Classes internes

Université Française d'Égypte
Version O 3.10.2 – 9/12/19
Richard Grin

1

2 types de classes internes

- Classes définies à l'extérieur de toute méthode
- Classes définies à l'intérieur d'une méthode

R. Grin

Classes internes

2

Classes internes non incluses dans une méthode

R. Grin

Classes internes

3

Emplacement du code

- A l'intérieur d'une classe englobante, au même niveau que les autres membres :

```
public classe ClasseE {
    private int x;

    class ClasseI {
        . . .
    }
    Code de la classe interne

    public String m() { . . . }
    . . .
}
```

R. Grin

Classes internes

4

Nommer une classe interne

- ClasseE.ClasseI

R. Grin

Classes internes

5

2 types de classes internes définies à l'extérieur d'une méthode

- Classe `static` : une instance n'est pas liée à une instance de la classe englobante
- Classe `non static` : une instance est liée à une instance de la classe englobante

R. Grin

Classes internes

6

5

6

Classe interne static

- A peu près le même rôle qu'une classe non interne
- Interne pour indiquer que la classe n'a de sens qu'en relation avec la classe externe

R. Grin

Classes internes

7

7

Classe interne non static

- Une instance d'une classe interne non static ne peut exister que liée à une instance de la classe englobante
- Le code de la classe interne peut désigner cette instance de la classe englobante par `ClasseE.this`

R. Grin

Classes internes

8

8

Exemple de classe interne non static

```
public class Tableau<T>
    implements Iterable<T> {
    private Object[] valeurs;
    private int nbElements;
    public Tableau(int n) {
        valeurs = new Object[n];
    }
    ...
    public Iterator<T> iterator() {
        return new IterTableau();
    }
    private class IterTableau implements Iterator<T> {
        // Définition de la classe interne dans le transparent suivant
    }
}
```

Classe qui enveloppe un tableau

Elle fournit un itérateur pour parcourir ce tableau ; cet itérateur est associé à une instance particulière de Tableau

Définition de la classe interne dans le transparent suivant

R. Grin

Classes internes

9

9

Définition de la classe IterTableau

```
public class Tableau<T> implements Iterable<T> {
    private Object[] valeurs;
    private int nbElements;
    ...
    private class IterTableau implements Iterator<T> {
        private int indiceCourant;

        public boolean hasNext() {
            return indiceCourant < nbElements;
        }

        public T next() {
            return (T) valeurs[indiceCourant++];
        }
    }
}
```

Accès direct aux attributs privés de la classe englobante

R. Grin

Classes internes

10

10

Utilisation de Tableau

```
Tableau<Integer> t1 = new Tableau<Integer>(10),
                t2 = new Tableau<Integer>(5);
// Ajout de quelques éléments dans t1 et t2
...
// Affichage des éléments de t1 et t2
for(int i : t1) {
    System.out.println(i);
}
for(int i : t2) {
    System.out.println(i);
}
```

R. Grin

Classes internes

11

11

Interne ou pas, static ou pas

- Si une classe n'est utilisée que par une seule autre classe, il est préférable qu'elle soit interne à l'autre classe
- Si une classe est interne et si son code utilise les variables d'état de la classe englobante, elle ne doit pas être static ; sinon elle doit être static

R. Grin

Java avancé

12

12

Classe interne locale (incluse dans une méthode)

R. Grin

Classes internes

13

13

Classe interne locale, définie à l'intérieur d'une méthode

- 2 possibilités :
 - classe avec un nom
 - classe anonyme

R. Grin

Classes internes

14

14

Utilisation d'une classe anonyme

- Une classe interne anonyme sert à (re)définir « à la volée » une ou plusieurs méthodes d'une classe (ou d'une interface)

R. Grin

Classes internes

15

15

Création d'une instance d'une classe interne anonyme

- Pas de constructeur puisque pas de nom
- La création fait appel au constructeur de la classe mère (il doit exister avec la bonne signature) :

```
Cercle c = new Cercle(p, r) {
    public void dessineToi(Graphics g) { ... }
};
```

Définition de la classe anonyme,
classe fille de Cercle

R. Grin

Classes internes

16

16

Classe anonyme qui implémente une interface

- La classe mère est remplacée par une interface qu'implémente la classe anonyme ; la liste des paramètres du « constructeur » doit être vide :

```
new Iterator<T>() {
    // Définition de la classe
    . . .
}
```

R. Grin

Classes internes

17

17

Exemple

```
public class Tableau<T> implements Iterable<T> {
    . . .
    public Iterator<T> iterator() {
        return
        new Iterator<T>() {
            private int indiceCourant;

            public boolean hasNext() {
                return indiceCourant < valeurs.length;
            }
            . . .
        };
    }
}
```

Définition classe anonyme

R. Grin

Classes internes

18

18

Avantages des classes anonymes

- Si le code de la classe anonyme est court, elle améliore la lisibilité car le code de la classe est proche de l'endroit où il est utilisé

R. Grin

Classes internes

19

19

Inconvénients des classes anonymes

- Si le code est long, la classe anonyme va au contraire nuire à la lisibilité
- Pas possible de créer plusieurs instances de la classe

R. Grin

Classes internes

20

20

Contexte d'exécution des classes internes locales

- Les classes internes locales ont accès
 - aux variables d'instance et de classe de la classe englobante (même privées)
 - aux paramètres et variables locales « effectivement final » (jamais modifiés après leur initialisation)

R. Grin

Classes internes

21

21