Java

Java is a technology introduced by Sun Microsystems in 1995. it consist of the following components:

1. Programming Language

1. Object-Oriented Programming Language (OOP).

- العنصر الأساسي في لغة Java هو الصنف (Class) , كل برنامج يحتوي على الأقل على صنف واحد. وكل صنف يحتوي على الأقل على طريقةMethod واحدة اسمها Main.
- الصنف يتألف من طرق Methods التي تصف السلوك Behavior وحقول Fields التي تمثل المواصفات Attributes
- الكائنات Objects يتم خلقها من الأصناف Classes.
- هناك نوعين من البرامج في Java:

1. Application Program : برامج تكتب بلغة Java وتنفذ مباشرة من قبل المفسرJVM.
2. Applet Program: وهو برنامج تفاعلي ينفذ من قبل متصفحات الانترنت على أي حاسوب.

2. Platform Independent (architectural)
   - Platform means a particular processor.
3. Distributed.
4. Multithreaded.
5. Robust, Portable (deal with different operating systems).
   - For each java class, the Java ***compiler*** produces a ***bytecode*** output (.class).

2. Application –Programming Interface (API)

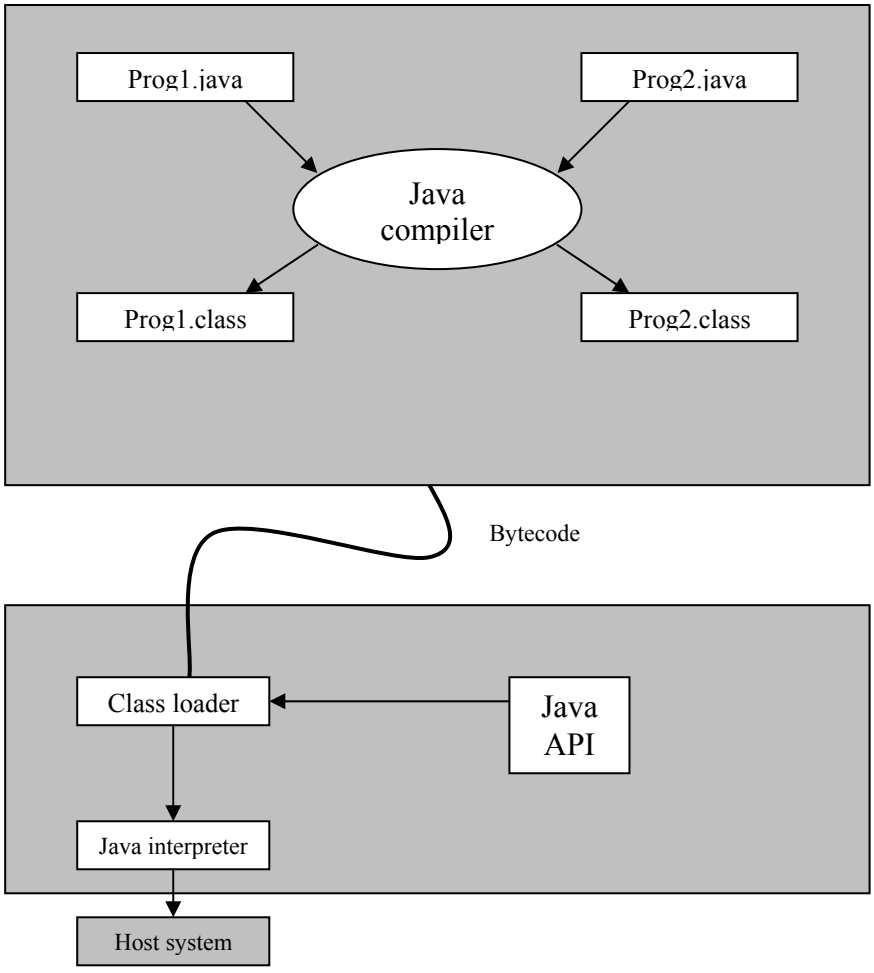
Java provide a set of packages for graphics, I/O, utilities, and networking. Such as java.lang, java.awt, java.io, java.net.

