

Java Dragon

Sm!le42

24 juillet 2021

Table des matières

0.1	Règles	2
0.1.1	Aide autorisée	2
0.1.2	Temps imposé	2
0.1.3	Remise des exercices	2
0.1.4	Ce que tu dois payer	3
1	Base	3
1.1	Exercice 1	3
1.1.1	Objectif	3
1.1.2	Contraintes	3
1.1.3	Exemples	3
1.2	Exercice 2	3
1.2.1	Objectif	3
1.2.2	Exemples	3
1.2.3	Aide	3
1.2.4	Variante	4
2	Boucles (itérations)	4
2.1	Exercice 3	4
2.1.1	Objectif	4
2.1.2	Aide	4
2.1.3	Variantes	4
2.2	Exercice 4	4
2.2.1	Objectif	4
2.2.2	Exemples	4
2.2.3	Aide	5
2.3	Exercice 5	5
2.3.1	Objectif	5
2.3.2	Aide	5
2.3.3	Variantes	5
3	Lectures robustes	5
3.1	Exercice 6	5
3.1.1	Objectif	5
3.1.2	Exemple	5
3.1.3	Aide	5
4	Boss1	5
4.1	Trouve Moi Si Tu Peux 1.0	5
4.1.1	Introduction	5
4.1.2	Objectifs	5
4.1.3	Exemple	6
4.1.4	Aide	6
4.2	Trouve Moi Si Tu Peux 1.1	6
4.2.1	Objectif	6
4.2.2	Exemple	6
4.3	Trouve Moi Si Tu Peux 1.2	6
4.3.1	Objectif	6
5	Méthodes	7
5.1	Exercice 7	7

5.1.1	Objectif	7
5.1.2	Aide	7
5.2	Exercice 8	7
5.2.1	Objectif	7
5.2.2	Aide	7
5.3	Exercice 9	7
5.3.1	Objectif	7
5.4	Exercice 10	7
5.4.1	Objectif	7
5.4.2	Aide	7
6	Boss2	7
6.1	Trouve Moi Si Tu Peux 1.3	7
6.1.1	Objectif	7
7	Tableaux	7
7.1	Exercice 11	7
7.1.1	Objectif	7
7.1.2	Exemple	8
7.1.3	Aide	8
7.2	Exercice 12	9
7.2.1	Objectif	9
7.2.2	Exemple	9
7.3	Exercice 13	9
7.3.1	Objectif	9
7.3.2	Exemple	9
7.3.3	Aide	10
8	Solutions	10
8.1	Exercice 1	10
8.2	Exercice 2	10
8.2.1	Variante	11
8.3	Exercice 3	11
8.3.1	Variante 1	11
8.3.2	Variante 2	12

Exercices de Java pour le Dragon. (Bonne chance mon reuf)

0.1 Règles

0.1.1 Aide autorisée

Tu as l'**obligation** de t'aider d'Internet.

Utilise les forums du genre *stackoverflow*, *commentcamarche*, *openclassrooms* ou même tout simplement *Google*.

C'est la bible des programmeurs. ;)

(Tu peux me demander de l'aide aussi bien sûr)

0.1.2 Temps imposé

Il n'y a **pas** de temps imposé mais je te conseille de ne pas traîner car il y a beaucoup de boulot. . .

Au travail chacal !

0.1.3 Remise des exercices

Tu me remettras les exercices en utilisant **Github**.

Tu devras donc :

1. Te créer un compte sur Github (si ce n'est pas déjà fait)
2. Créer un repository qui s'appellera JavaDragonReponse
3. Me transmettre le lien du repo
4. Y déposer tes réponses petit à petit
5. Me prévenir quand tu as déposé une nouvelle réponse

Remarque : Toutes les réponses doivent être dans le repo Java Dragon, et ce serait sympa pour moi que tout soit facile à trouver (genre faire des dossiers ex1, ex2, ex3 etc. . . t'as pigé), donc si tu pouvais tout hiérarchiser de manière intuitive (histoire que je cherche pas pendant des heures blbl) ce serait cool.

0.1.4 Ce que tu dois payer

Prix de cet entraînement intensif de Java : **420€ TTC**.

