



Abstract Class

الصف المجرد

Dr. REEMA AL-KAMHA

مقدمة

- يمتلك مفهوم الصف المجرد أهمية خاصة لأنه **يوفر تقنية فعالة لتعريف صفوف عامة الأغراض** يمكن تخصيصها فيما بعد باستخدام مفهوم الوراثة .
- وبكلمات أبسط **يعزز الصف المجرد مفهوم إعادة الاستخدام**، وذلك بكتابة صف يحتوي على الخصائص العامة التي تشترك فيها مجموعة من المسائل وترك التفاصيل الخاصة بكل مسألة لتعريفها من خلال الصفوف التي ترث هذا الصف المجرد.

التصريح عن الصف المجرد

- يمكن التصريح عن الصف المجرد باستخدام الكلمة المفتاحية `abstract`.
- على سبيل المثال، التصريح التالي يعرف صف مجرد `Shape`

```
abstract class Shape{  
    int x;  
}
```

الطرق الموجودة في الصف المجرد

يمكن للصف المجرد أن يحتوي:

❖ طرق معرفة بشكل كامل أو ما يسمى concrete methods

❖ طرق مجردة abstract methods.

- تتألف الطريقة المجردة فقط من ترويسة الطريقة ولا تملك أي تعليمات برمجية، و في هذه الحالة يجب أن تسبق الكلمة المفتاحية abstract اسم الطريقة.

مثال عن الصف المجرد

- يبين المثال التالي صف مجرد يصرح عن طريقة مجردة `getArea` و طريقة معرفة بشكل كامل `setName`

```
abstract class Shape{  
    string Name;  
    abstract int getArea();  
    void setName(String n){  
        Name=n;  
    }  
}
```

طريقة مجردة

طريقة غير مجردة

كما تلاحظ:

- تعرف الطريقة المجردة فقط بالترويسة و يسبق اسمها كلمة `abstract` و لا تملك تعليمات برمجية.
- أما الطريقة غير المجردة فتحتوي ترويسة إضافة إلى التعليمات البرمجية.

ملاحظات حول الصف المجرد

- الصف العادي لا يمكنه أن يحتوي طرق مجردة نوعها abstract.
- الصف المجرد يحوي طرق عادية و طرق مجردة.
- يمكن للصف المجرد أن يحتوي على حقول بمختلف أنواعها.
- لا يمكن إنشاء أغراض من الصف المجرد، لأن الغاية من هذا النوع من الصفوف هو توفير صف عام يمكن تخصيصه لحل مسائل في مجال معين.
- يجب تخصيص الصف المجرد من خلال مفهوم الوراثة قبل استخدامه أي يجب أن تتم وراثته، و يجب على الصف الابن (الذي يخصص الصف المجرد) كتابة تعليمات تنفيذية لكل طريقة مجردة معرفة في الصف المجرد (يؤدي خلاف ذلك إلى خطأ ترجمة).
- الصف الابن الذي يرث من الصف المجرد يجب أن يعيد كتابة كل الطرق المجردة الموجودة في الصف المجرد.

ملاحظات حول الطريقة المجردة

- لا تحتوي على جسم و يجب وضع فاصلة منقوطة بدل أقواس البداية و النهاية.
- الصف الذي يرث من صف نوعه `abstract` يجب أن يفعل `override` لكل طريقة مجردة، أي يجب كتابة جسم الطريقة.
- الطريقة المجردة لا تكون من النوع `private` لأنه لا يمكن إعادة تعريف (`override`) الطريقة الخاصة.
- وجدت الطريقة المجردة لعمل `override` لها في صف الابن.
- لا يشترط وجود طريقة مجردة في الصف المجرد، و لكن وجودها يجبر المبرمج على جعل الصف مجرد.

