Licence 3 Informatique — Examen — 13/01/2021

Durée: 1h30

— Barème donné à titre indicatif
— Le respect des conventions de nommage et des principes de POO vus en cours sera pris en compte dans la notation.

Dans Eclipse, créez un projet à partir de l'archive `Exam_2020-2021.zip` : `Import` → `General` → `Existing Projects into Workspace`, puis cochez la case `Select archive file`, et choisissez le fichier correspondant.
Cette archive contient des packages et des classes dans lesquels vous pouvez taper vos réponses aux exercices, ainsi que du code que vous pouvez utiliser directement sans avoir à le modifier. Par exemple, dans la classe exo1.NombreBinaire, il y a la méthode :

```
public int intValue() {
        throw new UnsupportedOperationException();
}
```

Remplacez le code **throw new** UnsupportedOperationException(); par votre réponse à la question 2. de l'**Exercice I**.
Renommez le projet en y ajoutant votre nom, afin de faciliter l'identification des « copies ». Pour cela, clic droit sur le nom du projet, `Refactor` → `Rename`. À la fin de l'épreuve, exportez votre projet sous forme d'archive `zip` afin de l'uploader sur Moodle : `Export` → `General` → `Archive File`. L'utilisation d'un autre format (notamment `rar`), ou l'absence de votre nom sur la « copie », seront sanctionnés.

Exercice I (6 points)

On souhaite créer une classe qui représente un nombre entier naturel sous forme binaire. Pour cela, on utilise un tableau de booléens qui correspondent aux bits du nombre, avec le premier élément du tableau qui correspond au bit de poids faible, et le dernier élément du tableau qui correspond au bit de poids fort. Par exemple, le nombre entier 12 peut se représenter en binaire par 1100, où le bit de poids fort est le premier 1, et le bit de poids faible est le dernier 0. Pour nous, ce nombre correspondra au tableau {**false**, **false**, **true**, **true**} : le premier **false** correspond au bit de poids faible, et le dernier **true** correspond au bit de poids fort.

(2) 1. Implémentez les différents constructeurs de la classe NombreBinaire.

(1) 2. Implémentez la méthode intValue() (héritée de la classe abstraite Number), qui retourne la valeur du nombre sous forme d'un **int**.

(1) 3. Implémentez la méthode incrementer(), qui modifie le nombre binaire pour représenter l'entier suivant.

(1) 4. Implémentez la méthode toString(), qui retourne une représentation du nombre binaire sous forme de chaîne de caractères, en suivant la convention habituelle (c'est-à-dire avec des 1 et des 0, et le bit de poids fort en premier).

(1) 5. Implémentez la méthode compareTo(NombreBinaire n), qui provient de l'interface Comparable<NombreBinaire>. Rappelons que cette méthode doit retourner un entier négatif si l'objet courant est plus petit que celui passé en paramètre, positif s'il est plus grand, et 0 si les deux nombres sont égaux.

La classe exo1.TestNombreBinaire vous permet de tester votre code.

Exercice II (6 points)

Le package exo2 fournit une implémentation possible de la classe CommunauteAgglomeration (simplifiée) du projet.

(2) 1. Dans le dossier `test`, vous trouverez une classe exo2. TestAccessibilite . Créez dans cette classe des tests unitaires pour la méthode accessibilite () de la classe CommunauteAgglomeration.

(1) 2. Implémentez la méthode accessibilite (String nomVille) qui vérifie si une ville satisfait la contrainte d'accessibilité.

(1) 3. Implémentez la méthode accessibilite () qui vérifie si la communauté satisfait la contrainte d'accessibilité.

(2) 4. Implémentez la méthode sauvegarder(String nomFichier), qui enregistre dans le fichier dont le nom est passé en paramètre une description textuelle de la communauté d'agglomération :
   — pour chaque ville, le fichier contient une ligne `ville(XXX).` où `XXX` est le nom de la ville ;
   — pour chaque route, le fichier contient une ligne `route(XXX,YYY).` où `XXX` et `YYY` sont les noms des villes (il n'est pas nécessaire d'indiquer les deux directions, une seule suffit) ;
   — pour chaque école, le fichier contient une ligne `ecole(XXX).` où `XXX` est le nom de la ville où se trouve l'école.

   La classe exo2.TestCommunauteAgglomeration vous permet de tester votre code.

Exercice III (8 points)

Un polynôme à une variable et à coefficients entiers est une expression du type

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

où x est la variable, $n \in \mathbb{N}$, et a_i est le coefficient de degré i (pour $0 \leqslant i \leqslant n$). Mathématiquement, on peut également représenter un polynôme par un ensemble d'associations "degré" $\rightarrow$

"coefficient". Par exemple, le polynôme $2+3x-4x^3$ est représenté par $\{0 \to 2, 1 \to 3, 3 \to -4\}$, car $a_0 = 2$, $a_1 = 3$ et $a_3 = -4$. Implicitement, si un degré n'est pas présent, cela signifie que le coefficient correspondant est 0 (par exemple, $2 \to 0$, car il n'y a pas de terme en x^2 dans le polynôme). On souhaite définir une classe Polynome qui implémente l'interface IPolynome.

(2) 1. Définissez les attributs nécessaires au fonctionnement de la classe exo3.Polynome. Implémentez (au moins) un constructeur pour cette classe, en vous assurant qu'il lève une IllegalArgumentException sur un des degrés passés en paramètre est négatif.

(1) 2. Implémentez la méthode coefficient (**int** degre), qui retourne le coefficient associé à un certain degré.

(2) 3. Implémentez la méthode addition(IPolynome p), qui retourne un nouvel objet correspondant à la somme du polynôme courant avec celui passé en paramètre. Par exemple, si on applique cette méthode au polynôme $2+3x-4x^3$ en lui passant en paramètre $3-x+4x^2$, on obtient un nouvel objet qui représente le polynôme $5+2x+4x^2-4x^3$.

(2) 4. Implémentez la méthode evaluer(**double** x) qui évalue le polynôme pour une certaine valeur de x. Par exemple, pour le polynôme $2+3x-4x^3$, si on applique cette méthode avec le nombre 3 en paramètre, on obtient $2+3\times 3-4\times 3^3 = 2+9-4\times 27 = 11-108 = -97$.

(1) 5. Implémentez la méthode toString() pour donner une représentation textuelle d'un polynôme.

La classe exo3.TestPolynome vous permet de tester votre code.