Université Joseph Fourier — UFR - IMAG
MIAGE — Licence 3
Année 2007-2008

La généricité

Sébastien Laborie et Philippe Morat

1 Objectifs

Construire des classes dites génériques et aborder l'héritage de telles classes.

2 Rappel

Le concept de "généricité" consiste à appliquer une classe (e.g., ses méthodes) à tous types d'objets.

En Java, on peut identifier deux formes de généricité :

1. Avec le type *Object*.
2. Avec le type $< T >$.

L'avantage du deuxième point est que l'on contrôle mieux le type.

Question 1 *Montrez via l'exemple de votre choix cette différence entre les points 1. et 2. précédents.*

3 Création de classes génériques

On souhaite implémenter des classes génériques modélisant des objets de type *File* qui contiennent des éléments de type T. Celles-ci doivent être en mesure d'ajouter et supprimer un élément de la file, accéder à un élément de la file, donner la taille de la file...

Question 2 *Réalisez l'interface* `_File` *identifiant les comportements d'une file.*

Deux types de file peuvent être envisageables :

- Une *Pile* (LIFO) : Dernier entré / Premier sorti.
- Une *Queue* (FIFO) : Premier entré / Premier sorti.

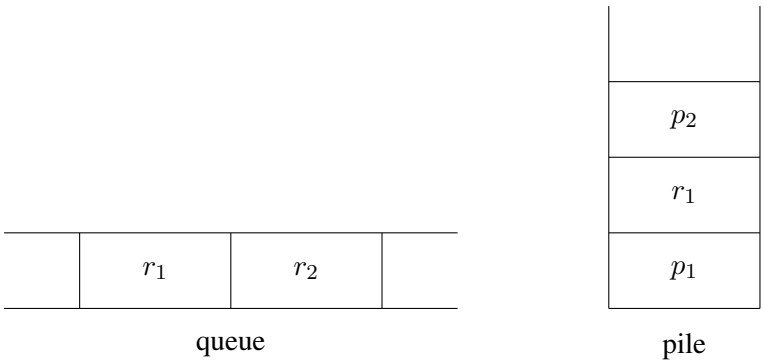
Question 3 *Réalisez les interfaces* `_Pile` *(i.e., LIFO) et* `_Queue` *(i.e., FIFO) ayant les comportements d'une file.*

Question 4 *Implémentez les interfaces* `_Pile` *et* `_Queue`*, vues dans la question précédente, dans deux classes* `Pile` *et* `Queue`*.*

4 Manipulation de classes génériques

On souhaite vérifier que nos classes `Pile` et `Queue` fonctionnent avec des objets de type `Parallelogramme` et `Rectangle`.

Soit la situation suivante contenant les références *queue* et *pile* :



Les variables p_i correspondent à des objets de type *Parallelogramme* et les r_i correspondent à des objets de type *Rectangle*.

Question 5 *Complétez la méthode* `public static void main(String[] args) {...}` *menant à la situation décrite ci-dessus.*

Question 6 *Quels sont tous les appels possibles permettant de construire la référence* queue *?*

5 Des classes génériques en paramètre de méthode

On souhaite disposer, dans le comportement d'une file, d'une méthode nommée `fusionner` qui prend en paramètre une file.

Nous proposons quatre spécifications possibles pour cette méthode :

- `public void fusionner(_File<T> file);`
- `public <T2> void fusionner(_File<T2> file);`
- `public void fusionner(_File<?> file);`
- `public void fusionner(_File<? extends T> file);`

Question 7 *Pour chacune des quatre propositions ci-dessus, dire si cette spécification est possible et identifiez les propositions limitantes.*