مقرر: مقدمة فى البرمجة

البرمجة بلغة جافا

البانيات + الوراثة

البانيات Constructors :

- تمهد البانية كيان object عند إنشائه وتحمل بانية الصف نفس اسم الصف وهي مشابهة للنهج methods من ناحية التركيب النحوي. تستخدم البانيات بشكل نموذجي لإعطاء قيم ابتدائية للمتغيرات التي يعرفها الصف.
- لكل الصفوف بانيات سواء عرفتها أم لا، لان الجافا توفر تلقائياً بانية افتراضية تمهد كل متغيرات الاعضاء بصفر. لكن بعدما تعرف بانيتك الخاصة، لا تستخدم البانية الافتراضي

استخدام البانيات Constructors لإعطاء قيم ابتدائية للمتغيرات:

```
public class prog53{
public static void main(String[ ] args){
cartype car1 = new cartype(7,16,21);
cartype car2 = new cartype(2,14,12);
double gal; int distance =200;
gal= car1.fuelneeded(distance);
System.out.println("for distance:"+distance+" miles
  car1 needs:"+gal+"gallons of fuel.");
gal= car2.fuelneeded(distance);
System.out.println("for distance:"+distance+"miles
  car2 needs:"+gal+"gallons of fuel.");
} // end main()
```



```
class cartype{
int passengers;
int fuelcap;        int mpg;        int retvalue;
cartype(int p, int f, int m){
passengers = p;      fuelcap = f;      mpg = m;
} // end constructor()
int rang() { retvalue = fuelcap*mpg;
            return retvalue; } // end range()
double fuelneeded(int miles){
      return (double) miles/mpg; } // end fuelneeded ()
} // end class cartype
```

النتيجة:

```
for distance:200 miles car1 needs:9.52 gallons of fuel.
for distance:200 miles car2 needs:16.66 gallons of fuel.
```

الوراثة:

الوراثة هي من المفاهيم الأساسية للبرمجة الكائنية حيث يمكننا انشاء صف أساسي أو صف فوقي (supper class) يحتوي على بعض السمات أو الخصائص المشتركة التي يمكن توريثها للصفوف الفرعية (sup class) . وتستطيع الصفوف الفرعية إضافة سمات خاصة بها تميز كل صف فرعي عن الآخر، وتسهم الوراثة في توظيف مبدأ اعادة استخدام الكود (reuseability) مما يوفر الوقت والجهد ويسهم في تطوير البرمجة.

البرنامج prog54 :

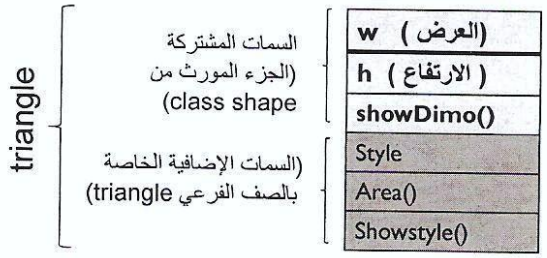
```
class shape{
  double w,h;
  void showDimo(){
       System.out.println("width="+w+"height="+h);
  }// end method showDimo()
}// end shape class
class triangle extends shape{
  String style;
  double area(){ double a; a=w*h/2;
  return a;} // end method area()
  void showStle(){ System.out.println("The triangleis"+style);
  } // end method showStle()
}//end of class triangle
```

```
class prog54{
  public static void main( String[ ] args){
      triangle t1=new triangle();
      triangle t2=new triangle();
      t1.w=4.0;    t1.h= 4.0;    t1.style= "isosceles";
      t2.w=8.0;    t2.h= 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());
  }// end main() method
}// end class prog54
```

4/14/2013

النتيجة:

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



8

4/14/2013