

UE Programmation Orientée Objet

(Devoir Surveillé du jeudi 20 mars 2008)

Les classes `Ville` et `Adresse` du paquetage `ds1.util` sont définies ainsi :

Ville
- nom : String
-codePostal : String
+Ville(nom : String, cp : String)
+equals(o: Object) : boolean
+toString() : String
+getNom():String
+getCodePostal() : String

Adresse
- numero : int
- rue : String
- ville : Ville
+Adresse(num : int, rue : String, v : Ville)
+equals(o: Object) : boolean
+toString() : String
+getNumero(): int
+getRue() : String
+getVille() : Ville

Q 1. Donnez le diagramme UML puis le code JAVA d'une classe `Habitant` du paquetage `ds1` sachant qu'un objet `Habitant` est caractérisé par son nom (une chaîne de caractères) et son adresse du type `Adresse` et que l'on souhaite disposer des méthodes :

- `getNom` et `getAdresse` permettant d'accéder à ces informations,
- `equals` (mêmes noms, mêmes adresses) et `toString` habituelles, `toString` fournira une chaîne de la forme : "*nom* habite à *adresse*", où *nom* et *adresse* sont remplacés par les informations correspondantes aux attributs.
- `demenage` permet de modifier l'adresse d'un habitant et prend en paramètre la nouvelle adresse de l'habitant,
- `memecp` qui prend en paramètre un habitant *h* et renvoie *true* si et seulement si *h* habite une ville de même code postal que *this*.

On dispose de l'interface suivante du paquetage `ds1.util` :

« interface » <i>Courrier</i>
+getCout():float +getExpéditeur():Habitant +getDestinataire():Habitant

Le coût d'un courrier, fourni par la méthode `getCout`, est exprimé en Euros.

On dispose également d'une classe `BureauPoste` dont une partie du code est :

```
package ds1;
import ds1.util.DesCourriers;
public class BureauPoste {
    /** coût d'un timbre */
    public static final float COUT_TIMBRE = 0.55f;
    private Ville ville;
    private DesCourriers lesCourriers;
    public BureauPoste(Ville v) {
        this.ville = v;
        this.lesCourriers = new DesCourriers();
    }
    /** fournit les courriers dont le destinataire est h
     * @return les courriers pour h
     */
    public DesCourriers courriersPour(Habitant h) { ... }
    ...
    public static void main(String[] args) {...}
}
```

La classe `DesCourriers` du paquetage `ds1.util` permet de gérer des paquets de `Courrier` et est définie ainsi :

DesCourriers
...
+DesCourriers() +ajouteCourrier(op : Courrier) +getNbObjetsContenus():int +getCourrier(index :int):Courrier

- à la construction le nombre de Courrier est 0,
- `getCourrier` lève une exception `NoSuchElementException` si il n'y a pas de courrier numéro `index`.

On identifie (au moins) 2 types de courriers des lettres et des colis, ce qui a pour conséquence que l'interface `Courrier` est implémentée par 2 classes : `Lettre` et `Colis` du paquetage `ds1`.

Q 2. On s'intéresse à nouveau à la classe `Habitant` et on ajoute à cette classe la méthode :

```
/** l'habitant (this) reçoit le courrier c
 * @param c le courrier reçu
 */
public void recoitCourrier(Courrier c) { ... }
```

Le code de cette méthode est supposé connu.

On ajoute également à la classe `Habitant` une méthode `mailingLettre` :

```
/** l'habitant (this) envoie une lettre à chacun des habitants de dest
 * @param dest les habitants destinataires
 * @param msg le message de la lettre envoyée
 */
public void mailingLettre(Habitant[] dest, String msg) { ... }
```

Q 2.1. Donnez la ligne de code permettant de créer un tableau de 10 habitants.

Q 2.2. Donnez le code JAVA de la méthode `mailingLettre` sachant que la classe `Lettre` dispose d'un constructeur :

```
/** construit une Lettre avec un texte donné
 * @param exp l'expéditeur de la lettre
 * @param dest le destinataire de la lettre
 * @param msg le message de la lettre envoyée
 */
public Lettre(Habitant exp, Habitant dest, String msg) { ... }
```

Q 3. Les instances de `Lettre` sont des objets postaux donc le coût fixe est celui d'un timbre.

Q 3.1. Donnez toutes les valeurs de type que peut prendre `T` dans l'expression :

```
T ref = new Lettre(...);
```

Q 3.2. Donnez le code de la méthode `get Cout ()` de la classe `Lettre` (et uniquement cette méthode).

Q 4. Les instances de `Colis` sont des objets postaux. Ils sont également caractérisés par leur poids, en grammes, et disposent donc d'une méthode `getPoids ()` permettant l'accès à cette information.

Le coût d'un colis est calculé ainsi :

- le coût de base est de 1 Euro par tranche de 1000g commencée (donc 1 pour un colis de 250g et 4 pour un colis de 3512g, etc.)
- cette base est augmentée de 10% si le code postal du destinataire diffère de celui de l'expéditeur.

Donnez le code complet de la classe `Colis`. La construction d'un objet `colis` est paramétré par l'expéditeur, le destinataire et le poids du colis.

Q 5. Après avoir précisé le répertoire courant, **indiquez quelle commande** permet d'exécuter le "main" de la classe `BureauPoste` ?

Q 6. Donnez le code de la méthode `courriersPour`.