IUT de Paris Descartes – Base de la Programmation Objet

Travaux Dirigés – Sujet n°4

Objectifs

- Concevoir une classe
- Distinguer les attributs et méthodes d'instance des attributs et méthodes de classe
- Respecter le protocole pour l'affichage des objets

Un compte bancaire

Une banque suisse nous demande de développer une classe représentant un compte bancaire. Chaque compte est caractérisé uniquement par un numéro et un solde (les comptes sont anonymes).

À faire

1. Déclarez une classe nommée `Compte` et ses attributs d'instance.

   Solution:

   ```
   public class Compte {
     private int num;
     private long solde;
   }
   ```

2. Un compte bancaire ne peut être débité que si cette opération ne laisse pas le solde inférieur à un découvert maximum fixé à 1000€. Cette règle est commune à tous les clients. Déclarez une constante représentant ce maximum au sein de la classe `Compte`.

   Solution:

   ```
   public class Compte {
     private static final long DECOUVERT_MAX = 1000;
     ...
   }
   ```

3. Après réflexion, la banque juge que ce découvert maximum pourra évoluer dans le futur. Adaptez la classe de manière à ce que cela soit possible. On notera qu'un client a toujours le droit de retirer tout l'argent présent sur son compte.

   Solution:

   ```
   public class Compte {
     private static long découvertMax = 1000;
   ```

```
  public static void setDécouvertMax(long val) {
    assert(val >= 0);
    découvertMax = val;
  }
  ...
}
```

4. Indiquez comment le découvert maximal des comptes peut être positionné à 1500€.

Solution:

```
Compte.setDécouvertMax(1500);
```

5. Le découvert maximal doit pouvoir être connu des utilisateurs de la classe. Transformez la classe pour rendre cela possible.

Solution:

```
public class Compte {
  ...
  public static long getDécouvertMax() {
    return découvertMax;
  }
  ...
}
```

6. Programmez les méthodes permettant de réaliser un dépôt et un retrait. Vous préciserez les préconditions de chaque méthode et ferez en sorte qu'un utilisateur de la classe puisse s'assurer qu'elles sont satisfaites.

Solution:

```
public class Compte {
  ...
  public void déposer(long val) {
    assert (val > 0);
    solde += val;
  }

  public boolean retraitPossible(long val) {
    return val > 0 && solde - val >= -découvertMax;
  }

  public void retirer(long val) {
    assert (retraitPossible(val));
    solde -= val;
  }
}
```

7. Introduisez un constructeur de compte. Vous ferez en sorte que le numéro de compte soit automatiquement déterminé et que deux comptes aient toujours deux numéros distincts. Le solde initial de tout compte devra pouvoir être choisi.

Solution:

```
public class Compte {
  ...
  private static int numéroSuivant = 1;
  ...
  public Compte(long soldeInitial) {
    assert(soldeInitial >= 0);
    numéro = numéroSuivant;
    ++numéroSuivant;
    solde = soldeInitial;
  }
}
```

8. L'affichage d'un compte créditeur doit donner une chaîne telle que
   `le compte n°1 est crédité de 1234€`
   alors que celui d'un compte débiteur doit ressembler à
   `le compte n°1 est à découvert de 500€`
   Complétez la classe `Compte` pour permettre de tels affichages.

Solution:

```
public class Compte {
  ...
  @Override
  public String toString() {
    String s = "le compte n°" + numéro + " est ";
    if (solde >= 0)
      s += "crédité de " + solde;
    else
      s += "à découvert de " + (-solde);
     return s + "€";
  }
}
```