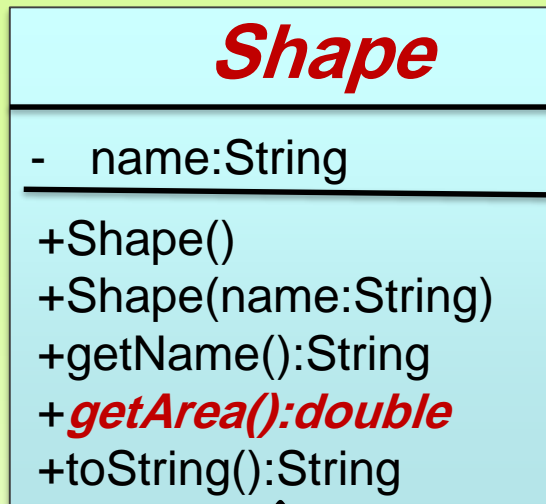
مثال على الصف المجرد في الجافا

اكتب برنامجا بلغة الجافا للتوصيف التالي:

- **تعريف صف مجرد اسمه Shape** ويمثل نوعا عاما للأشكال الهندسية. له اسم، وباني افتراضي (خالي)، إضافة لباني مناسب للتوصيف. كما يعرف الطريقة `getName` والطريقة `toString` والتي ترجع العبارة التالية: "the name of the shape". يعرف أيضا طريقة `getArea` مجردة.
- **تعريف صف Rectangle** علما أنه شكل هندسي، له طول و عرض. كما يحوي باني مناسب للتوصيف، إضافة للطريقة `toString` و التي ترجع العبارة التي ترجعها الطريقة `toString` في الصف المجرد، إضافة لاسم الشكل و الطول و العرض و المساحة.
- **تعريف صف Triangle** علما أنه شكل هندسي، له قاعدة و ارتفاع. كما يحوي باني مناسب للتوصيف، إضافة للطريقة `toString` و التي ترجع العبارة التي ترجعها الطريقة `toString` في الصف المجرد، إضافة لاسم الشكل و القاعدة و الارتفاع و المساحة.
- **تعريف صف AbstractShapeApp** يقوم بإنشاء أغراض من الصفوف السابقة و استدعاء الطرق الموجودة فيها.
- **ملاحظة:** يحوي الصف `Shape` على طريقة مجردة `getArea` لحساب مساحة الشكل الهندسي. بما أن طريقة حساب المساحة تختلف باختلاف الشكل الهندسي، فإن كل شكل هندسي كصف المستطيل و المثلث يرث صفات الصف `Shape` و يعرف التعليمات البرمجية التي تحسب مساحة الشكل المحدد.

تمثيل الصف المجرد باستخدام لغة UML

- اسم الصف المجرد مكتوب بخط مائل
- الطريقة مجردة مكتوبة بخط مائل



Rectangle

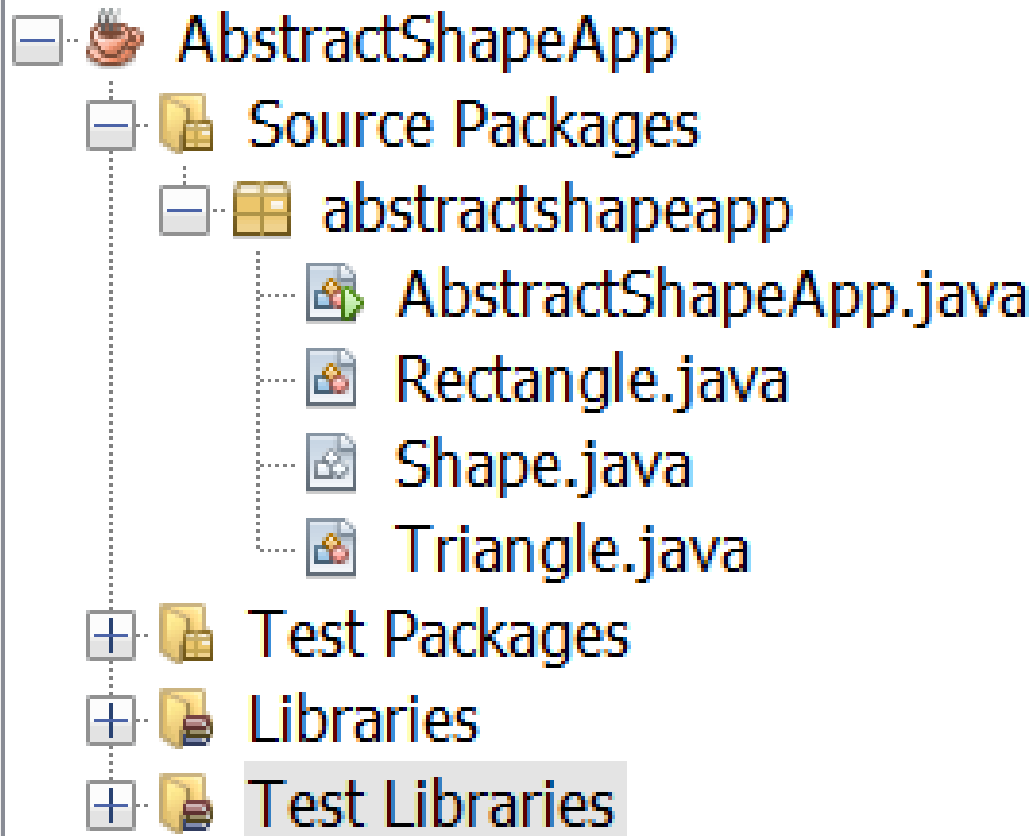
- length:double
- width: double

+Rectangle(name:String, length:double, width:double)
+toString():String

Triangle

- base:double
- height: double

+Triangle(name:String, base:double, height:double)
+toString():String



الصف المجرد Shape

```
package abstractshapeapp;
public abstract class Shape {
    private String name;
    public Shape () {
        name="";
    }
    public Shape (String name) {
        this.name=name;
    }
    public String getName () {
        return name;
    }
    public abstract double getArea ();
    @Override
    public String toString () {
        return "the name of the shape is ";
    }
}
```

استخدام الكلمة abstract للدلالة على أن الصف Shape مجرد

طريقة مجردة، سيتم إعادة تعريفها في الصفوف الأبناء

الصف الابن Rectangle

```
package abstractshapeapp;
public class Rectangle extends Shape {
    double length;
    double width;
    Rectangle(String name, double length, double width) {
        super(name);
        this.length=length;
        this.width=width;
    }
    @Override
    public double getArea() {
        return length*width;
    }
    @Override
    public String toString() {
        return super.toString()+getName()+" Length="+length+
            " width="+width+ " The area = "+getArea();
    }
}
```

الصف Rectangle يرث من الصف Shape

إعادة تعريف الطريقة المجردة getArea() الموجودة في الصف المجرد ضمن الصف الابن

الصف الابن Triangle

```
package abstractshapeapp;
public class Triangle extends Shape{
    private double base;
    private double height;
    Triangle(String name, double base, double height){
        super(name);
        this.base=base;
        this.height=height;
    }
    @Override
    public double getArea(){
        return 0.5*base*height;
    }
    @Override
    public String toString(){
        return super.toString()+getName()+" base=" +base+
            " height="+height+" The area = "+getArea();
    }
}
```

الصف Triangle يرث من الصف Shape

إعادة تعريف الطريقة المجردة
getArea() الموجودة في الصف المجرد
ضمن الصف الابن

الصف الرئيسي AbstractShapeApp على الشكل التالي:

```
package abstractshapeapp;
import java.util.*;
public class AbstractShapeApp {
    public static void main(String[] args) {
        Rectangle r=new Rectangle("rectangle",5,3);
        System.out.println(r.toString());
        Triangle t=new Triangle("triangle",10,5);
        System.out.println(t.toString());
    }
}
```

يمكن كتابة الصف الرئيسي **AbstractShapeApp** على الشكل التالي:

```
package abstractshapeapp;
public class AbstractShapeApp {
    public static void main(String[] args) {
        Shape r=new Rectangle("rectangle",5,3);
        System.out.println(r.toString());
        r=new Triangle("triangle",10,5);
        System.out.println(r.toString());
    }
}
```

تعريف متغير مرجعي من الصف المجرد Shape، و جعله يُوْشر على غرض من الصف Rectangle

المتغير المرجعي من الصف المجرد Shape يُوْشر على غرض من الصف Triangle

نتيجة التنفيذ

the name of the shape is rectangle Length=5.0 width=3.0 The area = 15.0

the name of the shape is triangle base= 10.0 height=5.0 The area = 25.0

ملاحظات

- ❑ لا يجوز وضع كلمة `abstract` قبل الطريقة التي قبلها كلمة `.static`.
- ❑ لا يجوز وضع كلمة `abstract` قبل الباني.
- ❑ لا يجوز وضع كلمة `abstract` قبل الطريقة التي تكون بشكلها النهائي أي `.final`.

تذكرة

- الصف المجرد هو صف قد يحتوي طرقا غير مكتملة (فقط ترويسة دون تعليمات و دون جسم الطريقة).
- توفر القدرة على تعريف طرق غير مكتملة إمكانية تعريف صف عام يمكن تخصيصه باستخدام مفهوم الوراثة.