- لغة Java توفر مجوعة من الأصناف الجاهزة المجمعة والمنضمة في حزم Packages تعرف بـ(Application programming Interface(API
  - هناك نوعين من الحزم Packages في Java:
    1. الحزم الأساسية التي تبدأ بكلمة Java.
    2. حزم الامتداد والتي تبدأ بكلمة javax.

3. Java Virtual Machine (JVM):

JVM consist of **class loader** (to load .class files from the Java program and Java API), and **a Java interpreter** that execute the bytecode

- **Java platform** consist of JVM and Java API, which is implemented on top of a host operating system, such as UNIX or Windows;



- برامج Java تنفذ وتفسر من قبل Java Virtual Machine (JVM) الذي يوفر Native operating System.

- **برنامج بسيط بلغة Java**

```
// welcome.Java: this application program print welcome to Java!
 Public class welcome{
    Public static void main (String args[]){
      System.Out.Println ("welcome to Java!");
     }
  }
```

- ملحوظة هامة: عند حفظ البرنامج أعلاه نحفظه بنفس اسم الصنف وهو welcome.java
- البرنامج أعلاه يتألف من:

1. تعليق Comment // , وهناك صيغة أخرى للتعليقات في Java وهي /* */
2. الكلمات المحجوزة, وهي:...,Class, Static, Void, public
3. الموجهات:Public, Static
4. التعليمات, وهي: System.Out.println( );
5. الكتل{ }
6. الطرق, استخدمن الطريقة Println
7. الطريقة main التي تمثل بداية البرنامج.
8. الأصناف, استخدمنا هنا صنف واحد وهو welcome.

المعرفات Identifiers

تستخدم لتسمية المتغيرات والثوابت والطرق والحزم والأصناف. في لغة Java هناك عدة قواعد لتسمية المعرفات, هي:

1. المعرف سلسلة من الحروف والأرقام والرموز الخاصة _ ,$.
2. لا يبدأ المعرف برقم وإنما بحرف أو $ أو _.
3. لا يكون كلمة محجوزة.

- لغة Java حساسة للفرق بين الحرف الكبير والحرف الصغير, حيث a≠A.

المتغيرات Variables

تستخدم المتغيرات لخزن البيانات. صيغة الإعلان عن متغير هي :

Data_type Varible_name;

Eg: int x;

double radius;

Or Data_type var_1, var_2,..., var_n.

Eg: char a,b,c;

تعليمة الإسناد: Assignment Statement

Var= expression;

Ex_1: int X;

X=1;

Ex_2: y=3*5+x;

الثابت Constant

يستخدم لخزن البيانات بصورة ثابتة لا تتغير.

final Data_type name= value;

Ex: final double pi= 3.14159;

أنواع البيانات الرقمية

توفر لغة Java ستة أنواع من البيانات الرقمية, هي:-

Name	Storage Size
1. byte	8-bit
2. short	16-bit
3. int	32-bit
4. long	64-bit

5.float	32-bit
6. double	64-bit

العمليات الحسابية

1. + : جمع
2. - : طرح
3. * : ضرب
4. / : قسمة
5. % : باقي القسمة

المتغيرات الرمزية

```
char  letter ='A';
String message="hello";
```

المتغيرات المنطقية

المتغير المنطقي يأخذ إما قيمة true أو false

Ex: boolean flag =true;

العمليات العلاقية

Operator	Name
<	less than
<=	less than equal
>	greater than
>=	greater than equal
==	equal to
!=	not equal

العمليات المنطقية

Operator	Name
!	not
&&	and
||	or

Ex: if (a>b)&&(C != 0) …..

عمليات الإسناد المتقدمة

Operator	Example	Equivalent
+=	i+=8	i=i+8
-=	i-=7	i=i-7
*=	i*=5	i=i*5
/=	i/=8	i=i/8
%=	i%=2	i=i%2

عوامل الزيادة والنقصان:

++i; : زيادة i بمقدار واحد ثم استخدام القيمة الجديدة في التعبير.

i++; : استخدام القيمة الحالية لـ i ثم الزيادة بمقدار واحد.

