

Introduction à la Programmation Objet, TD1

1 Questions de cours

Soit le code suivant :

```
1 public class User {
2     private int age;
3     private String username;
4     private String role;
5
6     public User(int age, String username, String role) {
7         this.age = age;
8         this.username = username;
9         this.role = role;
10    }
11
12    public User(int age, String username) {
13        this(age, username, "suscriber");
14    }
15
16    public int getAge() {
17        return age;
18    }
19
20    public void setAge(int age) {
21        this.age = age;
22    }
23
24    public String getUsername() {
25        return username;
26    }
27
28    public boolean isAdult() {
29        return age >= 18;
30    }
31
32 }
```

Question 1 Désigner par leur numéros de lignes les parties du code correspondant à la déclaration d'attributs, aux constructeurs et aux méthodes.

Question 2 Quel est le nom de la classe? Le nom complet du fichier?

Question 3 Que signifie le mot-clé `private` à la ligne 2?

Question 4 Écrire une instruction permettant de créer une instance de cette classe.

Question 5 Écrire une instruction permettant d'en changer l'âge à 25.

Question 6 Que signifie le mot-clé `this` à la ligne 7?

Question 7 Que signifie le mot-clé `this` à la ligne 13?

1.1 Convention de code

Question 8 On vous demande d'écrire un programme gérant un jeu se déroulant en plusieurs manches. Parmi ceux-ci, quel nom de variable choisir pour désigner le score d'un joueur pour une manche ? Pourquoi ?

1. `score_d_une_manche`
2. `nb`
3. `currentpoints`
4. `scorePlayerRound`
5. `MonScoreCetteManche`

1.2 Debug

À l'exécution d'un programme, l'exception suivante est levée :

```
1 Exception in thread "main" java.lang.NullPointerException
2   at cours.Mail.send(Mail.java:7)
3   at cours.Mail.main(Mail.java:14)
```

Question 9 Quel est le soucis ? Que faudrait-il aller regarder dans le code/comment le corriger ?

1.3 static

Soit une classe `Equation` contenant la méthode :

```
public static Double solver(String equation, ArrayList<Double> parameters)
```

Question 10 Comment appeler cette méthode depuis une autre classe ?

2 Code, compréhension et ajouts

Nous allons ici gérer l'historique des scores pour un jeu de style arcade. A chaque partie, un nom de joueur est entré. À la fin de chaque partie, un score est enregistré, associé au nom du joueur. On veut ensuite pouvoir retrouver le meilleur score de chaque joueurs et les meilleur score général. Le contenu du jeu ne vous est pas demandé et n'a pas d'importance pour cet exercice, on a juste besoin de savoir que chaque partie aboutit à un score positif.

On possède pour cela de les squelettes de code suivant, ainsi que de la classe `Partie` :

Partie
public Partie(String joueur) public int jouer()

```
1 import java.util.ArrayList;
2 import java.util.HashMap;
3
4 public class Historique {
5     private HashMap<String, ArrayList<Integer>> historiqueScores;
6
7     public Historique(){
8         this.historiqueScores = new HashMap<>();
9     }
10
11     public void ajoutScore(String joueur, int score){
12         ArrayList<Integer> scoresJoueur = historiqueScores.get(joueur);
13         if (scoresJoueur == null){
```

```

14         scoresJoueur = new ArrayList<>();
15         scoresJoueur.add(score);
16         historiqueScores.put(joueur, scoresJoueur);
17     } else {
18         scoresJoueur.add(score);
19     }
20 }
21
22 private static Integer max(ArrayList<Integer> tab) {
23     if (tab == null || tab.isEmpty()) {
24         return null;
25     }
26     Integer max = tab.get(0);
27     for (Integer i : tab) {
28         if (i > max) {
29             max = i;
30         }
31     }
32     return max;
33 }
34 }

```

```

1 import java.util.Scanner;
2
3 public class Lanceur {
4
5     public static String entrerNom(Scanner sc){
6         System.out.println("Entrez_votre_nom");
7         return sc.nextLine();
8     }
9
10    public static void main(String [] args) {
11        Scanner sc = new Scanner(System.in);
12
13        //TODO Question 20
14    }
15
16 }

```

Question 11 Ligne 5 de Historique.java, pourquoi le type `Integer` est-il utilisé dans la structure de `historiqueScores` plutôt que le type `int` ?

Question 12 Que représente l'attribut `historiqueScores` défini ligne 5 de Historique.java ? Quelles données contiendra-t-il ?

Question 13 Décrire dans quelles situations chacun des cas du `if` (ligne 13 et 17 de Historique.java) se produiront.

Question 14 Ligne 22 de Historique.java, que signifie le mot-clé `static` ?

Question 15 Ligne 22 de Historique.java, que signifie le mot-clé `private` ?

Question 16 À quoi sert la méthode `max` de Historique.java (lignes 22 à 33) ?

Question 17 Dans quel cas la méthode `max` de `Historique.java` pourrait-elle lancer une exception ? Cela peut-il se produire pour des données issues de notre historique de score, si l'on suppose que l'on y ajoute des données uniquement par la méthode `ajoutScore` ?

Question 18 Écrire une méthode `public Integer trouverHighScore(String joueur)`, dans la classe `Historique`, qui prend le nom d'un joueur et renvoie son meilleur score.

Question 19 Écrire une méthode `public ArrayList<String> getListeJoueurs()` dans la classe `Historique`, qui renvoie la liste des noms de tous les joueurs ayant déjà enregistré un score.

Indication : Vous pouvez utiliser la méthode `Set<K> keySet()` de `HashMap<K,V>`

Question 20 Écrire une méthode `HashMap<String, Integer> highScoreParJoueur()` dans la classe `Historique`, qui renvoie une `HashMap` associant à chaque joueur son meilleur score.

Question 21 Écrire une méthode `public Integer trouverHighScoreGlobal()` dans la classe `Historique`, qui renvoie le meilleur score parmi tous les joueurs.

Question 22 Dans la classe `Lanceur`, écrire un `main` qui permette via une boucle infinie d'entrer un nom, de lancer une partie, d'enregistrer le score obtenu puis d'afficher le meilleur score du joueur et le meilleur score global.

Un rendu possible serait : (on suppose que c'est la méthode `jouer()` de `Partie` qui affiche le score de la partie à la fin de celle-ci)

```
Entrez votre nom
Alice
score de la partie :
298
Meilleur score du joueur : 298
Meilleur score global : 298
Entrez votre nom
Alice
score de la partie :
15
Meilleur score du joueur : 298
Meilleur score global : 298
Entrez votre nom
Lucas
score de la partie :
3450
Meilleur score du joueur : 3450
Meilleur score global : 3450
Entrez votre nom
Alice
score de la partie :
830
Meilleur score du joueur : 830
Meilleur score global : 3450
Entrez votre nom
```

Question 23 Proposer une solution pour ne pas avoir à recalculer le meilleur score global à chaque fois.