

1 Révision du cours

Remplir les trous du texte suivant en utilisant les mots suivants (un même mot pourra être utilisé plusieurs fois) : classes, instances, objets, attributs, constructeurs, méthodes.

- Les objets sont les _____ des _____.
- Les classes définissent les _____, les _____ et les _____ de leurs instances.
- Les _____ servent à fabriquer les _____ des _____.
- L'état des _____ dépend des valeurs de leurs _____.

2 Vocabulaire

Parmi les termes suivants, grouper ceux ayant un sens proche ou qui sont synonymes : classe, paramètre, propriété, objet, méthode, constructeur, champ, instance, attribut, argument.

3 Manipulation de chaînes de caractères

Voici un extrait de l'interface de la classe `String` :

```
// Returns the char value at the specified index.  
char charAt(int index)  
// Compares two strings lexicographically.  
int compareTo(String anotherString)  
// Concatenates the specified string to the end of this string.  
String concat(String str)  
// Tests if this string ends with the specified suffix.  
boolean endsWith(String suffix)  
// Returns the index within this string of the first occurrence of the  
// specified character.  
int indexOf(int ch)  
// Returns true if, and only if, length() is 0.  
boolean isEmpty()  
// Returns the length of this string.  
int length()  
// Returns a string resulting from replacing all occurrences of oldChar  
// in this string with newChar.  
String replace(char oldChar, char newChar)  
// Returns a string that is a substring of this string.  
String substring(int beginIndex, int endIndex)  
// Converts all of the characters in this String to lower case using  
// the rules of the default locale.  
String toLowerCase()  
// Converts all of the characters in this String to upper case using  
// the rules of the default locale.  
String toUpperCase()
```

```
// Returns a string whose value is this string, with any leading and
// trailing whitespace removed.
String trim()
```

3.1 Repérage

Repérer les noms des méthodes, les types des paramètres, et le type de retour de chaque méthode.

3.2 Utilisation de la classe String

On se place dans le corps d'une méthode possédant une variable `String text` initialisée, comme ceci :

```
public void workWithString() {
    String text = " Hello World! I like programming in Java. ";
    ... // TODO
}
```

1. Quelle instruction doit-on utiliser pour passer la chaîne en majuscule ?
2. Comment supprimer les espaces en début et fin de texte ?
3. On souhaite enlever les 10 premiers caractères et les déplacer à la fin. Comment réaliser cette opération ?
4. Quelles instructions doit-on utiliser pour obtenir le premier mot du texte ? On se repère par rapport à la première espace suivant un caractère non blanc, par exemple le premier mot de " Hello! World! " doit être "Hello!".
5. On souhaite remplacer les 'o' par des '0', les 'e' par des '3', let 't' par des '7'. Comment le faire ? Utiliser des méthodes auxiliaires pour éviter de vous répéter.

4 Date

```
public class Date{
    private final int year;
    private final int month;
    private final int dayInMonth;

    public Date(int year, int month, int dayInMonth){
        this.dayInMonth = dayInMonth;
        this.year = year;
        this.month = month;
    }

    int getDayInMonth(){
        return this.dayInMonth;
    }

    int getYear(){
        return this.year;
    }

    int getMonth(){
        return this.month;
    }
}
```

1. Quels sont les attributs de la classe `Date` ?
2. Quelles sont les méthodes de la classe `Date` ?
3. Est-ce que la classe `Date` a un constructeur ?

5 Les articles du supermarché

1. Écrivez une classe `Article` pour représenter les articles vendus dans un supermarché. Chacun comporte quatre attributs : un numéro de référence unique, une description textuelle de l'article, un prix hors taxe, et une quantité disponible en stock.
2. Ajouter un constructeur initialisant les propriétés.
3. Ajouter une méthode `void provision(int quantity)` permettant d'approvisionner les stocks de cet article.
4. Ajouter une méthode `boolean sell(int quantity)` permettant de vendre l'article. Si l'article n'a pas suffisamment de disponibilité, la vente est annulée et `false` est retourné.
5. Ajouter une méthode `double price()` retournant le prix TTC de l'article (appliquer une TVA de 21%).

6 Point

1. Écrivez une classe `Point` qui représente un point dans le plan en coordonnées cartésiennes. Cette classe devra avoir deux attributs de type `double` nommés `x` et `y`.
2. Ajoutez à la classe `Point` un constructeur afin de permettre l'initialisation de `x` et `y`.
3. Ajouter à la classe `Point` une méthode `double distanceTo(Point p)` qui retourne la distance du point au point `p`. Utiliser `double Math.sqrt(double d)` pour calculer la racine carrée.
4. Ajouter à la classe `Point` une méthode `Point translate(double dx, double dy)` qui retourne un point obtenu par translation du point de `dx` sur les abscisses et de `dy` sur les ordonnées.
5. Écrivez une classe `LineSegment` qui possède deux propriétés `endPoint1` et `endPoint2` de type `Point` représentant les deux extrémités d'un segment.
6. Ajoutez à la classe `LineSegment` un constructeur afin de permettre l'initialisation de `point1` et `point2`.
7. Ajouter à la classe `LineSegment` une méthode `double length()` qui calcule et retourne la longueur du segment. Est-ce qu'une partie du calcul peut être déléguée à une méthode de la classe `point` ?