذلك فالدالة وظيفتها الرجوع بجمله فقط أى أنه تختلف أشكال الدوال حسب الوظيفة المنوط بها ..
- هناك بعض الدوال لا تعود بقيمة أصلا هذه الدوال لا تحتوى على جملة Return. وبالتالي لا نكتب نوع للقيمة التي نعود بها ولاكن نكتب كلمة Void ...

مثال : داله تقوم بالطباعة فقط

```
Void maher(int x)  
{
```

لا تحتوى على جملة Return

للاتصال بالكاتب : Islammaher4@yahoo.com

System.Out.Println (x)

}

كيفية كتابة كود ال CLASS (5)

كما ذكرنا سابقا أن ال Class يتكون من جزأين

(1) جزء خاص بالمتغيرات ويكون له الصورة التالية:

اسم لل Class Δ Class

{

هذه المتغيرات يتم تحديد قيمه لها في الجزء الثاني من ال Class داخل داله تسمى ال Constructor

(2) الجزء الثاني والذي يختص بعمل الدوال أو الطرق يتكون من جزأين ...

- دالة ال Constructor :-

كما تعرفنا سابقا أن هذه الدالة خاصة بتحديد قيم للمتغيرات التي في الجزء الأول من ال Class وهي داله بنفس اسم ال Class ولا يوجد بها جملة return ويتم استخدامها عند أخذ Object من هذا ال Class وتأخذ هذه الدالة الصيغة التالية :

(المتغيرات التي تمرر للدالة) اسم ال Class

{

كود دالة ال constructor

}

- الدوال العادية وتستخدم لأداء بعض الوظائف كما أشرنا في النقطة الرابعة من هذا الكتاب وتأخذ الصيغة التالية :-

(بعض المتغيرات التي تمرر للدالة) اسم الدالة Δ نوع القيمة التي تعود بها الدالة

```
{  
    كود الدالة  
}
```

وبالمثال تتضح المقال :

```
Class myclass  
{  
    int x ;  
    float y;  
    string z;
```

الاسم لل Class

بعض المتغيرات التي ستعطى لها قيم لاحقاً

```
myclass (int a , float b , string c)
```

```
{  
    x=a;  
    y= b;  
    z =c;  
}
```

دالة constructor وفيها يتم اعطاء قيم للمتغيرات السابقة

```
float sum (float a , float b)
```

```
{  
    return a + b;  
}
```

دالة تجمع رقمين

```
int square(int a)
```

```
{  
    return a * a;  
}
```

دالة لإيجاد مربع العدد

```
}
```

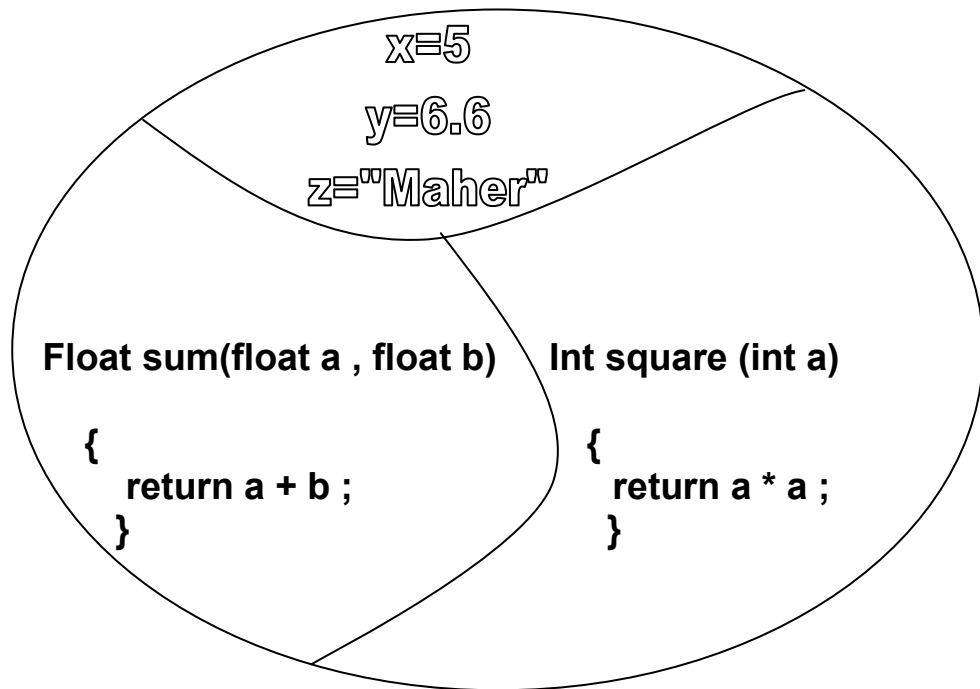
(6) إنشاء Objects

كما ذكرنا من قبل أنه لا يمكن استخدام الـ **Class** مباشرة ولكن للاستفادة منه لابد أن نأخذ منه **Object** وهذا هو ما نحن سوف نستعرضه الآن ويتم ذلك كما يلي :

