IUT de Paris Descartes – Base de la Programmation Objet

Travaux Dirigés – Sujet n°6

Objectifs

- Comprendre la visibilité des classes
- Généraliser une classe par l'emploi d'un paramètre générique

Listes génériques

Le but de l'exercice est de généraliser une classe conçue pour gérer des listes d'entiers à des listes d'éléments d'un type quelconque. La définition initiale de la classe `Liste` est la suivante :

```
// Liste.java
package listes;

// visiblité paquetage
class Maillon {
  public int valeur;
  public Maillon suivant;
  public Maillon(int valeur) {
    this.valeur = valeur;
    this.suivant = null;
  }
}

public class Liste {
  private Maillon premier;

  public Liste() {
    premier = null;
  }

  public Liste(Liste li) {
    this();
    for (Maillon m = li.premier;
         m != null; m = m.suivant)
      ajouter(m.valeur);
  }

  public void ajouter(int v) {
    Maillon m = new Maillon(v);
    if (premier == null)
      premier = m;
    else {
      Maillon tmp = premier;
      while (tmp.suivant != null)
        tmp = tmp.suivant;
      tmp.suivant = m;
    }
  }
}
```

Notez que cette classe repose sur une autre classe représentant les maillons de la liste chaînée. Les classes `Liste` et `Maillon` sont définies dans un unique fichier `Liste.java`. Cela est possible uniquement parce que la classe `Liste` est la seule classe publique. La classe `Maillon` a une visibilité *paquetage* et les deux classes sont les deux seules ayant été définies au sein du paquetage `listes`. Ainsi, la classe `Liste` est la seule à pouvoir employer la classe `Maillon`.

1. Ajoutez une méthode `toString` à la classe `Liste`.
2. Écrivez un court programme qui ajoute les valeurs 7, 9, 11 et 6 à une liste puis affiche son contenu. Ce programme doit être défini en dehors du paquetage `listes`.

3. La méthode `ajouter` de la classe `Liste` doit parcourir tous les éléments avant de pouvoir ajouter la nouvelle valeur à la fin de la chaîne. Ajoutez un attribut référençant le dernier maillon de la chaîne et exploitez-le pour accélérer l'ajout d'une valeur.
4. Pour permettre l'accès aux données, le paquetage `listes` propose la classe `Itérateur` ainsi définie :

```
package listes;

public class Itérateur {
  private Maillon courant;

  public Itérateur(Liste li) {
    courant = li.getPremier();
  }

  public boolean estFini() {
    return courant == null;
  }

  public int valeur() {
    return courant.valeur;
  }

  public void suivant() {
    courant = courant.suivant;
  }
}

public class Liste {
  ...
  // visiblité paquetage
  Maillon getPremier() {
    return premier;
  }
  ...
}
```

   Notez qu'une méthode (`getPremier`) a été ajoutée à la classe `Liste`. Pour des raisons évidentes d'encapsulation, cette méthode est de visibilité *paquetage*. Réécrivez la méthode `toString` en vous reposant sur un itérateur.
5. Généralisez la classe `Liste` à tout type d'élément. Il est suffisant d'indiquer les différences à introduire par rapport à la définition des classes `Maillon`, `Itérateur` et `Liste`.
6. Adaptez en conséquence le programme de la question 2.