المصفوفات ARRAYS

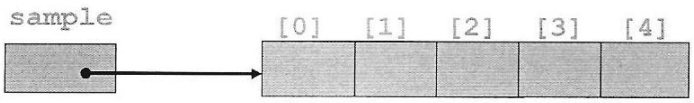
المصفوفات عبارة عن مجموعة من المتغيرات من نفس النوع يشار اليها باسم مشترك.
المصفوفات أحادية البعد تستخدم الشكل العام:

```
type array_name[ ] = new type[size];
```

مثال: `int sample[ ] = new int[5];`

وكما في الكيانات (objects) ، من الممكن كسر الاعلان السابق الى اثنين:

```
int sample[ ];
sample= new int[5];
```



ترقيم العناصر يبدأ من 0 ، لذا يكون العنصر الأول في sample هو sample[0] والعنصر الأخير هو sample[4] .

كما يمكن تعريف المصفوفة sample السابقة بالشكل:

```
int[ ] sample = new int[5];
```

أو

```
int[ ] sample;
sample= new int[5];
```

مثال progAr1 :

```
public class Ar1{
public static void main (String[] args){
  int i;
  double x[]=new double[5];
  for (i=0;i<=4;i++){
  x[i]= Math.sqrt(i);
  System.out.println("x["+i+"]:"+x[i]);
}}}
```

النتيجة:

```
x[0]:0.0
x[1]:1.0
x[2]:1.4142135623730951
x[3]:1.7320508075688772
x[4]:2.0
```

مثال progAr2:

```
public class Ar2{
public static void main (String[] args){
int i,sum=0;
int x[]={1,2,3,4,5,6,7,8,9};
for (i=0;i<=8;i++)
if (x[i]%2==0)
sum=sum+x[i];
System.out.println("sum="+sum);}}
```

النتيجة:

(مجموع عناصر المصفوفة الزوجية فقط)

```
sum=20
```

واجب :

عدلي البرنامج السابق ليقوم بحساب مجموع عناصر المصفوفة الفردية فقط وطباعة مجموعها.

مثال progAr3 :

```
public class Ar3{
public static void main (String[] args){
int i,min;
int x[]={7,9,25,5,2,8};
min= x[0];
for (i=0;i<=5;i++)
if (x[i]<min)
min=x[i];
System.out.println("min="+min);}}
```

النتيجة:

البرنامج يطبع العدد الأصغر min من بين الأعداد الصحيحة المخزنة في المصفوفة.

واجب:

عدلي البرنامج السابق ليقوم بطباعة العدد الأكبرmax من بين الأعداد الصحيحة المخزنة في المصفوفة.

مثال progAr3_2

```
public class Ar3_2{
public static void main (String[] args){
int i,min;
int x[]={25,7,9,5,2,8};
min= x[0];
for (i=0;i<=6;i++)
if (x[i]<min)
min=x[i];
System.out.println("min="+min);}}
```

النتيجة: (لايوجد خطأ عند ترجمة البرنامج، لكن يظهر الخطأ التالي عند تنفيذ البرنامج:

```
Exception in thread "main"
    java.lang.ArrayIndexOutOfBounds
    Exception: 6
```

وذلك بسبب تجاوز حدود المصفوفة.

اذاً تكون النتيجة هي:

البرنامج يعطي خطأ عند التنفيذ

المصفوفات ثنائية البعد:

مثال TowDim يخزن أعداد صحيحة في مصفوفة ثنائية مكونة من 4 صفوف و 5 أعمدة:

```
public class TowDim{
public static void main (String[] args){
int r,c;
int a[][]=new int[4][5];
for (r=0;r<4;r++){
for (c=0;c<5;c++){
a[r][c]=(r*5) + c +1;
System.out.print("   "+a[r][c]+"   ");}
System.out.println( );} } }
```

النتيجة:

```
1  2  3  4  5
6  7  8  9  10
11 12 13 14 15
16 17 18 19 20
```

الأعضاء الخاصة private members

في البرنامج prog54 اذا أضفنا كلمة private لعناصر الصف shape w,h (أي هكذا private double w,h;)تصبح w و h عناصر خاصة بالصف الفوقيshape فقط، ولا يمكن النفاذ أو الوصول اليها بواسطة أي طرق methods أخرى من خارج الصف shape الذي يحوي العناصر w,h .