--i; : نقصان i بمقدار واحد ثم استخدام القيمة الجديدة في التعبير.

i--; : استخدام القيمة الحالية لـ i ثم النقصان بمقدار واحد.

```
Eg:  int i=4, y=0;
     y+= ++i;  //  y= 5,  i=5
     y+= i++; //   y= 4, i=5
```

القطع :Casting:

تحويل قيمة النوع البياني إلى قيمة بنوع آخر.

```
Eg: float f=10.1;
    int i= (int) f; // casting (i=10);
```

تحويل الخيط الرمزي إلى أرقام وبالعكس:

- لتحويل الخيط الرمزي إلى قيمة صحيحة, نستخدم:

```
int int_value=Integer.ParseInt(int_string);
Eg:  String num="19";
      int x=Integer.ParseInt(Num);   // x=19
```

- لتحويل الخيط الرمزي إلى نوع حقيقي, نستخدم:

```
double double_value=Double.ParseDpuble(double_string);
```

صناديق الحوارMessage Boxes

- هي وسيلة متقدمة للطباعة وإدخال البيانات.
- للتعامل مع صناديق الحوار في لغة Java, يجب استدعاء import الحزمة الخاصة بالصناديق وهي: import javax.swing.JOptionPane

حيث import:تعليمة استدعاء حزمة package.

Javax.swing: اسم الحزمة المستدعاة.

JOptionPane: اسم الصنف Class الخاص بصناديق الحوار داخل الحزمة أعلاه.

```
Eg: import javax.swing.JOptionPane;
    Public class welcome1 {
       Public static void main (String args[]){
          JOptionPane.showMessageDialog(null,"hellow\nJava");
          System.exit(0);
       }
    }
```

الصنف ← JOptionPane | الطريقة ↑ | المعلمات ↑ | إنهاء البرنامج ↑ System.exit(0)

- الطريقة showMessageDialog :تستخدم للطباعة على الشاشة

```
Eg: JOptionPane.showMessageDialog(null, "Hellow",
"Welcome",PLAIN_MESSAGE);
```

حيث, المعلمة الأولى: null

المعلمة الثانية: الشيء المراد طباعته.

المعلمة الثالثة: العنوان Title.

المعلمة الرابعة: الرمز, هناك عدة أنواع من الرموز:

ERROR_MESSAGE : رسالة الخطأ.

INFORMATION_MESSAGE : رسالة معلومات.

WARNING_MESSGE : رسالة تحذير.

QUSTION_MESAGE : رسالة سؤال.

PLAIN_MESAGE : رسالة بدون رمز.

- الطريقة showInputDialog : تستخدم للإدخال من لوحة المفاتيح.

Eg: String num=JOptionPane.showInputDialog("Please Input Value"); // num="10"

Int year=Integer.ParseInt(num); // year=10

Eg: Write Java program to input two numbers and add them.

Solution:

```
import javax.swing.JOptionPane;
 public class addition{
     public static void main (String args[]){
        String first,second;
        int num1,num2,num3;
        first =JOptionPane.showInputDialog("Enter First Number");
         second=JOptionPane.showInputDialog("Enter Second Number");
         num1=Integer.ParseInt(first);
         num2=Integer.ParseInt(second);
         num3=num1+num2;
JOptionPane.showmessageDialog(null,"The sum is "+num3, "Result", PLAIN_MESSAGE);
System.exit(0);
}  }
```

تعليمات الطباعة:

1. System.out.print( ); تطبع
2. System.out.println( ); تطبع وتنزل سطر

Eg: System.out.println("welcome to");
System.out.print("Java programming");

Or

JOptionPane.showMessage Dialog(null, "welcome to \nJava programming");

تعليمات السيطرة Control Statements:

1. تعليمة if

```
if (Boolean expression){
   statement(s);
}
```

```
Eg: if (number %2 ==0)
       System.out.println( number+"is even");
    If (number %2 !=0)
       System.out.println( number+"is odd");
```