(Ou alors juste une p'tite Jup' à la fin c'est bon aussi tkt bg)

1 Base

1.1 Exercice 1

1.1.1 Objectif

Le logiciel devra effectuer ces tâches :

1. Demander mon pseudo
2. Demander mon âge
3. Me répondre "Bonjour @PSEUDO, tu es @MAJEUR"

Bien sur, @PSEUDO devra être remplacé par le pseudo entré par l'utilisateur, et @MAJEUR devra afficher mineur si l'utilisateur a entré un âge plus petit que 18.

1.1.2 Contraintes

- Interdiction d'utiliser Netbeans
- Interdiction d'utiliser des packages
- Une seule méthode est permise (main)

1.1.3 Exemples

```
@PSEUDO : "Antho" @AGE : 23 Affiche : "Bonjour Antho, tu es majeur!"
@PSEUDO : "42" @AGE : 16 Affiche : "Bonjour 42, tu es mineur!"
@PSEUDO : "" @AGE : -30 Affiche : "Bonjour , tu es mineur!"
@PSEUDO : "Test" @AGE : "blbl" "ERREUR (IllegalArgumentException)"
```

1.2 Exercice 2

1.2.1 Objectif

Même chose que l'exercice 1, mais cette fois-ci affiche un message d'erreur :

- Si l'utilisateur n'entre pas de pseudo (c'est à dire @PSEUDO -> "")
- Si l'utilisateur entre un âge invalide (@AGE<0 ou @AGE>120)

1.2.2 Exemples

```
@PSEUDO : "" @AGE : 18 Affiche : "Pseudo invalide"
@PSEUDO : "Jean" @AGE : -30 Affiche : "Tu n'es pas encore né!"
@PSEUDO : "Raoul" @AGE : 142 Affiche : "Tu es sans doute décédé..."
```

1.2.3 Aide

```
1 int nombre;
2 String mot;
3
4 nombre == 42;           //ok
5 mot.equals("Test");    //ok
6 //mot == "Test"; NE MARCHERA PAS CORRECTEMENT car String est un Objet
```

1.2.4 Variante

Si l'utilisateur se trompe pour le pseudo **ET** l'âge, on pourrait avoir une nouvelle possibilité.
Imaginons que je ne tape pas de pseudo, et que j'entre 142 pour l'âge, alors on pourrait afficher :
"T'es bourré ou quoi?"

2 Boucles (itérations)

2.1 Exercice 3

2.1.1 Objectif

Même chose que l'exercice 1, sauf que cette fois, on va effectuer une pseudo lecture robuste.

Le programme devra :

1. Demander mon pseudo
2. Redemander mon pseudo tant que j'ai entré un pseudo vide
3. Demander mon âge
4. Redemander mon âge tant que je n'ai pas entré un âge valide
5. Afficher "Bonjour @PSEUDO, tu es @MAJEUR"

(Bien sûr, afficher mineur si âge < 18)

2.1.2 Aide

Ne pas utiliser de boucle for !

2.1.3 Variantes

(Attention ces variantes sont un peu plus compliquées. Il va falloir utiliser des compteurs)

1. On pourrait envisager que si l'utilisateur entre 3 fois une donnée invalide, on arrête le programme en affichant un message du genre "Tu te fous de moi? Allez ciao."
2. Ou alors on pourrait décider que si l'utilisateur se trompe 3 fois de pseudo on lui attribue automatiquement le pseudo "invité", et s'il se trompe 3 fois d'âge on lui attribue automatiquement l'âge 0.

Astuce : La commande pour stopper un programme en Java est : `System.exit(0);`

2.2 Exercice 4

Allez chacal ! Courage :)

2.2.1 Objectif

Créer un logiciel qui :

1. Demande un nombre à l'utilisateur
2. Redemande ce nombre jusqu'à ce qu'il entre 0

2.2.2 Exemples

```
1 //Exemple 1
2 Entre un nombre: 14
3 Entre un nombre: 7
4 Entre un nombre: 21
5 Entre un nombre: 0
6 Au revoir!
7
8 //Exemple 2
9 Entre un nombre: 0
10 Au revoir!
11
12 //Exemple 3
13 Entre un nombre: blbl
14 *ERROR IllegalArgumentException*
```

2.2.3 Aide

Il faudra demander le nombre tant que... ;)

2.3 Exercice 5

Je te conseille vivement de refaire tous les exercices à partir du début. (Sans aide)

2.3.1 Objectif

Créer un simple logiciel qui affiche les nombres de 0 jusque 420.