(بعض المتغيرات) اسم الـ **new Class** = اسم للـ **object** اسم الـ **class** المرغوب أخذ **Object** منه على سبيل المثال لنفترض أننا قمنا بإنشاء **Class** كما في النقطة الخامسة من هذا الكتاب ونريد عمل أو إنشاء **Object** من هذا الـ **Class**.

اسم الـ **Object**
 ↓
Myclass object11 = new myclass (5 , 6.6 , " Maher ");
 ↑
 اسم الـ **Class**
 ↑
 قيم لدالة الـ **constructor**

يمكن كتابة ما سبق داخل دالة الـ **main** لأخذ **Object** من الـ **Class** المسمى **myclass** والشكل التالي يوضح شكل منطقي للـ **object** في ذاكرة الجهاز (شكل 2. 4)



(شكل 2. 4)

- المرجو من الرسم السابق أن الـ **Object** الذي يسمى **Object1** المأخوذ من **Class** يسمى **myclass** يحتوي الآن على جميع الدوال التي كانت في **myclass** وأيضا يحتوي على المتغيرات لكن كل **object** له قيم للمتغيرات تختلف عن أي **Object** آخر مأخوذ من نفس الـ **Class**..

- السؤال الآن إذا كان حقا أن الـ **Object** الذى قمنا بإنشائه من الـ **Class** يحتوى على المتغيرات والدوال فكيف استفيد منها أو أستخدمها داخل البرنامج..

(1) استخدام المتغيرات الخاصة بالـ **Object**

الصيغة المستخدمة لذلك داخل الكود كما يلى :-

اسم المتغير . اسم الـ **object**

مثال:

إسم المتغير
↓
System . Out. Print (" الرقم هو " + Object1.x) ;
↑
إسم الـ **object**

يكون خرج هذا السطر هو طباعة هذه الجملة باعتبار أننا ما زلنا نبني على البرنامج السابق هو ..

الرقم هو 5

(2) استخدام الدوال الخاصة بالـ **Object** :-

يمكن استخدام الدوال بطريقة مشابهة لاستخدام المتغيرات والصيغة العامة لها كما يلى :

اسم الدالة . اسم الـ **object**

مثال :

Float result ;

إسم المتغير
↓
Result = Object1.Sum (5.5 ; 9.5) ;
↑ ↓
إسم الـ **object** القيم

System . Out. Print (" الرقم هو " + result) ;

يكون ناتج هذا المثال هو طباعة جملة

الرقم هو 15

ملاحظات :-

- كما ذكرنا سابقا أنه يمكنك أخذ أكثر من **Object** من **Class** وكل **Object** له قيم للمتغيرات خاصة به واليك الأمثلة التالية :-

```
Public Static Void main( String[ ] args)
{
```

رقم 1 Object myclass Object_1 = new myclass(8 , 9.5 , " Ahmed ")

رقم 2 Object myclass Object_2 = new myclass(5 , 3.4 , " Ahmed ")

هنا يتم استخدام المتغيرات Float result = Object_1.y + Object_2.x

```
System . Out. Print ("الرقم هو" + result ) ;
}
```

سؤال: ما هو ناتج الكود السابق ???

أرجو إرسال الحل إلى :

بريد الكتروني : islam Maher4@yahoo.com

إلى اللقاء إن شاء الله في الكتاب الثالث
من هذه السلسلة وهو تحت عنوان (جمل الإدخال في لغة الجافا) انتظروني ...

أخيرا لا يسعني إلى أن أقول (فإن كان من توفيق فمن الله وحده وان كان من ذلل أو سهو أو نسيان أو جهل
فمنى ومن الشيطان) والسلام عليكم ورحمة الله وبركاته .

ماهر محمد محمد فايز
الفرقة الرابعة - كلية الهندسة - جامعة المنصورة - مصر
قسم الحاسبات والنظم
30 مارس 2006م

