

IUT de Paris Descartes – Base de la Programmation Objet

Travaux Dirigés – Sujet n°2

Objectifs

- Savoir définir des types de données structurés
- Savoir manipuler des listes simples

Manipulation de commandes

1. Sachant qu'un produit est caractérisé par son libellé (une chaîne de caractère) et son prix (un réel), définissez un type de donnée nommé `Produit` réunissant ces caractéristiques.

Solution:

```
public class Produit {
    String libellé;
    double prix;
}
```

2. Écrivez un court programme définissant un produit ("Sophie la girafe" à 11€99) puis affichant, sur la sortie standard, son libellé suivi de son prix entre parenthèses.

Solution:

```
public class Commande {
    public static void main(String[] args) {
        Produit p = new Produit();
        p.libellé = "Sophie_la_girafe";
        p.prix = 11.99;
        System.out.println(p.libellé + "(" + p.prix + ")");
    }
}
```

3. Une commande est composée d'une liste de produits. Pour chacun de ces produits est spécifié la quantité commandée. Définissez le type `LigneDeCommande` qui associe une quantité à un produit.

Solution:

```
public class LigneDeCommande {
    Produit produit;
    int quantité;
}
```

4. La bibliothèque standard de Java dispose d'un type nommé `ArrayList`. Ce type représente les tableaux dynamiques (la taille croie en fonction des besoins tel que vous les avez vu dans le module SDA). Nous décidons de représenter une commande par une variable de type `ArrayList<LigneDeProduit>`.

En vous basant sur l'extrait de la documentation de `ArrayList` fourni en annexe, écrivez une fonction permettant de connaître le montant global d'une commande.

Solution:

```
public class Commande {
    public static double montant (ArrayList<LigneDeCommande> cde) {
        double total = 0.;
        for (int i = 0; i < cde.size (); ++i)
            total += cde.get(i).quantité * cde.get(i).produit.prix;
        return total;
    }
}
```

5. Sachant que les éléments composant une `ArrayList` peuvent être parcourus par une boucle 'foreach', écrivez une fonction retournant la chaîne de caractère représentative d'une commande.

Chaque ligne de la commande devra faire apparaître la quantité commandée suivi du libellé du produit et de son prix unitaire. Ce dernier sera entre deux parenthèses. Les lignes de la commande seront séparées par des retours à la ligne (employez le caractère '\n', ou mieux, la fonction standard `System.lineSeparator()`).

Solution:

```
public class Commande {
    public static String toString (ArrayList<LigneDeCommande> cde) {
        String s = "";
        for (LigneDeCommande ldc : cde)
            s += ldc.quantité + "_" + ldc.produit.libellé + "(" +
                ldc.produit.prix + ")" + System.lineSeparator();
        return s;
    }
}
```

6. Écrivez une fonction permettant d'ajouter une ligne de commande à une commande existante. Vous ferez en sorte qu'un même produit (même libellé et même prix) n'apparaisse qu'une seule fois au sein d'une même commande. Si le produit commandé est déjà présent dans la commande, sa quantité doit être augmentée de celle à nouveau commandée.

Solution:

```
public class Commande {
    public static void ajouter (ArrayList<LigneDeCommande> cde,
                                LigneDeCommande ligne) {
        for (LigneDeCommande ldc : cde)
            if (ldc.produit.libellé.equals(ligne.produit.libellé) &&
                ldc.produit.prix == ligne.produit.prix) {
                ldc.quantité += ligne.quantité;
                return;
            }
        cde.add(ligne);
    }
}
```

```
}
```

Annexe – ArrayList

E désigne le type des éléments stockés au sein de la liste (e.g. `LigneDeCommande` dans l'exercice).

- `ArrayList<E>()`
Constructs an empty list with an initial capacity of ten.
- **boolean** `add(E e)`
Appends the specified element to the end of this list and returns **true**
- **int** `size()`
Returns the number of elements in this list.
- `E` `get(int index)`
Returns the element at the specified position in this list.

La documentation complète du type `ArrayList` est accessible ici :
<https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>