2. تعليمة if…else

```
if (Boolean expression){
   statement(s);
} // for the true case
else{
    statement(s);
} // for the false case
```

```
Eg: if (number %2==0)
        System.out.println(number +" is even);
```

```
Else
   System.out.println(number+"is odd");
```

3. تعليمة if المتداخلة:

```
Eg: if (grade>=90)
      System.out.println("A");
    else if (grade>=80)
      System.out.println("B");
     else if (grade>=70)
      System.out.println("C");
     else if (grade>=60)
      System.out.println("D");
    else
      System.out.println("F");
```

ملاحظة هامة جدا: عند تنفيذ أكثر من تعليمة عند تحقق الشرط فإننا نستخدم الأقواس{}

```
Eq: if (A>B){
     System.out.println("A>B");
    C=A}
   else{
     System.out.println("B<=A");
      C=B;
   }
```

4. تعليمة switch

```
switch (expression){
  case value1 : statement(s)1 ;break;
  case value2 : statement(s)2 ;break;
  case valuen : statement(s)n ;break;
   default :statement(s);}
```

الشرط expression يجب أن يكون من النوع [short int, int, byte, char] والقيم $value_1, value_2,\ldots, value_n$ يجب أن تكون من نوع الشرط.

```
Eg: switch (month){
        case 1: name="jan"; break;
        case 2: name="feb"; break;


        default : name="invalid month";
      }
```

5. تعليمة while

```
while (condition){
   loop body;
}

Eg: int i=0;
     while(i<100){
        System.out.println("the no. is"+i);
         i++;
      }
```

6. تعليمة do…while

```
do{
    loop body;
}while (condition);
```

7. تعليمة for

```
for (i=1;i<=10;i++)
   System.out.println("the no. is"+i);
```

Methods الطرق

الطرق في Java هي نفسها الدوال functions في اللغات الأخرى.

- الطريقة: مجموعة من التعليمات التي تجمع مع بعضها البعض لإتمام مهمة معينة.

<u>توليد وخلق طريقة:</u>

في Java نتبع الصيغة التالية لإعلان طريقة:

```
return_value_type   method_name (parameters)
   {
    decarations and statements
   }
```

Ex1:
```
static int maximum (int num1, int num2)
      {
       if (num1>num2)
         return num1;
       else
          return num2;
       }
```

Ex2:
```
static int sequare(int i)
      {
       return (i*i);
       }
```

الدالتين أعلاه يمكن استدعائهم كالتالي:

```
M=maximize(x,y);
Z=sequare(5);
```

- لغة Java توفر مجموعة من الطرق الجاهزة ضمن أصناف وحزم خاصة. مثل الصنف Math الذي يحتوي على مجموعة من الطرق الرياضية.

الطريقة	وصف الطريقة	مثال
`abs(x)`	القيمة المطلقة لـ x.	`Math.abs(6.2)` → 6.2 `Math.abs(-2.4)` → 2.4
`ceil(x)`	تقرّب x إلى أقل عدد صحيح ليس اقل من x.	`Math.ceil(5.1)` → 6 `Math.ceil(-5.1)` → -5
`floor(x)`	تقرب x إلى أكبر عدد صحيح ليس أكبر من x.	`Math.floor(5.1)` → 5 `Math.floor(-5.1)` → -6
`max(x,y)`	أكبر قيمة من x و y.	`Math.max(7,6)` → 7
`min(x,y)`	أقل قيمة من x و y.	`Math.min(-7,-8)` → -8
`pow(x,y)`	x مرفوعة للأس y.	`Math.pow(6,2)` → 6^2 → 36
`sqrt(x)`	الجذر التربيعي لـ x.	`Math.sqrt(9)` → $\sqrt{9}$ → 3
`random()`	تكوّن رقم عشوائي بين الصفر والواحد.	`Math.random()` → 0.23121

