

الجلسة الثالثة - برمجة 3

الغاية من الجلسة: مقدمة عن مفهوم الوراثة في JAVA من خلال مثال بسيط

في البداية، سننشئ صفياً يسمى Person يحتوي على واصفات اسم و عمر و عنوان للشخص وطرقاً لتعيين والحصول على قيم هذه الواصفات، وأيضاً طرقاً لطباعة بيانات هذا الشخص ، ثم سنقوم باشتقاق صنف ابن Student (طالب) من صنف الاب (شخص) ، سيتم إضافة الواصفات والطرائق الخاصه بصنف الابن . ثم سنستخدم كلا الصنفين في الطريقة الرئيسية (main).

تعريف صنف الأب Person

```
class Person {  
  
    نعرف ثلاثة واصفات خاصة private تعبر عن الاسم و العمر و العنوان.  
  
    private String name;  
    private double age;  
    private String address;  
  
    ننشئ الباني الافتراضي و تاخذ فيه واصفة العمر قيماً صفرية أما واصفة الاسم والعنوان فتاخذ القيمة null أي لا شيء:  
    // Default Constructor  
    public Person(){  
    }  
  
    ننشئ الباني بوسطاء:  
    // Constructor  
    public Person (String name ,double age ,String address){  
        this.name = name;  
        this.age = age;  
        this.address = address;  
    }  
  
    ننشئ الطرق getters و setters التي تعطي وتعيد قيم للواصفات:  
    // Setters  
    public void setName(String name) {  
        this.name = name;  
    }  
    public void setAge(double age) {  
        this.age = age;  
    }  
    public void setAddress(String address) {  
        this.address = address;  
    }  
}
```

```
// Getters
public String getName(){
    return name;
}
public double getAge(){
    return age;
}
public String getAddress(){
    return address;
}
```

الطريقة الآتية من أجل طباعة بيانات الشخص :

```
public void printPersonDetails()
{
    System.out.println("Name = " + name + "\n Age " + age + "\n Address " + address);
}
```

(طريقة ثانية) يمكن استخدام الدالة toString من أجل طباعة بيانات الشخص بدلاً من الدالة السابقة

```
public String toString ()
{
    return ("Name = " + name + "\n Age " + age + "\n Address " + address);
}
```

تعريف الصف المشتق (الابن) Student حيث سيرث الصف Student كل الواصفات والطرائق الموجودة في الصف الاب وستقوم بإضافة واصفات وطرائق خاصة بالطالب .

```
class Student extends Person {
```

نعرف واصفتين خاصتين private أحدهما تعبر عن مستوى الدراسة والأخرى عن التخصص.

```
private int studyLevel;
private String specialization;
```

ننشئ الباني الافتراضي :

```
// Default Constructor
public Student(){
}
```

ننشئ الباني بوسطاء:

```
// Constructor
public Student (String name ,double age ,String address, int studyLevel ,String specialization)
{
    Super(name, age, address);
}
```

```
this.studyLevel = studyLevel;
this.specialization = specialization;
}
```

ننشئ الطرق setters التي تعطي قيم للواصفات:

```
// Setters
public void setStudyLevel (int studyLevel) {
    this.studyLevel = studyLevel;
}

public void setSpecialization (String specialization) {
    this.specialization = specialization;
}
```

ننشئ طرق ال getters التي تعيد قيم الواصفات:

```
// Getters
public int getStudyLevel (){
    return studyLevel;
}

public String getSpecialization (){
    return specialization;
}
```

الطريقة الآتية من أجل طباعة بيانات الطالب كاملة :

```
public void printStudentDetails()
{
    Super.printPersonDetails();
    System.out.println("\n studyLevel = " + studyLevel + "\n Specialization " + specialization);
}
```

(طريقة ثانية) يمكن استخدام الدالة toString من أجل طباعة بيانات الطالب بدلاً من الدالة السابقة السابقة printStudentDetails نلاحظ أن الدالة toString معرفة أيضاً عند الصف الاب ، يعرف هذا ب overriding حيث يقوم الصف الابن باعادة تعريف دالة موجوده في الصف الاب

```
public String toString ()
{
    Super.toString();
    return ("\n studyLevel = " + studyLevel + "\n Specialization " + specialization);
}
}
```

سنقوم بتعريف صف Test يحتوي على main لاختبار الصفوف المعرفة وعملية الوراثة :

```
public class Main {
```

```
public static void main(String[] args) {  
//    Create Person & Student  
    Person p1 = new Person();  
    Person p2 = new Person (" Ahmad" ,15, "lattakia");  
    Student s1 = new Student ();  
    Student s2 = new Student (" Sara" ,25, "lattakia" ,2 , "informatics Engineer");  
  
//    Print dimensions  
P2.printPersonDetails();  
S2.printStudentDetails();  
  
//    Print dimensions  
System.out.println(P2);  
System.out.println(S2);  
}}
```

وهنا في السطرين التاليين نطبع معلومات الشخص الثاني ونطبع معلومات الطالب الثاني:

(طريقة ثانية) يمكن الطباعة من خلال toString: