

شرح الكلاسات ال concrete,abstract,interface

مقالة

28 ي

r.k096igvjt

السلام عليكم ورحمة الله

ارجو مراجعة هذا الموضوع عن شرح الدوال أن كنت لا تعرف أجزائها

في البداية ساشرح أنواع الكلاسات الثلاث ال concrete و ال abstract و ال interface سابدأ من ال concrete لأننا نستخدمه دائما

لكن لاوضح امر هام الفرق بين ال concrete method وبين ال abstract method

ال **concrete method** : هي الدالة او ال method التي تحتوي على body

مثال

```
public int myMethod(){
//this is the body هذا هو البيدي
return 5;
}
```

ال **abstract method** : هي الدالة أو ال method التي لا تحتوي على body

مثال

```
public abstract int myAbstractMehtod();
```

لاحظ كيف عرفنا ال abstract method بعد كلمة public وضعنا كلمة abstract وكيف لم نكتب body

الان لندخل في الموضوع الهم أنواع الكلاسات

أولاً ال Concrete class

هو عبارة عن كلاس جاهز تماماً كل الدوال ال method فيه تحتوي على body أي (concrete method)

مثال دائماً نستخدمه عن ال concrete class هو **AppCompatActivity** لذلك لكي نستدعي أي دالة من هذا الكلاس يجب علينا ان نضع

@override

قبلها

مثال دالة onCreate الغنية عن التعريف

```

1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main2);
11    }
12 }
13
14

```

ثانياً ال Abstract class

هو عبارة عن كلاس شبه جاهز أي أن جزء من الدوال التي فيه تحتوي على body أي (concrete method) و الجزء الآخر لا يحتوي على body أي (abstract method) مثال على ذلك **view group**

في حال ورث كلاس من abstract class يجب أن يستدعي الدوال التي ليس لها body عن طريق **@Override** و أن نقوم بكتابة body لهذه الدوال

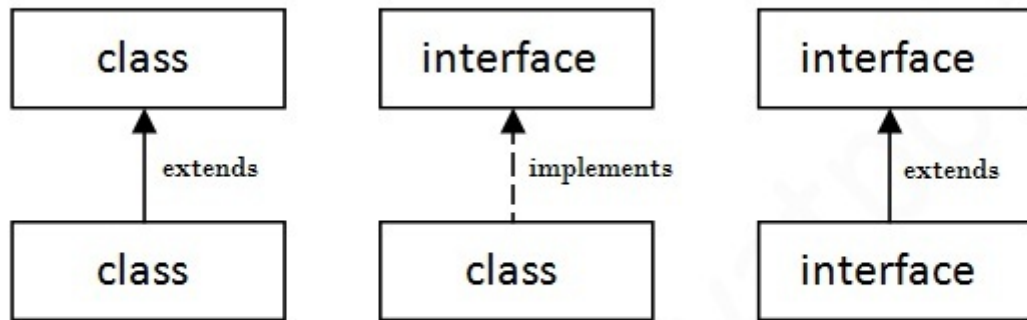
ثالثاً ال Interface

هو عبارة عن كلاس جميع الدوال التي في داخله لا تحتوي على body أي (abstract method) مثال عليه ال **onClicklistener** هو interface وهو هيملك دالة واحدة وهي **onClick(View v)** وهي الدالة التي نستدعيها دائماً عندما نستدعي ال **onClicklistener** ونكتب ال body لها

```

1 public class MainActivity extends AppCompatActivity {
2
3     private Button myButton;
4
5     @Override
6     protected void onCreate(Bundle savedInstanceState) {
7         super.onCreate(savedInstanceState);
8         setContentView(R.layout.activity_main2);
9
10        myButton = (Button) findViewById(R.id.my_button);
11
12        myButton.setOnClickListener(new View.OnClickListener() {
13            @Override
14            public void onClick (View v) {
15
16            }
17        });
18    }
19 }

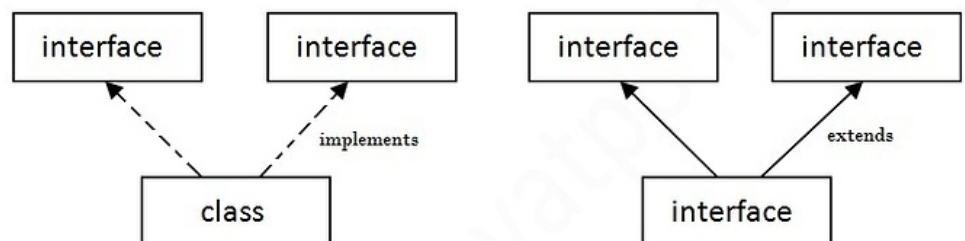
```

الان الفرق بين **extend** و **implement**

الكلاس ال concrete يستطيع أن يرث extend من كلاس concrete
 الكلاس ال concrete يستطيع أن يرث extend من كلاس abstract
 الكلاس ال concrete يستطيع ينفذ implement من interface

الكلاس ال abstract يستطيع أن يرث extend من كلاس abstract
 الكلاس ال abstract يستطيع أن يرث extend من كلاس concrete
 الكلاس ال abstract يستطيع أن ينفذ implement من interface

interface ال يستطيع أن يرث extend من كلاس interface
 interface ال يستطيع أن ينفذ implement من interface



Multiple Inheritance in Java

الكلاس ال concrete يستطيع أن يرث extend و أن ينفذ في أن واحد
 interface ال يستطيع أن ينفذ اثنان من ال interface في أن واحد
 interface ال يستطيع أن يرث extend اثنان من ال interface في أن واحد

عدى ذلك لا يمكن

مصدر الصورتين

بالتوفيق

ماهو الفرق بين **extends** و **implements**

شو الفرق بينهم الانواع

الفرق بين **Concert class** & **interface** & **abstract class**

الفرق بين **concrete class** و **abstract class** و **interface**

ال **Classes** في الدرس ال 12