The following class uses (read, sum, max, and print) methods to read 3 numbers, sum them, find the maximum, and print the results.

```
// Method1.java
1.  import javax.swing.JOptionPane;
2.  public class Method1{
3.   public static void main(String[ ] args) {
4.    int num1,num2,num3,sumOfnNmbers,maxValue;
5.    num1=read("input the first number");
6.    num2=read("input the second number");
7.    num3=read("input the third number");
8.    sumOfnNmbers=sum(num1,num2,num3);
9.    maxValue=max(num1,num2,num3);
10.   print ("the maximum number is ",maxValue);
11.   print ("the sum of  numbers is ",sumOfnNmbers);
12.  }       //end of main
13.
14.  // a method to read an integer value
15.  static int read(String m){
16.   int n = Integer.parseInt(JOptionPane.showInputDialog(m));
17.   return n;
18.  }       // end of read
19.
20.  // this method is to calculate the sum of three
21.  // integer numbers
22.  static int sum(int n1,int n2,int n3){
23.   int s;
24.   s=n1+n2+n3;
25.   return s;
26.  }       //end of sum
27.
28.  // this method returns the max of 3 integer numbers
29.  // using Math Class
30.  static int  max(int n1,int n2,int n3){
31.   int max;
32.   max=Math.max(Math.max(n1,n2),n3);
33.   return max;
34.  }       //end of max
35.
36.  // this methos prints out an integer value with a
37.  // message
38.  static void  print(String m,int n){
39.    JOptionPane.showMessageDialog(null ,m+"  "+n);
40.  }       //end of print
41. }
```

Write Java program to calculate the following formula.

$$y = 4\sqrt{5x} + 3z^2$$

```
// Math_Methods_1.java
1.  import javax.swing.JOptionPane;
2.  public class Math_Methods_1 {
3.   public static void main(String[] args) {
4.    double y,x,z;
5.    x=Integer.parseInt(
    JOptionPane.showInputDialog("Enter value for x"));
6.    z=Integer.parseInt(
    JOptionPane.showInputDialog("Enter value for z"));
7.    y=4 * Math.sqrt(5*x) + 3* Math.pow(z,2);
8.    JOptionPane.showMessageDialog(null,"y = "+y);
9.   }
10. }
```

Recursion التداخل

وهو عملية استدعاء الطريقة من قبل نفسها.

Ex: factorial function → 0!=1, n!= n*(n-1)*…*1.

Write Java Program, to compute the factorial of a number by using the recursion.

Solution:

```
import javax.swing.JOptionPane;
public class computefacturial {
 public static void main (String args[ ]) {
 String  st_num=JOptionPane.showInputDialog("Enter number" );
      int n=Integer.parseInt(st_num);
      JOptionPane.showMessageDilog(null, "the fact of " +n+" is
"+factorial(n), "Result", JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
 static long factorial(int n){
   if(n==0)
     return 1;
    else
     return n*factorial(n-1);
}
```

المصفوفات Arrays

- إعلان المصفوفة في لغة Java يمر بمرحلتين:

ا: الإعلان Declare: ويأخذ الصيغة التالية:

```
Data_type array_var [];
Ex: int c[];
```

ب: الخلق Create: حيث يتم خلق كائن المصفوفة للمتغير المعلن في الخطوة أعلاه, وذلك باستخدام التعليمة new. (كل كائن object في Java يخلق بتعليمة new).

```
array_var=new data_type [array_size];
Ex: c=new int[12];
```

وبذلك يصبح إعلان المصفوفة c من النوع int والحجم 12 :

```
int c[];
c= new int [12];
```

- يمكن جمع التعليمتين أعلاه بتعليمة واحدة: int c[]=new int [12];
- الدليل index في لغة Java يبدأ من 0 إلى c.length-1
- بعد خلق المصفوفة نقوم بإدخال عناصرها:

```
Ex: for (int i=0;i<c.length;i++)
        C[i]=i;
```

- يمكن إسناد قيم للمصفوفة لحظة الإعلان, كما في المثال التالي:

```
Ex: int c={10, 20, 30, 40, 50};
```

Write Java program, to read 6 numbers, and find the maximum.