عند تنفيذ البرنامج prog54 بعد تعديل السطر الثاني لـ:

```
private double w,h;
```

يفشل البرنامج في الترجمة وتظهر الأخطاء التالية عند ترجمته:

```
prog54.java:9: error: w has private access in shape  double area(){ double a; a=w*h/2; return a;}
                                                                                ^
prog54.java:9: error: h has private access in shape double area(){ double a; a=w*h/2; return a;}
                                                                                ^
prog54.java:16: error: w has private access in shape     t1.w=4.0;
                                                           ^
prog54.java:16: error: h has private access in shape     t1.h=4.0;
                                                           ^
prog54.java:17: error: w has private access in shape     t2.w=8.0;
                                                           ^
prog54.java:17: error: h has private access in shape     t2.h= 12.0;
                                                           ^
6 errors
```

الوصول الى الأعضاء الخاصة private members عن طريق (Accessor methods) طرق لها قابلية نفاذ :

```
class shape {
private double w,h;
double getw()  { return w;}
double geth() { return h;}
 void setw(double w1) { w=w1;}
void seth(double h1) { h=h1;}
void showDimo(){
System.out.println("width and
height  are: "  +w+" and "+h);
} // end method showDimo()
} // end shape class
```

```
class triangle extends shape {
    String style;
    double area(){
        return getw()*geth()/2;
    } // end method area()
    showStle(){
    System.out.println("The triangleis"
    +style) ; } // end method showStle()
    } // end of class triangle
```

```
class prog55{
public static void main( String[ ] args){
    triangle t1=new triangle();
    triangle t2=new triangle();
    t1.setw(4.0) ;    t1.seth( 4.0)  ;
t1.style= "isosceles";
    t2.setw(8.0) ;    t2.seth(12.0) ;
t2.style= "Right angle";
System.out.println("Tringle1 info:");
t1.showStle();        t1.showDimo();
System.out.println("Tringle1 area = "  +
t1.area());
System.out.println("Tringle2 info:");
 t2.showStle();       t2.showDimo();
System.out.println("Tringle2 area =
"+t2.area()); } }
```

النتيجة:

```
Tringle1 info:
The triangle is isosceles
Width and height are : 4.0 and 4.0
Tringle1 area = 8.0
Tringle2 info:
The triangle is Right angle
Width and height are : 8.0 and 12.0
Tringle2 area = 48.0
```

البانيات والوراثة:

يمكن أن يكون لكل من الصفين (الفوقي والفرعي) بانيته الخاصة بحيث يبني كل منهما الجزء الخاص به من الكيان ويمكن لكلا الصفين استخدام البانية الإفتراضية أو تعريف بانيته (غير الافتراضية):

: Prog57

```
class shape {
private double w,h;
double getw()  { return w;}
double geth() { return h;}
 void setw(double w1) { w=w1;}
void seth(double h1) { h=h1;}
void showDimo(){
System.out.println("width and height  are: "
 +w+" and "+h);
} // end method showDimo()
} // end shape class
```

```
class triangle extends shape {
  String style;
  triangle(String s,double w1,double h1 ) {
    setw(w1) ; seth(h1) ;  style=s ;
  } // end of constructor triangle()
  double area() {
        return getw()*geth()/2;
   } // end method area()
   showStle(){
   System.out.println("The triangleis"
   +style) ; } // end method showStle()
   } // end of class triangle
```

```
class prog57{
   public static void main( String[ ] args){
   triangle t1=new triangle("isosceles",4.0,4.0);
   triangle t2=new triangle("Right angle",8.0,12.0);
   System.out.println("Tringle1 info:");
   t1.showStle();  t1.showDimo();
   System.out.println("Tringle1 area = "+t1.area());
   System.out.println("Tringle2 info:");
   t2.showStle();  t2.showDimo();
  System.out.println("Tringle2 area = "+t2.area());
   } // end main() method
} // end class prog54
```

```
Tringle1 info:
The triangle is isosceles
Width and height are : 4.0 and 4.0
Tringle1 area = 8.0
Tringle2 info:
The triangle is Right angle
Width and height are : 8.0 and 12.0
Tringle2 area = 48.0
```