Java EE Spring, prêt à l'emploi

Types génériques

Université de Lille

Introduction

Types génériques (en anglais generics)

- Élément de syntaxe du langage Java
- Apparu avec Java 5 (2004)

- But : écrire du code générique utilisable avec différents types de données

- Concept existant dans d'autres langages (ex. C#, C++)

Motivation

Une classe pour gérer une bibliothèque de livres

```
public class Bibliotheque {
   public boolean emprunter( Livre l, int duree ) {...}
   public void rendre( Livre l ) {...}
   public boolean estDisponible( Livre l ) {...}
   public Livre lePlusEmprunte() {...}
}
```

La classe `Bibliotheque` utilise la classe `Livre`

- Et maintenant une bibliothèque de DVD ?
- Écrire une 2ème classe en remplaçant Livre par DVD ?
  - fastidieux, source d'erreurs, perte de temps

Motivation

Solution : généraliser le type utilisé par la classe `Bibliotheque`

```
public class Bibliotheque<T> {
   public boolean emprunter( T l, int duree ) {...}
   public void rendre( T l ) {...}
   public boolean estDisponible( T l ) {...}
   public T lePlusEmprunte() {...}
}
```

Utilisation

```
Bibliotheque<Livre> bl = new Bibliotheque<Livre>();
Bibliotheque<DVD> bdvd = new Bibliotheque<DVD>();
Bibliotheque<DVD> bdvd8 = new Bibliotheque<>();
```

Motivation

API Collection

```
List<String> l = new ArrayList<>();

Set<Livre> s = new HashSet<>();

Map<Integer,String> m = new HashMap<>();

List<Bibliotheque<Livre>> lbl = new ArrayList<>();
```

Améliore le typage des programmes

- Sans les types génériques

```
List l = new ArrayList();
l.add("hello world!");
l.add(42);
```

- Avec les types génériques

```
List<String> ls = new ArrayList<>();
ls.add("hello world!");
ls.add(42);  // erreur indiquée par le compilateur

List<Object> lo = new ArrayList<>();
```

Autres concepts

1. Type inconnu
2. Borne sur le paramètre de type

Type inconnu

Usages avec plusieurs types

```
void trier( List<Livre> l ) {...}
void trier( List<DVD> l ) {...}
```

- Une liste d'éléments dont le type est inconnu (wildcard)

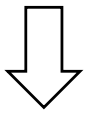
```
void trier( List<?> l ) {...}
```

- On peut invoquer la méthode

```
trier( new ArrayList<Livre>() );
trier( new ArrayList<DVD>() );
```

Borne sur le paramètre de type

Restriction avec une borne supérieure (type parent)



```
class Personne {...}
class Etudiant extends Personne {...}
class Bachelier extends Etudiant {...}

void classer( List<? extends Etudiant> l ) {...}

classer( new ArrayList<Etudiant>() );
classer( new ArrayList<Bachelier>() );
classer( new ArrayList<Personne>() );
classer( new ArrayList<String>() );
classer( new ArrayList<Object>() );
```

Borne sur le paramètre de type

Restriction avec une borne inférieure (type enfant)

```
class Personne {...}
class Etudiant extends Personne {...}
class Bachelier extends Etudiant {...}

void trier( List<? super Etudiant> l ) {...}

trier( new ArrayList<Etudiant>() );
trier( new ArrayList<Bachelier>() );
trier( new ArrayList<Personne>() );
trier( new ArrayList<String>() );
trier( new ArrayList<Object>() );
```

Borne sur le paramètre de type

Utilisation dans l'API Collections

```
interface List<E> {
        boolean addAll( Collection<? extends E> c );
        ...
}

List<Etudiant> le = new ArrayList<>();
le.addAll( new ArrayList<Etudiant>() );
le.addAll( new ArrayList<Bachelier>() );
le.addAll( new ArrayList<Personne>() );
```

Borne sur le paramètre de type

Utilisation dans l'API Collections

```
class Collections {
        static <T> void copy(
                List<? super T> dest,
                List<? extends T> source ) {...}
        ...
}

List<Personne> lp = new ArrayList<>();
List<Etudiant> le = new ArrayList<>();
List<Bachelier> lb = new ArrayList<>();
Collections.copy( new ArrayList<Etudiant>(), le );
Collections.copy( new ArrayList<Etudiant>(), lb );
Collections.copy( new ArrayList<Etudiant>(), lp );
```

En résumé

- Un mécanisme qui permet d'écrire du code générique
- Effort d'abstraction
- Améliore le typage des programmes
- Évite des erreurs
- Pas de surcoût à l'exécution grâce au mécanisme type erasure
- Abondamment utilisé dans l'API Collections