Solution:

```
import javax.swing.JOptionPane;
 public class testarray {
   public static void main (String args[ ]) {
      int total=6;
      int list []=new int [total];
      for ( int i=0;i<list.length;i++) {
     String num=JOptionPane.showInputMessage("Enter number");
      List[i]=Integer.parseInt(num);
      }
     int max=list[0];
     for (int i=1;i<list.length;i++){
        if (max <list[i])
          max=list[i];
      }
  JOptionPane.showMessageDilog(null, "the max no. is " +max ,
 "Result", JOptionPane.PLAIN_MESSAGE);
 }}
```

Write Java program to search in the array.

```
// array1.java
1.  import javax.swing.*;
2.  public class array1{
3.   public static void main(String[] args) {
4.    int grades[] = new int[10];
5.    String names[] = new String[10];
6.    int ids[]=new int[10];
7.    String name, s;
8.    int c, no;
9.    String out;
10.   for(int i=0; i<grades.length; i++){
11.    names[i]=JOptionPane.showInputDialog("Enter student's name:");
12.    s=JOptionPane.showInputDialog("Enter his grade:");
13.    grades[i]=Integer.parseInt(s);
14.    ids[i]=i+100;
15.   }
16.   do{
17.    out="There is no student with this ID";
18.      s=JOptionPane.showInputDialog("Enter you choice:\n 1- Search for a
            student's name and grade using his ID.\n 2- Exit.");
19.    c=Integer.parseInt(s);
20.    switch(c){
21.      case 1:
22.       s=JOptionPane.showInputDialog("Enter the student's ID:");
23.       no=Integer.parseInt(s);
24.        for(int i=0; i<ids.length; i++)
25.         if(ids[i]==no)
26.           out="The student name is: " + names[i] + "\nThe student's grade is: " +
                                                          grades[i];
27.        JOptionPane.showMessageDialog(null, out, "Results",
                                              JOptionPane.PLAIN_MESSAGE);
28.            break;
29.      case 2: System.exit(0);
30.     }
31.   } while(c!=2);
32.   System.exit(0);
33.  }
34. }
```

تمرير المعلمات:

__ هناك طريقتان لتمرير المعلمات إلى الدوال في لغة Java:

1. تمرير القيمة pass-by- value (إرسال قيمة المتغير)وتطبق على الأنواع البيانية البسيطة مثل int, byte, double, float,..
2. تمرير العنوان pass-by-reference (إرسال العنوان) وتطبق على الأنواع المهيكلة مثلString, array .

المثال التالي يبين كيفية تمرير المعلمات إلى الدوال.

```
// Passing_Parameters.java

1. public class Passing_Parametres{
2.  public static void main(String args[]){
3.  int x;
4.  int a[] = {1, 2, 3, 4};
5.  x = a[1];
6.  System.out.println("The value of x before change is" + x) ;
7.  System.out.println("The value of a elements before change is: ");
8.  printArray(a);
9.  change(a, x);
10.    System.out.println("The value of x after change is" + x);
11.    System.out.println("The value of a elements after change is: ");
12.    printArray(a);


13. } //end of main
14.  static void change(int b[], int i){
15.       i *= 2 ;
16.    for (int index=0; index < b.length; index++)
17.    b[index]*= 2;
18. }       //end of method change

19.  static void printArray(int c[]){
20.    for (int index=0; index < c.length; index++)
21.    System.out.print(c[index] + "\t");
22.    System.out.println();
23.  }       //end of method print
24. }       //end of class Passing_Parametres
```

المصفوفات متعددة الأبعاد Multidimensional Arrays

- لإعلان مصفوفة ثنائية الأبعاد 4*3 نستخدم ما يلي:

```
int b[][];
b=new int [3][4];
Or  int b[][]=new int[3][4];
```

