❤ Java ❤

المحاضرة الحادية والعشرون «الأخيرة» الاثنين 18/12/2017

تعددية الأشكال

تنبع تعددية الأشكال من أن الـobject يأخذ أكثر من شكل أي يتبدل بنوعه (أي يكون من الصف A أو من الصف B أو ... وهكذا).

هذا الكلام آتٍ من تعريف علاقة الوراثة التي هي: الابن هو أب أي الصف الابن هو صف أب وبالتالي طالما الابن هو أب أي الوارث هو مورث

ليكن لدينا:

```
class A {
    ....
    ....
}
class B extends A {
    ....
    ....
}
class C extends B {
    ....
    ....
}
```

أب — ابن — حفيد

C يرث B ظاهرياً ولكن فعلياً هو يرث A أيضاً

```
A a;
a = new A(); ✓
a = new B(); ✓
a = new C(); ✓
```

الفرض a أخذ أكثر من شكل أخذ شكل الصف A وشكل الصف B وشكل الصف C ومنها أتت تسمية تعددية الأشكال

* الـ B هو A وبالتالي استطيع بناء الفرض a باستخدام باني الصف B
* الـ C هو A وبالتالي استطيع بناء الفرض a باستخدام باني الصف C

ولكن العكس غير صحيح أي:

```
B b;
b = new A(); ✗
b = new C(); ✓
```

دائماً يمكن استخدام باني الابن لبناء object من نوع الأب

Scanned by CamScanner

* عند بناء الـ a مع باني الصف C فالنوع الفعلي له يكون C وليس A.
* عادة نعلم نوع المعمول من تعريفه أي إذا نظرنا إلى المتغير a فإنه معرف من الصف A ولكن عندما بنيناه من أحد الصفوف الأبناء فأصبح من نوع الصف الذي بُني منه

النوع الفعلي هو من نوع الباني الذي بُني به

لماذا تعددية الأشكال:
في كثير من الأحيان لا يكون لدينا معلومات كافية عن الـobject ولذلك نحتاج لتعددية الأشكال

مثال: برنامج يتيح للمستخدم أن يدخل مضلع ويقوم البرنامج بحساب محيطه

ليس لدينا علم بنوع المضلع

```
class Polynom{
   public String name;
      Polynom(String name){
        this.name = name;}
      void print(){
       System.out.print(name);}
                }
class Square extends Polynom{
    double l;
    Square(double l){
         this.l = l;}
     void print(){
      System.out.print(4*l);}
                }
class Tringle extends Polynom{
    double l;
    Tringle(double l){this.l = l;}
     void print(){
      System.out.print(3*l);}
}
```

(مثلث متساوي الأضلاع)

Scanned by CamScanner

داخل ال main : Polynom p; Strin d;

```
p=new Ploynom(d);
  p.print();
```

(سيطبع الاسم المخزن داخل ال d)

```
p = new Squaare(6);
  p.print();
```

(راح يطبع 24 أي 4*6)

```
p=new Tringle(2.3);
  p.print();
```

(راح يطبع 18.9 أي 3*2.3)

لو كتبنا: ضمن كتابة الهوز السابق:

```
class Polynom {
 public double length;
 public Polynom(){length = 0;}
 public void print(){
   System.out.println("I am a Polynom");}
 public double perimeter(){
  System.out.println(" cannot compute the preimeter");
      return -999;}
        }
 class Square extends Polynom{
  public Square(double a){length = a;}
  public void print(){
  System.out.println("I am a Square");}
   public double perimeter(){
      return 4* length;}
       }
```

Scanned by CamScanner

```
class Tringle extends Polynom{
 public Tringle (double a) { length = a ;}
 public void print(){
  System.out.println ("I am a Tringle ");;}
 pwblic double perimeter(){
     return  3 * length ;}
   }
```

: main في

```
  class Main{
   static Polynom  get_Polynom(){
System.out.println("Enter 1-Square, 2-Tringle, any-Polynom: ");
int x = Stdin.readInt();
 if-(x==1){
System.out.println("Enter length: ");
 double l= Stdin.readDouble();
    return  new Squre (l);
          }
  else if(x==2){
  System.out.println("Enter length: ");
  double l=Stdin.readDouble();
    return new Tringle(l);
          }
  else return  new Polynom();
                                  }
 pwblic static void main (String args[]){
    Polynom  p= get_Polynom();
     p.print();
    System.out.println(p.perimeter());
       } //end of main
       } //end of class
```

Scanned by CamScanner

شرح: الدالة get_Polynom وضعت لتتحدد من المستخدم ما المطلوب إنشاؤه (مربع أم مثلث أم مضلع آخر) وهذه هي فكرة تعددية الأشكال امكانية بناء كائن من الصنف الأب عن طريق بوابي الأبناء.

