Université Nice Sophia Antipolis

Classes internes

F. Mallet

miage.m1@gmail.com

http://deptinfo.unice.fr/~fmallet/

Université Nice Sophia Antipolis

Classes internes

- Définition
  - Dans une classe
  - Dans une méthode, entre { }
  - En paramètre (anonyme)
- Observable
  - **`ClassePrincipe.class`**
  - **`ClassePrincipe$1NomInterne.class`**
  - **`ClassePrincipe$2.class`**

Université Nice Sophia Antipolis

Exemple: java.io.FilenameFilter

- Classe interne

```
class Exemple {
  private String extension;
  class Filtre implements FilenameFilter {
    public boolean accept(File dir, String name) {
      return name.endsWith("."+extension);
    }
  }
  void liste(String dir) {
    File f = new File(dir);
    f.list(this.new Filtre());
  }
}
```

Université Nice Sophia Antipolis

Lien avec l'objet englobant

- La classe interne peut accéder les membres (champs et méthodes) de l'**objet** englobant
- Accès à l'instance de la classe avec **`<NomDeClasse>.this`**
- Création d'une instance d'une classe interne
  - **`<Instance de NomDeClasse>.new <Inner>( )`**
  - Ex : **`Exemple.Filtre f = ex.new Filtre();`**
    - **`Exemple`** est une classe contenant une classe interne **`Filtre`** ;
    - **`ex`** est une instance de la classe **`Exemple`**

Université Nice Sophia Antipolis

Exemple: java.io.FilenameFilter

- Classe interne statique

```
class Exemple {
  static class Filtre implements FilenameFilter {
    public boolean accept(File dir, String name) {
      return new File(dir, name).isDirectory();
    }
  }
  void liste(String dir) {
    File f = new File(dir);
    f.list(new Exemple.Filtre());
  }
}
```

Université Nice Sophia Antipolis

Lien avec la classe englobante

- La classe interne **statique** peut accéder auxmembres **statiques** (champs et méthodes) de la classe englobante
- Création d'une instance d'une classe interne
  - **`new <NomDeClasse>.<StaticInner>( )`**
  - Ex : **`Exemple.Filtre f = new Exemple.Filtre();`**
    - **`Exemple`** est une classe contenant une classe interne statique nommée **`Filtre`** ;

Université Nice Sophia Antipolis

Dans un bloc d'instructions

- Ne peuvent pas être utilisées en dehors du bloc
- Utiles pour personnaliser des objets sans créer de classe (masquer l'implantation).
- Exemple:

```
java.awt.Point getP(int x, int y) {
  class MyPoint extends java.awt.Point {
    MyPoint(int x, int y){
      this.x = x;
      this.y = y;
    }
  }
  return new MyPoint(x,y);
}
```

Université Nice Sophia Antipolis

Classes anonymes

- Une classe sans nom
  - définie par un `new <NomDeClasse>(…) { … }`
- Spécialise n'importe quelle classe, interface ou classe abstraite
- Une classe anonyme **ne peut pas avoir de constructeur !**
- Si on a besoin d'une référence extérieure,
  - il faut que l'élément soit **`final`** !

```
java.awt.Point getP(int x, int y) {
  return new java.awt.Point(x,y) {
    public String toString {
      return "MonPoint("+x+","+y+")";
    }
  };
}
```