



Orienté objet, héritage classes abstraites et interfaces

Année académique 2019-2020

Prof Pierre Manneback

Ir Olivier DEBAUCHE

En Java

Le mot clé **final** permet de rendre constant un élément comme une variable, une fonction, un tableau, etc.

En c++: `const int a = 3;`

En java: **final** `int a = 3;`

Un **package** permet de regrouper un ensemble de classes et d'éviter des conflits entre classes qui porteraient le même nom.

package `examples.aPackage;`

En C++

Héritage de classe

```
class Cercle : public Forme {  
  
};
```

```
class Cercle : public Forme {  
    Cercle(int xx, int yy)  
    {  
        Forme(xx,yy);  
    }  
};
```

Constructeur de
la classe Cercle

Appel du
constructeur de
Forme dans la
classe cercle

Forme

Cercle

En Java

Héritage de classe

```
class Cercle extends Forme {  
  
}
```

```
class Cercle extends Forme {  
    public Cercle(int xx, int yy)  
    {  
        super(xx,yy);  
    }  
}
```

Constructeur de
la classe Cercle

Appel du
constructeur de
Forme dans la
classe cercle

Forme

Cercle

En Java

Sécurité

- Méthode **private** accessible uniquement de la classe elle-même.
- Méthode **public** est accessible par toutes les autres classes.
- Méthode **protected** sont accessibles par les classes filles

En Java

Constructeur

Exemple:

```
class MyClass {  
    private int data1;  
    private String data2;  
  
    public MyClass() { /* ... */ }  
}
```

Un constructeur
n'a pas de type
est **public** et ne
renvoi rien

En Java

On peut redéfinir une méthode de la classe mère dans une classe fille en la faisant précéder par **@Override**

Exemple:

@Override

```
public int getField1() { return super.getField1() * 3; };
```

En Java

Pas d'héritage multiple en Java

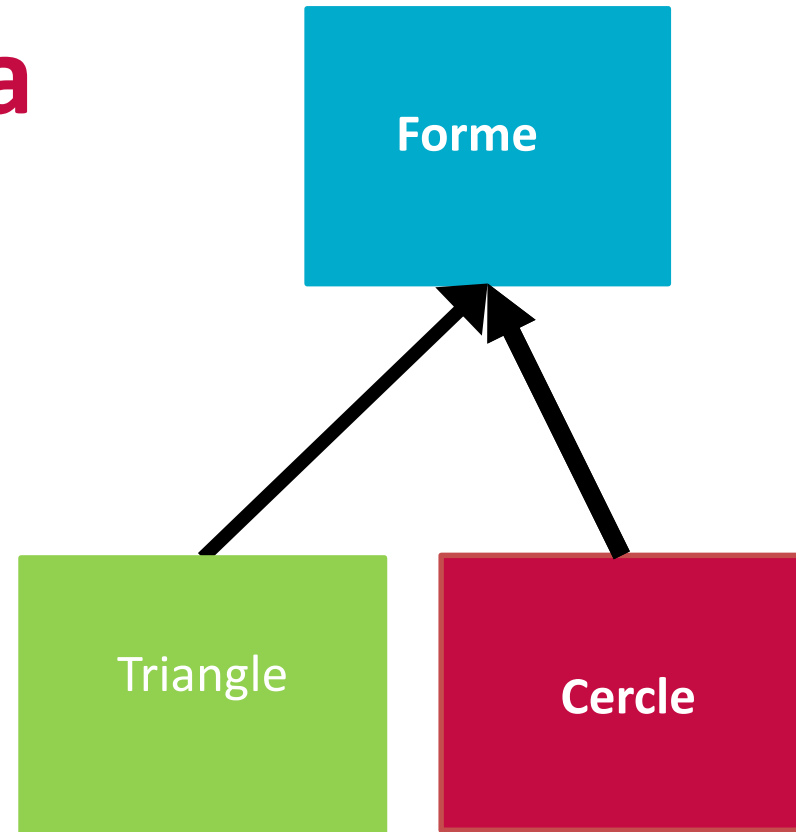
Moyen de s'en sortir

la class abstraite qui n'est pas
Instanciable mais héritable par

les autres classes

Exemple:

```
public abstract class Forme{  
    public type methode1();  
    public type methode2();  
}
```



En Java

Interface

```
Interface calculGeometrique {  
    default public type methode1() {...}  
    default public type methode2() {...}  
}
```

calculGeometrique

```
public type methode1();  
public type methode2();
```

Forme

Cercle

```
class Forme implements calculGeometrique {  
    public type methode1()  
    {  
    }  
    public type methode2()  
    {  
    }  
}
```

Redéfinition des
méthodes de
l'interface