* قمنا بإنشاء دالة لها نوع إرجاع Polynom وبداخلها تم اختيار ما المطلوب إنشاؤه مربع أم مثلث أم مضلع.
* المفروض p من النوع Polynom أسندنا له القيمة المرجعة من الدالة get_Polynom وبالتالي نوعه الحقيقي هو إما مربع أو مثلث أو مضلع بحسب النوع المرجع من الدالة.
* عندما استدعائنا للدالة perimeter عن طريق p فإنه سيستدعي الدالة الموجودة ضمن نوعه (مربع - مثلث - مضلع) وبالمثل الدالة print

سؤال امتحاني:

لدينا الملف A.java الذي يحوي على البرنامج التالي:

```
class A {
 private int x,y;
 public A(int x, int y) {
   this.x = x;
   this.y=y;}
    }
public class B extends A{
   int 3;
 public B(int 3) {this.3=3;}
  int att_odd {int sum= x+y+3;
        return sum;}
}
```

عند ترجمة الملف السابق نحصل على 4 أخطاء حددها مع الشرح.

الحل: 1- اسم الملف خاطئ لأن اسم الملف هو اسم الصنف العام أي يجب أن يكون Bjava

2- عدم استدعاء بواني الأب (A) في أول تعليمة من بواني الابن (B)

3- x هو معطى بواني خاص للصنف A ومستخدم في الصنف B وهذا لا يمكن

4- y هو معطى بواني خاص للصنف A ومستخدم في الصنف B وهذا لا يمكن

ALAUID.net

Scanned by CamScanner

سؤال امتحاني : أكتب صفا يقوم بما يلي (نفرض n عدد طبيعي) :

1- البحث عن عنصر ما (حرف) في مصفوفة محارف من البعد n x n

2- ارجاع الكلمة المشكلة من محارف القطر الرئيس

3- طباعة الكلمة المشكلة من محارف القطر الثانوي

4- ايجاد منقول مصفوفة محارف من البعد n x n

5- طباعة مصفوفة محارف من البعد n x n

وباستخدام الصف السابق أكتب برنامج يقوم :

1- ادخال مصفوفة محارف من البعد m x m حيث $1 < m \in IN$

2- طباعة منقول هذه المصفوفة على شكل مصفوفة

3- طباعة الكلمة المشكلة من عناصر قطرها الثانوي

4- طباعة الكلمة المشكلة من عناصر قطرها الرئيس

الحل : اسم الملف : Matrix.java

```
class Matrix{
 static void input(char A[][]){
    for(int i=0;i<A.length;i++)
        for(int j=0;j<A[0].length;j++)
          A[i][j]=Stdin.readChar();
         }
 static boolean search(char A[][],char x){
    for(int i=0;i<A.length;i++)
      for(int j=0;j<A[0].length;j++)
        if(A[i][j]==x) return true;
    return false;
         }
  static String word(char A[][]){
     String S=" ";
    for(int i=0;i<A.length;i++)
       S=S+A[i][i];
      return S;
         }
```

Scanned by CamScanner

```
static void print_word(char A[][]){
    for(int i=0; i<A.length; i++)
     System.out.print(A[i][A.length-i-1]);
      }
static char [][] trans (char A[][]){
          char B[][] = new char [A[0].length][A.length];
    for(int i=0; i<B.length ; i++)
    for(int j=0; j<B[0].length; j++)
       B[i][j] = A[j][i];
  return B;
      }
  static void print (char A[][]){
      for(int i=0; i<A.length; i++){
        for(int j=0; j<A.length; j++)
      System.out.print(A[i][j]+"  ");
      System.out.println();
      } }
      } //end of class
```

الملف : Main.java

```
class Main{
public static void main(String args[]){
  int m;
do{ m=Stdin.readInt();} while (m<=1);
char A[][]=new char [m][m];
  Matrix.input(A);
Matrix.print(Matrix.trans(A));
Matrix.print_word(A);
 System.out.print(Matrix.word(A));
  } //end of main
  } //end of class
```

Scanned by CamScanner