

IPO – TD 7 : préparation avant partiel

<https://www.lri.fr/~blsk/IPO/>

1 Questions de cours

Structure d'une classe 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 }
```

1. Désigner par leurs numéros de lignes les parties du code correspondant aux déclarations d'attributs, aux constructeurs et aux méthodes.
2. Quel est le nom de la classe? Le nom complet du fichier?
3. Que signifie le mot-clé **private** à la ligne 2?
4. Écrire une instruction permettant de créer une instance de cette classe.
5. Écrire une instruction permettant d'en changer l'âge à 25.
6. Que signifie le mot-clé **this** à la ligne 7?
7. Que signifie le mot-clé **this** à la ligne 13?

Convention de style

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?
 - score_d_une_manche
 - nb
 - currentpoints
 - scorePlayerRound
 - MonScoreCetteManche

Debug Lors du test d'un programme, le message suivant s'affiche :

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

9. Ce message est-il susceptible de se produire lors de la compilation, ou lors de l'exécution du programme compilé?
10. Quel est le souci? Que faudrait-il aller regarder dans le code et où? Comment le corriger?

Fonctions Soit une classe Equation contenant la méthode

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

11. Comment appeler cette méthode depuis une autre classe?

2 Compréhension de code et programmation

Nous allons ici gérer l'historique des scores pour un jeu de style arcade. À 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 joueur et le 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 se donne les trois squelettes de classes montrés en page annexe.

Questions

12. Ligne 5 de Historique.java, pourquoi le type Integer est-il utilisé dans la structure de historiqueScores plutôt que le type int?
13. Que représente l'attribut historiqueScores défini ligne 5 de Historique.java? Quelles données contiendra-t-il?
14. Décrire dans quelles situations chacun des cas du if (ligne 13 et 17 de Historique.java) se produiront.
15. Ligne 22 de Historique.java, que signifie le mot-clé static?
16. Ligne 22 de Historique.java, que signifie le mot-clé private?
17. À quoi sert la méthode max de Historique.java (lignes 22 à 33)?
18. 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 scores, si l'on suppose que l'on y ajoute des données uniquement par la méthode ajoutScore?
19. Écrire une méthode public Integer trouverHighScore(String joueur), dans la classe Historique, qui prend le nom d'un joueur et renvoie son meilleur score.
20. 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>.
21. Écrire une méthode HashMap<String, Integer> highScoreParJoueur() dans la classe Historique, qui renvoie une hashMap associant à chaque joueur son meilleur score.
22. Écrire une méthode public Integer trouverHighScoreGlobal() dans la classe Historique, qui renvoie le meilleur score parmi tous les joueurs.
23. Dans la classe Lanceur, écrire un main contenant un boucle infinie qui, à chaque tour : demande d'entrer un nom, lance une partie, enregistre le score, et enfin affiche le meilleur score du joueur et le meilleur score global.
24. Proposer une solution pour ne pas avoir à recalculer le meilleur score global à chaque fois.

```

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 23
14     }
15
16 }

```

```

1  public class Partie {
2
3      public Partie(String joueur) { ... }
4
5      public int jouer() { ... }
6
7  }

```