II.1102 - Algorithmique et Programmation
Examen écrit

Guillaume Lachaud

27 janvier 2022

1 Consignes générales

- Ce partiel est noté sur *42*.
- La durée de cet examen est de *3 heures*.
- Prenez le temps de lire l'énoncé dans son intégralité et assurez-vous que le sujet comporte bien *7 pages*.
- Le sujet est à rendre avec la copie.
- Aucun document n'est autorisé pour cet examen.
- Attention à ne pas confondre *afficher* et *retourner*.
- Veuillez faire attention à la syntaxe utilisée en Java. Quelques erreurs pourront être tolérées mais un oubli systématique des règles de syntaxe sera pénalisé.

2 Questions de cours (6 points)

1. Quels sont les types primitifs ?
2. Quelles sont les structures conditionnelles en Java ?
3. Quelles sont les structures itératives en Java ?
4. Une fonction est-elle obligée d'afficher ou de retourner une valeur ? Peut-elle retourner plusieurs valeurs ?
5. Qu'est-ce qu'un objet ? Qu'est-ce qu'une classe ?
6. Quelle fonction permet d'instancier un objet ?

3 Questions de compréhension (6 points)

3.1 Question 1 (3 points)

```java
public class Main {
    public static void main(String[] args) {
        int i = 888;
        System.out.println(i);
        fonctionA(i);
        System.out.println(i);
        int[] tableau = {4, 8, 15, 16, 23, 42};
        System.out.println(tableau[5]);
        fonctionB(tableau);
        System.out.println(tableau[5]);
    }
    public static void fonctionA(int i){
        if (i % 3 == 0){
            i = i / 3;
            System.out.printf("Multiple de 3 : %d%n", i);
        } else {
            int residu = i % 3;
            i = (i - residu) / 3;
            System.out.printf("Quotient par 3 : %d. Résidu : %d%n", i, residu);
        }
    }
    public static void fonctionB(int[] tableau){
        for (int i = 0; i < tableau.length; i++){
            if (tableau[i] % 7 == 0)
                tableau[i] /= 7;
        }
    }
}
```

Listing 1: Code mystère

Question: Quel est le résultat affiché en console par ce programme ?

3.2 Question 2 (3 points)

```java
public class Main {
    public static void main(String[] args) {
        int n = 12;
        System.out.println(n);
        int c = 1000;
        while (c > 0 && n != 1){
            n = f(n);
            System.out.println(n);
            c--;
        }
    }

    public static int f(int n) {
        if (n % 2 == 0)
            return n / 2;
        else
            return 3 * n + 1;
    }
}
```

Listing 2: Code mystère

Questions:

1. Quel est le résultat affiché en console par ce programme ?
2. *Bonus:* Que cherche à faire ce programme ? Reconnaissez-vous ce problème ?

4 Correction de code (8 points)

4.1 Traversée d'un tableau (3 points)

Le méthode `traversee` parcourt un tableau et affiche la parité de chacun de ses éléments.

```java
public static void traversee(int[] tableau){
for(int i = 0; i <= tableau.length; i++){
    if (tableau[i] % 2 = 0) {
        System.out.printf("Le %d-eme element est pair");
    } else {
        System.out.printf("Le %d-eme element est impair");
    }
}
```

Listing 3: Traversée d'un tableau et affichage de la parité de ses éléments

Question: Corrigez cette méthode et recopiez sa version corrigée sur votre copie.

4.2 La suite de Fibonacci (5 points)

La suite de Fibonacci est la suite de nombres définie par la relation

$$\begin{cases} F_0 = 0, F_1 = 1 \\ F_{n+2} = F_n + F_{n+1} \quad n \geq 2 \end{cases} \tag{1}$$

La méthode `fibonacci` calcule et retourne un élément de la suite.

```java
public static int fibonacci(int n){
    return fibonacci(n-2) + fibonacci(n-1);
}
```

Listing 4: Suite de Fibonacci en Java

Questions:

1. Corrigez cette méthode et recopiez sa version corrigée.

2. Que se passe-t-il lorsque l'on essaye de calculer `fibonacci(100)` ? Proposez une modification pour régler le problème et réécrivez le code en incorporant votre modification.

3. Indiquez la complexité de votre méthode modifiée.

5 Conception et programmation (14 points)

5.1 Les premiers jumeaux (6 points)

Deux nombres premiers p_1 et p_2 sont dit *jumeaux* lorsqu'ils ne diffèrent que de 2.

Question: Écrivez la méthode `premiersJumeaux` qui prend en entrée deux nombres entiers *quelconques* et retourne un booléen indiquant si les deux nombres sont des nombres premiers jumeaux.

5.2 Trouvez la somme (8 points)

Étant donné un tableau d'entiers et un nombre donné, on cherche à savoir si ce nombre peut être obtenu en additionnant des éléments du tableau.

Questions:

1. Écrivez la méthode `trouverSomme` qui prend en argument un tableau d'entiers et un entier, et qui retourne une `List` contenant les éléments à sommer pour obtenir l'entier fourni. S'il est impossible de sommer des éléments du tableau de manière à obtenir cet entier, la liste retournée est vide.

2. Indiquez la complexité de votre code.

3. On considère à présent que la liste fournie est *triée*. Pouvez-vous améliorer votre code afin de diminuer la complexité ? Si oui, écrivez la méthode modifiée et indiquez sa complexité.

6 Modélisation (8 points)

Dans cet exercice, on souhaite modéliser une version simplifiée du réseau des Bibliothèques de Paris. Ce réseau comprend des bibliothèques qui sont identifiées par un

code postal et un nom. Les utilisateurs des bibliothèques sont identifiés par un numéro unique et une adresse email.

Dans chaque bibliothèque, on peut réserver différents types de ressources : des livres, des CDs et des DVDs. Ces ressources sont à leur tour réparties en catégories : fantasy, science-fiction, comédie romantique, etc. Chaque bibliothèque contient une liste de ressources, ainsi qu'un historique des emprunts avec la date de l'emprunt et celle du rendu.

Un utilisateur peut emprunter des ressources dans plusieurs bibliothèques. Cependant, le nombre de ressources empruntées à un moment donné est limité et s'applique à l'ensemble du réseau.

Questions:

1. Représentez une classe, une classe abstraite et une interface dans un diagramme UML de classe.

2. Rappelez la différence entre une classe et une classe abstraite.

3. Rappelez la différence entre une classe abstraite et une interface.

4. Rappelez les symboles utilisés pour représenter les différents types de visibilité de la programmation orientée objet dans un diagramme UML de classe.

5. Rappelez les types de flèches utilisées pour représenter l'héritage et l'implémentation dans un diagramme UML de classe.

6. Representez le diagramme UML de classe permettant de modéliser cette version simplifiée du réseau des bibliothèques de Paris.