IUT Paris Descartes – Base de la Programmation Objet

DST 2017 – 2018

3 heures – tous documents autorisés – *barème indicatif*

Il nous est demandé de proposer des moyens informatiques permettant à une entreprise de petites annonces (ventes et achats de bien et de service, propositions et demandes d'emploi, etc) de gérer les aspects financiers liés à ses affaires.

Mise en bouche

Nous commençons par un composant technique. L'application va employer beaucoup de prix et de taux. Tous sont constants et doivent être accessibles à partir de différentes classes. Il a été décidé de centraliser la gestion des constantes. Toute constante est identifiée par une chaine de caractères et seules les valeurs réelles sont supportées. Pour ce faire, une classe nommée `Constantes` doit être développée et elle doit passer avec succès le test suivant.

```
public class ConstantesTest {
  @Test
  public void test() {
    Constantes.ajouter("forfait", 5.0); // un montant
    double d = Constantes.obtenir("forfait");

    assertTrue(d == 5.0);

    Constantes.ajouter("pourcentage", 0.05); // un pourcentage
    assertTrue(Constantes.obtenir("forfait") == 0.05);

    try {
      d = Constantes.obtenir("inconnue");
      fail("Une RuntimException doit être levée car ce nom est inconnu");
    }
    catch (RuntimeException e) {
    }
    try {
      Constantes.ajouter("forfait", 2.0);
      fail("Une RuntimException doit être levée car ce nom est déjà occupé");
    }
    catch (RuntimeException e) {
    }
  }
}
```

1. (2 points) Programmez la classe `Constantes`. Vous noterez que les seuls éléments publics sont les deux méthodes de classe `ajouter` et `obtenir`.

Entrée

Une annonce est toujours associé à un montant exprimé en euro. Selon la nature de l'annonce, il peut représenter un prix de vente, un salaire, etc. Lorsqu'une annonce est conclue (la vente est réalisée, le salarié est embauché, etc), notre entreprise est rémunérée pour le service rendu. Le montant de cette rémunération dépend de l'annonce. Il peut être fixe (en appliquant un forfait ou déterminé lors d'une négociation) ou être un pourcentage du montant associé à l'annonce.

Nos ingénieurs ont décidés de se reposer sur l'interface `Annonce` suivante.

```
public interface Annonce {
  double montant(); // montant de l'annonce
  double dime();    // rémunération perçue lorsque l'annonce est conclue
}
```

2. (3 points) Écrivez une classe nommée `AnnoncesConclues` stockant toutes les annonces déjà conclues. Pour chaque annonce, vous devez mémoriser la date à laquelle elle a été conclue. Avec votre classe, nous devons pouvoir
   - ajouter une annonce (sa date de conclusion sera la date courante lors de l'ajout) ;
   - connaître le volume global (le cumul des montants) des annonces conclues ;
   - connaître combien nous a rapporté (le cumul des dimes) les annonces conclues entre deux dates ;
   - purger la liste.

   Pour ce faire, vous disposez de la classe `Date` de la bibliothèque standard (`java.util.Date`) :

```
public class Date {
    public Date() { ... } // construit une date représentant la date courante
    public boolean before(Date d) { ... } // indique si la date est antérieure à d
    public boolean after(Date d) { ... } // indique si la date est postérieure à d
    ...
}
```

3. (3 points) La majeur partie des annonces vont devoir mémoriser leur montant. Écrivez une classe nommée `AnnonceValuée` concrétisant partiellement une annonce munie d'un montant. Vous devez faire au plus simple. Toutefois, le montant d'une telle annonce doit pouvoir être modifié. De plus, il est nécessairement supérieur ou égal à la valeur de la constante nommée `"prix plancher"`.

Plat principal

4. (3 points) Définissez deux classes concrètes d'annonce valuée. La première, nommée `AnnonceNégociée`, rapporte à l'entreprise une rémunération (une dîme) dont le montant est négocié à la création de l'annonce. Vous devez vous assurer que le montant de la rémunération est bien couvert par le montant de l'annonce. La seconde, nommée `AnnonceAuPourcentage`, rapporte un montant proportionnel au montant de l'annonce. Le pourcentage appliqué est stocké dans la constante nommée `"pourcentage"`.
5. (3 points) Nos commerciaux souhaitent proposer un nouveau type d'annonce, les annonces groupées. Les clients répondant à une telle annonce recevront une réduction (le pourcentage correspondant est défini dans la constante nommée `"réduction"`) sur le prix individuel des annonces composant le groupe. *A contrario*, notre entreprise n'appliquera aucune réduction sur sa rémunération. Programmez une classe nommée `AnnonceGroupée` représentant un tel type d'annonce.

Dessert

Pour faciliter la recherche des clients parmi les annonces, il nous est demandé de proposer un moyen de sélectionner celles qui satisfont un critère donné. Après réflexion, les ingénieurs proposent de nous reposer sur l'interface suivante.

```
public interface Critère {
  boolean estSatisfaitPar(Annonce a);
}
```

6. (2 points) Programmez la méthode de classe permettant de sélectionner les annonces respectant un critère donnée et ayant le prototype suivant :

```
List<Annonce> sélection(List<Annonce> annonces, Critère critère)
```

7. (2 points) Écrivez le critère permettant de sélectionner les annonces dont le montant est compris entre deux valeurs données.
8. (2 points) Proposez un moyen de combiner logiquement (négation $\neg$, conjonction $\wedge$, disjonction $\vee$) des critères dans une recherche.