Write Java program to print the multiplication table using 2-D arrays.

```
// array2.java
1.  import javax.swing.*;
2.  public class array2{
3.    public static void main(String[] args) {
4.     int mult[][]=new int[7][7];
5.     for(int i=0; i<mult.length; i++)
6.      for(int j=0; j<mult[i].length; j++)
7.       mult[i][j]=(i+1)*(j+1);
8.     System.out.println();
9.     for(int i=0; i<mult.length; i++)
10.     System.out.print("\t"+(i+1));
11.   System.out.println();
12.   System.out.print("\t");
13.   for(int i=0; i<mult.length; i++)
14.    System.out.print("-------");
15.   System.out.println();
16.   for(int i=0; i<mult.length; i++){
17.    System.out.print("  "+(i+1)+"\t|");
18.    for(int j=0; j<mult[i].length; j++)
19.     System.out.print(mult[i][j]+"\t");
20.    System.out.println();
21.   }
22.   System.out.println();
23.   System.out.exit(0);
24.  }
25. }
```

classes الأصناف

البرمجة الكيانية الموجهة OOP تستخدم الأصناف classes لتضمين encapsulate البيانات والطرق.
مثال: جهاز الحاسوب يتضمن مجموعة من الماديات والبرمجيات التي يستخدمها المستخدم الاعتيادي بدون الحاجة لمعرفة التفاصيل الداخلية وهو ما يعرف بإخفاء المعلومات information hiding .

- الصنف class يمثل القالب العام للشيء المراد وصفه, أما الكائن object فهو نسخة من ذلك القالب.

Write class to implement the point.

```
public class point {
  private int x;
  private int y;
  public point() {}
```

```
 public point(int xv,int yv)
    {
    x=xv;
    y=yv;
    }
 public void setx(int xv)
   {x=xv;}
  public void sety(int yv)
  {y=yv;}
  public int getx()
   { return x;}
 public int gety()
   { return y;}
public String tostring(){
  return "[[ "+x+ ", "+y+" ]] ";}

}
```

ملاحظات

1. الصنف point يحتوي المتغيرات الصحيحة x, y وهي تعرف بالحقول field وهي معلنة خارج الطرق.
2. الكلمات public, private تعرف بموجهات الوصول access modifiers.
3. المتغيرات والطرق من النوع الخاص private يوصل لها فقط من قبل الطرق المعلنة داخل الصنف.
4. المتغيرات والطرق من النوع العام public يوصل من أي مكان في البرنامج حتى خارج الصنف.
5. قاعدة مفضلة: المتغيرات تعلن من النوع private والطرق تعلن من نوع public.
6. البنّاء constructer : دالة لها نفس اسم الصنف تستخدم لإعطاء القيم الابتدائية للصنف. وهو ليس فبه تعليمة return.

- بعد أن قمنا بتصميم الصنف point فإننا نقوم بإعلان وخلق كائنات objects من ذلك الصنف.
  - لإعلان الكائن point1 من الصنف point نكتب: point point1
  - ولخلق الكائن نستخدم التعليمة new: point1= new point( );

  ويمكن جمع الإعلان والخلق بتعليمة واحدة: point point1=new point( );
- الآن نقوم بتصميم صنف تطبيق يستخدم الصنفpoint وهذا التطبيق هو مثلا بالاسم pointtest

```
import javax.swing.JOptionPane;
public class pointtest{
 public static void main (String args[ ]){
   point point1=new point(72, 115);
   String output="x is "+point1.getx()+"\n y is "+point1.gety();
    point1.setx(10);
```

```
        point1.sety(20);
        output+="\n\n the new location is "+ point1.tostring();
        JOptionPane.showmessageDialog(null,output);
        System.exit(0);
      }}
```

الصنف point والصنف pointest يبدان بالكلمة public لذا يجب حفظ كل واحد بملف منفصل. لان لا يجوز حفظ اكثر من صنف من نوع public بنفس الملف.

Write Java class to implement the circle.

