

Cours de programmation orientée-objet

Examen du 21 août 2014

Livres fermés. Durée : 3 heures 1/2.

Veillez répondre à chaque question sur des feuilles séparées sur lesquelles figurent nom, prénom et section. Soyez bref et concis, mais précis.

1. Dans le cadre du développement d'un traitement de texte, on souhaite créer deux classes `WordCounter` et `SentenceCounter` capables de compter respectivement le nombre de mots et le nombre de phrases dans un texte. Un mot est défini comme une suite de lettres consécutives, ne comprenant ni espace ni symbole de ponctuation. Une phrase est une suite de caractères commençant par une majuscule et se terminant par un point ("."), un point d'interrogation (" ? ") ou un point d'exclamation (" ! ").

A titre d'exemple, l'analyse du texte *"Programming is awesome! Isn't it?"* doit retourner 6 mots et 2 phrases.

Chacune des deux classes `WordCounter` et `SentenceCounter` doit respecter l'interface suivante :

- Leur instanciation s'effectue sans paramètre, et produit un objet représentant un compteur initialisé à zéro.
- Un texte peut être transmis à une instance d'une de ces classes en invoquant successivement une méthode `public void putChar(char ch)` pour chacun des caractères `ch` composant ce texte.
- A chaque instant, la valeur courante d'un compteur peut être consultée grâce à une méthode `public int getCounterValue()`.

- (a) Programmer ces deux classes en Java.
- (b) Programmer une classe de démonstration supplémentaire capable d'analyser la phrase donnée en exemple dans l'énoncé.

Notes :

- Vous êtes libres de développer des classes supplémentaires nécessaires à votre solution.
- Les situations d'erreur éventuelles doivent être signalées par des exceptions implémentées par vos soins. On **ne** demande **pas** de détecter les erreurs syntaxiques dans le texte, comme par exemple la présence de symboles autres que des lettres, des espaces ou de la ponctuation, l'absence d'un point final, ...
- Toutes les classes implémentées doivent être placées dans le groupe de classes `oop.lexico`.
- La conformité aux mécanismes d'équivalence et de clonage **n'est pas** demandée.
- Une partie de la documentation de la bibliothèque standard Java pouvant être utile à la résolution de ce problème est fournie en fin d'énoncé.

2. (a) Décrivez deux applications fréquentes (au choix) du mécanisme d'héritage, et précisez dans quelles circonstances ces applications sont pertinentes. Accompagnez chacune de vos descriptions d'un exemple simple.
(b) Donnez deux procédés différents de création de contexte en Java, et expliquez pour quelle raison ces deux procédés ont été définis.
3. Les deux extraits de code ci-dessous définissent une classe `Pair` capable de représenter de manière générique une paire d'éléments :

```
public abstract class AbstractPair<E>
{
    public abstract E get(boolean first);
    public abstract void set(E value, boolean first);
}
```

```
public class Pair<E> extends AbstractPair<E>
{
    private E v1, v2;

    public E get(boolean first)
    {
        if (first)
            return v1;
        else
            return v2;
    }

    public void set(E value, boolean first)
    {
        if (first)
            v1 = value;
        else
            v2 = value;
    }
}
```

Modifier ces classes de façon à permettre la comparaison d'objets instanciés à partir de la classe `Pair`.

Note : Il doit être possible de comparer deux paires même si le type utilisé lors de leur instantiation diffère.

Documentation Java

Classe `Character` : encapsulation et services relatifs à une valeur de type `char`

- `public static boolean isLowerCase(char ch)` : retourne un booléen indiquant si le caractère `ch` est minuscule.
- `public static boolean isUpperCase(char ch)` : retourne un booléen indiquant si le caractère `ch` est majuscule.
- `public static boolean isLetter(char ch)` : retourne un booléen indiquant si le caractère `ch` est une lettre.
- `public static boolean isWhitespace(char ch)` : retourne un booléen indiquant si le caractère `ch` est un espace.

Classe `String` : représentation standard d'une chaîne de caractères

- `public int length()` : retourne la taille de la chaîne de caractères.
- `public char charAt(int index)` : retourne le caractère présent à l'indice `index` dans la chaîne de caractères. Les indices prennent leurs valeurs dans l'intervalle $[0, \text{length}() - 1]$.