

Génie logiciel CB2

TP4 Noté et individuel

Extension du projet de génération

Jean-Claude Royer

17 août 2010

1 Le projet existant

Le projet fourni comme sujet est la solution du TP3 précédent, y compris la question subsidiaire permettant d'imbriquer des listes. Il va s'agir de restructurer et d'étendre le modèle UML puis de modifier le projet en conséquence. Voici pour rappel le diagramme des classes, le fichier `tp3.zargo` vous est également fourni.

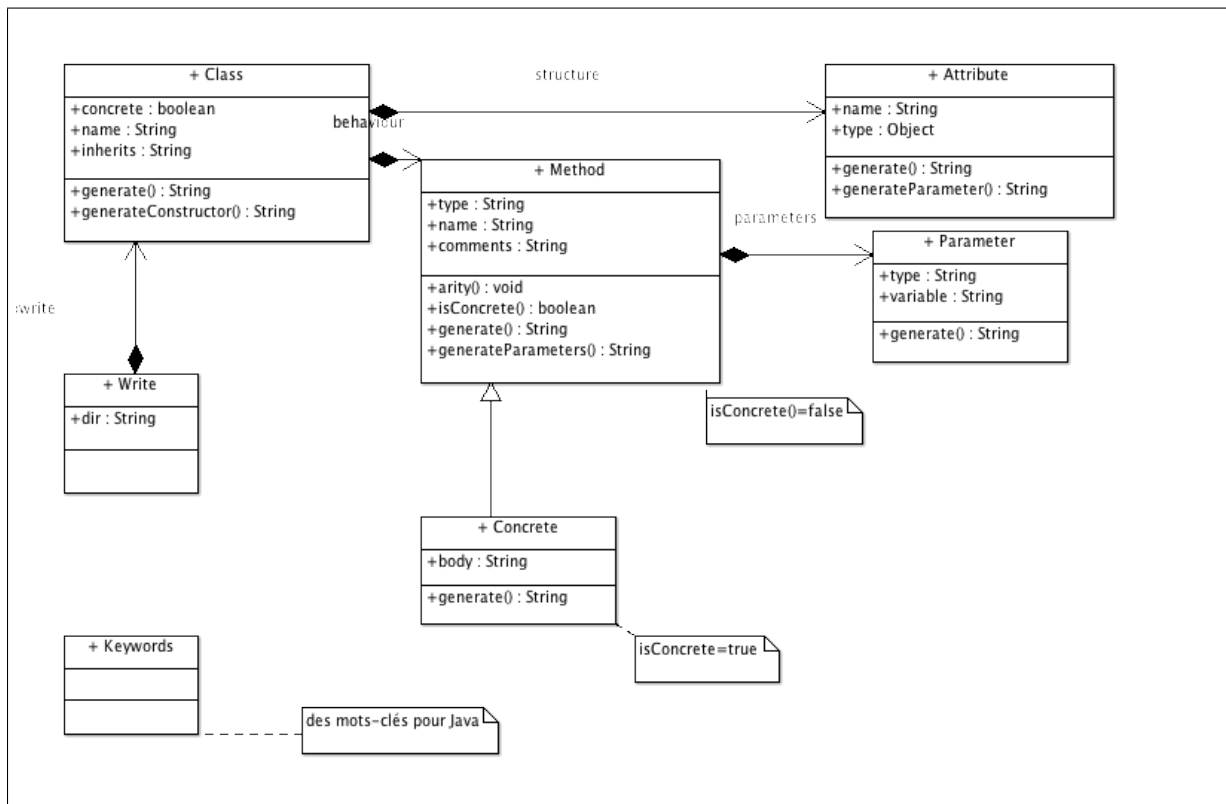


FIGURE 1 – Diagramme des classes du TP3

2 Extension du projet

2.1 Le modèle

Lancer ArgoUML avec le .zargo et résoudre les points suivants.

1. Enrichir le diagramme pour qu'un attribut puisse avoir un **scope** et un **gener**. Le **scope** indique la visibilité qui sera **private**, **protected** ou **public**. Le **gener** peut prendre quatre valeurs : **no**, **get**, **set**, **getset**.
2. Ajouter aussi le **scope** pour les méthodes.

Dans votre diagramme vous devez placer les champs, les méthodes et leurs redéfinitions.

2.2 L'implémentation

Reprendre le projet Eclipse du TP3 qui vous est fourni comme sujet.

1. Définir la génération d'un constructeur de copie superficielle. Si la classe n'est pas abstraite et si elle a des attributs le générateur doit créer un nouveau constructeur. Celui-ci prends un argument du même type que la classe et crée une nouvelle instance en initialisant les champs avec les valeurs correspondantes dans l'objet argument du constructeur de copie. Le principe est illustrée sur l'exemple de la classe `NotEmpty` :

```
public NotEmpty(NotEmpty other) {  
    this.head = other.head;  
    this.tail = other.tail;  
}
```

2. Prendre en compte la génération du **scope** pour les attributs et les méthodes.
3. Implémenter la sémantique du **gener** pour les attributs qui ne sont pas **public**. Le principe est le suivant : **no** signifie rien à générer, **get** signifie générer un getter publique, **set** générer un setter publique et **getset** les deux. Bien sûr il faudrait suivre les conventions de nommage en usage en Java, i.e. générer les noms des accesseurs à partir des noms des attributs et en majuscule la première lettre.
4. Vous générerez également un commentaire de la forme

```
/**  
 * getter for head attribute  
 * @return String element  
 */
```

et de même pour un setter.

5. +++ autre chose

3 Travail à rendre

Archive .zip à déposer sur campus et contenant :

1. le code source du projet avec la même structure que le projet initial
2. le diagramme UML étendu, nommé `diagrammeV2.png` et placé dans **ressources**
3. un fichier **COMMENTS** contenant vos remarques au sujet de votre travail et placé dans **ressources**