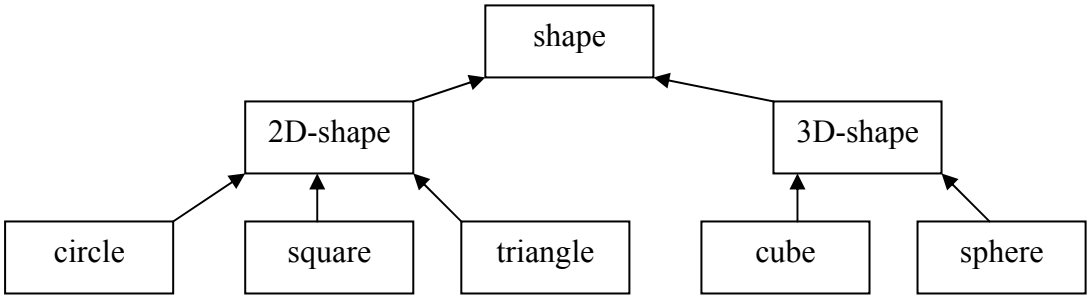
```
public class circle  {
   private double radius;
   private int x;
   private int y;
// no argument constructor
  public circle (){}
 //   Constructor
  public circle( int xv,int yv,double radiusv)
       {
        x=xv;
        y=yv;
        setradius(radiusv);
        }
      public void setx(int xv)
        {x=xv;}
        public void sety(int yv)
        {y=yv;}
       public int getx()
        { return x;}
      public int gety()
        { return y;}
      public void setradius( double r)
         {radius=(r <0.0? 0:r);}
      public double getradius()
        { return radius;}
      public double getdiameter()
        { return 2*radius;}
      public double getcircumference()
        { return Math.PI *getdiameter();}
      public double getarea()
       { return Math.PI * radius * radius}
     public String tostring()
     {      return "Center = [ "+x+" , "+y+" ] radius= "+radius;}
```

الصنف التالي يمثل تطبيق للصنف circle.

```
// test class circle
import javax.swing.JOptionPane;
import java.text.DecimalFormat;
public class circletest{
  public static void main (String args[]){
    circle circle1=new circle(37,43,2.5);
    String out="x is "+circle1.getx()+"\n y is "+circle1.gety()+"\n radius
is circle1.getradius();
     circle1.setx(35);
     circle1.sety(20);
     circle1.setradius(4.25);
    out+="\n\n the new location and radius of circle are
\n"+circle1.tosting() ;
     DecimalFormat  td=new DecimalFormat("0.00");
     out+="\n diameter is "+td.Fdormat(circle1.getdiameter());
     out+="\n area is "+td.Format(circle1.fetarea());
     JOptionPane.showmessageDialog(null,out);
     System.exit(0);
}
}
```

الوراثة Inheritance

- تعتبر الوراثة احد أهم خصائص البرمجة الكيانية وفيها يقوم الصنف بوراثة طرق methods ومتغيرات من أصناف أخرى.
- الصنف الموروث يعرف ب super class والصنف الوارف يدعى sub class.
- هرمية الصنف class hierarchy هي سلسلة الأصناف المتوارثة. مثل:



ملاحظة هامة: الطرق في الأصناف sub classes لا يمكنها الوصول للمتغيرات الخاصة private في الأصناف super classes.

المثال التالي يوضح كيف أن الصنف circle يمكن أن نصممه بحيث يرث عناصر members من الصنف point.

```
// circlenew.java
 public class circlenew extends point{
    private double radius;
    public circlenew(){}
    public circlenew(int xv,int yv,double radv)
      {super(xv,yv);
       setradius(radv);
      }
   public void setradius(double radv){
      radius=radv;}
  public double getradius(){
     return radius;}
  public double getdiameter(){
    return 2*getradius();}
  public double getcircumference(){
     return Math.PI*getdiameter();}
  public double getarea(){
     return Math.PI*radius*getradius();}
  public String tostring(){
     return "center= "+super.tostring ()+"; radius= "+getradius();
  }
 }// end class
```

- التعليمة extend تعني ان الصنف circlenew يرث من الصنف point (تعليمة الوراثة).
- التعليمة super تستخدم عندما نريد التعامل مع عناصر من نوع private في الصنف الموروثsuper class (point).
- الصنف circletest يبقى نفسه.