

Patrick GIRARD
University of Poitiers
2009-2014©

Object-Oriented Programming in Java™

Advanced Inheritance



Chapter 6 – Section 5



Table of contents

- ☉ Summary of inheritance characteristics
- ☉ Polymorphism
- ☉ Abstract classes
- ☉ Interfaces
- ☉ Object inspection, the great strength of objects



The Object class

- Root of object hierarchy in Java
- Offers very interesting methods
 - equals()
 - clone()
 - toString()
 - finalize()
 - hashCode(), notify(), notifyAll(), wait()
 - getClass()



The Object class: equals()

```
public boolean equals(Object obj)
```

☉ The operator == compares only...
references

☉ returns true only if the object is exactly the same
in memory

☉ To check values, equals(), which is a
method of Object, can be used

☉ We then can compare what we want...

☉ ... by overriding it!



The Object class: equals()

```
public static void main(String args[]){  
    Animal a1 = new Animal ("Medor", 2002);  
    Animal a2 = new Animal ("Medor", 2002);  
    Animal a3 = a1;  
    System.out.println ("a1 == a2 : " + (a1 == a2));  
    System.out.println ("a1 == a3 : " + (a1 == a3));  
    System.out.println ("a1.equals(a2) : " + a1.equals(a2));  
    System.out.println ("a1.equals(a3) : " + a1.equals(a3));  
}
```

```
a1 == a2 : false  
a1 == a3 : true  
a1.equals(a2) : false  
a1.equals(a3) : true
```



The Object class: equals()

```
public boolean equals(Object obj)
```

```
public boolean equals(Animal obj) {  
    return this.Nom.equals(obj.Nom) &&  
        this.anneeNaissance == obj.anneeNaissance ;  
}
```

```
a1 == a2 : false  
a1 == a3 : true  
a1.equals(a2) : true  
a1.equals(a3) : true
```



The Object class: clone()

```
protected Object clone()  
                throws CloneNotSupportedException
```

```
public class Animal implements Cloneable {  
    ...  
    public Object clone() {  
        return new Animal (Nom, anneeNaissance);  
    }  
}
```

```
public static void main(String args[]){  
    Animal a1 = new Animal ("Medor", 2002);  
    Animal a2 = (Animal) a1.clone();  
    Animal a3 = a1;  
    System.out.println ("a1 == a2 : " + (a1 == a2));  
    System.out.println ("a1 == a3 : " + (a1 == a3));  
    System.out.println ("a1.equals(a2) : " + a1.equals(a2));  
    System.out.println ("a1.equals(a3) : " + a1.equals(a3));  
}
```

```
a1 == a2 : false  
a1 == a3 : true  
a1.equals(a2) : true  
a1.equals(a3) : true
```



The Object class: toString()

```
public String toString()
```

```
public static void main(String args[]){  
    Animal a1 = new Animal ("Medor", 2002);  
    System.out.println (a1)  
} equivalent to a1.toString()
```

Animal@2e7263



```
public String toString() {  
    return Nom + " (" + anneeNaissance + ")";  
}
```

Medor (2002)



The Object class: finalize

- Method called by the garbage collector when it erases the last reference on the object
- Overriding it allows to make something before to suppress the object
 - Example: count the number of current objects

```
protected void finalize()  
                throws Throwable
```



The Object class:

hashCode(), notify(), notifyAll(), wait()

• hashCode()

- Calculates a hash code for the object

• notify(), notifyAll(), wait()

- Thread management



The Object class: introspection

```
public class Inspecteur {
    private Object obj;
    public Inspecteur (Object o) { obj = o; }

    public String classesMeres () {
        return classesMeres(obj.getClass());
    }
    private String classesMeres (Class uneClasse) {
        Class classeMere;
        String retour;
        if ((classeMere = uneClasse.getSuperclass()) == null)
            retour = Object.class.getName() + "\n";
        else {
            retour = uneClasse.getName() + "\n";
            retour += classesMeres (classeMere);
        }
        return retour;
    }
}
```



The Object class: introspection

```
...  
    public static void main (String argv[]) {  
        Loup l = new Loup("Wolff", 2000);  
        Inspecteur insp = new Inspecteur (l);  
        System.out.println (insp.classesMeres());  
    }  
}
```

Loup
Therien
Mammifere
Animal
java.lang.Object



The Object class: introspection

the class Class

```
public boolean isInstance(Object obj)
public String getName()
public Class getSuperclass()
public Class[] getInterfaces()
public Method getMethod(String name,
                        Class[] parameterTypes)
                        throws NoSuchMethodException,
                            SecurityException
public Field[] getFields()
                        throws SecurityException
```



Table of contents

- ☉ Summary of inheritance characteristics
- ☉ Polymorphism
- ☉ Abstract classes
- ☉ Interfaces
- ☉ Object inspection

the end