

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 Date

```
public class Date{
    int year;
    int month;
    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;
    }
}
```

Question 1 : Quels sont les attributs de la classe Date ?

Question 2 : Quelles sont les méthodes de la classe Date ?

Question 3 : Est-ce que la classe Date a un constructeur ?

Question 4 : Donner le code d'une méthode `toString()` qui renvoie une chaîne de caractères au format `dd/mm/yyyy`.

Question 5 : Donner le code d'une méthode `before(Date when)` qui renvoie `true` si la date avec laquelle est appelée la méthode est avant (ou en même temps) que la date `when`.

Question 6 : Peut-on créer une date telle que l'appel à `toString()` renvoie `99/99/9999` ?

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 from
// beginIndex (inclusive) to endIndex (exclusive).
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 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  
}
```

Question 7 : Quelle instruction doit-on utiliser pour que `text` contienne une chaîne de caractères avec les mêmes caractères mais en majuscule ?

Question 8 : Quelle instruction doit-on utiliser pour que `text` contienne une chaîne de caractères sans les espaces en début et fin de texte ?

Question 9 : On souhaite enlever les 10 premiers caractères et les déplacer à la fin. Quelles instructions doit-on utiliser pour que `text` contienne cette nouvelle chaîne de caractères ?

Question 10 : On souhaite créer une nouvelle chaîne de caractères avec des 'o' par des '0' et les 'e' par des '3'. Quelles instructions doit-on utiliser pour que `text` contienne cette nouvelle chaîne de caractères ?

Question 11 : Quelles instructions doit-on utiliser pour obtenir le premier mot du texte ? On se repère par rapport au premier espace suivant un caractère non blanc, par exemple le premier mot de " Hello! World!" doit être "Hello!".

3.2 Méthodes sur les String (bonus)

Question 12 : Donnez le code d'une fonction `prefixEqualsSuffix(String s, int n)` qui renvoie `true` si le préfixe de `s` de longueur `n` (la sous-chaîne composée des `n` premiers caractères de `s`) est égal au suffixe de `s` de longueur `n` (la sous-chaîne composée des `n` derniers caractères de `s`).

Question 13 : Donnez le code d'une fonction `boolean equalsIgnoreCase(String s1, String s2)` qui renvoie `true` si `s1` et `s2` ont les mêmes caractères en ne distinguant pas les majuscules et les minuscules ("tOt0" et "T0t0" sont par exemple considérées comme étant égales).

Question 14 : Donnez le code d'une fonction `int lastIndexOf(String s, char ch)` qui renvoie l'indice de la dernière occurrence de `ch` dans `s` (renvoie `-1` si `ch` n'est pas présent dans `s`).

Question 15 : Donnez le code d'une fonction `String repeat(String s, int count)` qui renvoie la chaîne de caractères composé de la concatenation de `s` répétée `count` fois (par exemple, `String repeat("to",3)` renvoie `"tototo"`).