

[Inheritance]

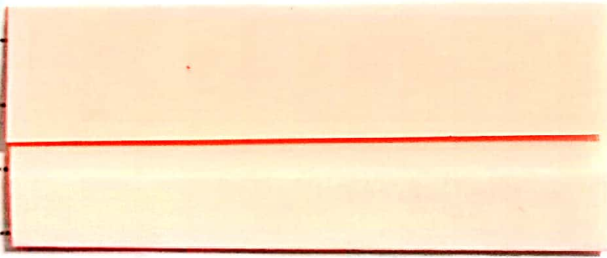
called: derived, derived class
subclass

you can define specialized classes that extend the
generalized class → called: parent, base class
superclass

↳ *Inheriting the properties and methods from the general class.

*Adding new properties and methods.

sub class has private data fields setters and getters



What about inherited data fields, how do they get initialized?

Using `super();` keyword

`super();` → calling superclass constructor
" " method

✓ constructors of superclass are not inherited by the subclass

→ it's not allowed in Java for a subclass to be extending more than one superclass in which is called multiple inheritance directly

(But this can be achieved by interfaces in Java)

[Note]

the statement super() ; must be the first statement appearing in the subclass constructor

[Constructor chaining]

constructor لا subclass لو كان ال constructor implicitly will add no-arg ← compiler

implicitly super() ; ← compiler

لا يوجد دائماً في ال constructors لكن بين ال constructors وال superclass بالترتيب من ال highest-superclass constructor

→ a constructor may invoke an overloaded constructor (using this) or its super class constructor, but if neither is invoked [both if exist should be the first statement appearing], the compiler automatically puts super(); as the first statement in the constructor.

→ For a superclass, you should provide a no-arg constructor to avoid programming errors.

[Dealing with methods] in Inheritance relationship

[Regarding to visibility]

→ Public methods of superclass

← بشكل عام ، أي inherited method ، يجب أن نأخذها من subclass
بأنها عادية بدون كلمة (super)

← لكن إذا كان في subclass ← method ← مكتوبة مطابقة
الكتابة من ال signature و return type و
(implementation و يكون modified)
الزائدي (دون استخدام ال (super) ✓ إذا كتبت في ال
superclass method

→ Private superclass methods

→ a private method cannot be overridden, because it is not accessible outside its own class

→ if a method defined in a subclass is private in its superclass, the two methods are completely unrelated

Regarding to it's static or instance

→ Instance

public private و public ال

→ Static

→ a static method cannot be overridden

But If a method defined in the superclass is redefined in a subclass, the method defined in the superclass is hidden

the it can be invoked this way:-

SuperclassName.StaticMethodName;

Overriding

~~→ multiple methods with~~

→ providing a new implementation for a method in subclass (same signature, same return type)

Overloading

→ multiple methods with the same name but different signature

parameter list compatibility ←

most specific matching

Override Annotation

→ it's not an instruction that can be converted into byte code

→ message to JVM

→ If the method with this annotation does not override its superclass's method, the compiler will report an error

[Object class]

→ Any class defined, will be by default extending object class, directly, or indirectly.

The method toString(); [public String toString()]

→ will be inherited to every single class defined in Java.

وظيفة الأصلية أنه ترجع وصف الكلاس التي ينتمي اليه
 ال object وال address
 نتج ال object

كلاي الميرود الكلاص وحيثما كان (class)

① ← تغيرنا نأخذها على اسم ال object سواء
 جملة طابعة اون لاين استقال
 sout (object.toString());

② ← اذا حطينا اسم ال object طابا
 طابعة طابعة، بدوننا كاي طابا الميرود
 sout (object);

overriden ← method لاننا نأخذها ال reference لانها وظيفتها الأصلية

✓ [Note] Usually you should override the toString method so that it returns more informative string and [descriptive] (of the object)

objects, Arrays, reference دلائل ال سواء

reference implicitly is invoking its own toString() method.