2.3.2 Aide

On sait d'avance combien de fois la boucle va être exécutée.

2.3.3 Variantes

1. Affiche uniquement les nombres pairs
2. Affiche uniquement les multiples de 3 et 5 (pas 6, 15 ok, pas 25)
3. Affiche les nombre 7 par 7 (0, 7, 14, 21, 28, ..., 420)

3 Lectures robustes

3.1 Exercice 6

3.1.1 Objectif

Refaire l'exercice 4, mais en vérifiant que l'utilisateur entre bien un nombre entier.

3.1.2 Exemple

```
1  Entre 0 pour quitter: 14
2  Entre 0 pour quitter: 42
3  Entre 0 pour quitter: blbl
4  Ce n'est pas un nombre! Recommence: Karamel
5  Ce n'est pas un nombre! Recommence: 7
6  Entre 0 pour quitter: 0
7  Au revoir!
```

3.1.3 Aide

- Tant que n'est pas un entier -> `while(!kb.hasNextInt())`
- Jète la valeur du Scanner -> `kb.next();`
- Recommence
- Après la boucle, Récupère la valeur du Scanner -> `int x = kb.nextInt();`

4 Boss1

4.1 Trouve Moi Si Tu Peux 1.0

4.1.1 Introduction

Bravo ! Tu as appris les bases, les boucles et les lectures robustes.
Maintenant on va créer un petit jeu.

4.1.2 Objectifs

1. Demander le niveau à l'utilisateur (entre 5 et 100)
2. Choisir un nombre au hasard entre 1 et le niveau (ne pas montrer bien sûr)
3. Demander un nombre à l'utilisateur jusqu'à ce qu'il trouve le nombre caché aléatoire
4. Si l'utilisateur se trompe, lui préciser si c'est trop petit ou trop grand

5. Lorsque l'utilisateur a trouvé le nombre caché, le féliciter
(On veut des lectures robustes !)

4.1.3 Exemple

```
1 Niveau? (5 à 10): 2
2 Invalide. Entre un niveau entre 5 et 100: 101
3 Invalide. Entre un niveau entre 5 et 100: blbl
4 Invalide. Entre un niveau entre 5 et 100: 10
5 Ok, essaie de trouver le nombre aléatoire entre 1 et 10: 4
6 Trop petit! Réessaie: -3
7 Invalide. Entre un nombre entre 1 et 10: blbl
8 Invalide. Entre un nombre entre 1 et 10: 8
9 Trop grand! Réessaie: 7
10 C'était ça! Bravo! Au revoir
```

4.1.4 Aide

Pour trouver un nombre aléatoire entre 1 et 100 :

```
1 (int) ((Math.random() * 100) +1)
```

- `Math.random()` (retourne un double entre 0 et 1 non inclus (donc de 0 à 0.9999999...))
- `*100` (multiplie par 100 et sera donc entre 0 et 99.9999999...)
- `+1` (ajoute 1 et sera donc entre 1 et 100.9999999...)
- `(int)` (cast le double en entier et sera donc entre 1 et 100! parfait :D)

4.2 Trouve Moi Si Tu Peux 1.1

4.2.1 Objectif

On veut laisser 3 chances à l'utilisateur pour trouver le nombre caché.
S'il se trompe 3 fois, il perd le jeu !

4.2.2 Exemple

```
1 Bonjour, entrez le niveau (5 à 100): 21
2 Essayez de trouver le nombre entre 1 et 21 (3 essais restants): 5
3 Trop petit! Réessaie (2 essais restants): 7
4 Trop petit! Réessaie (1 essai restant): 9
5 Perdu! C'était 14. Au revoir.
```

4.3 Trouve Moi Si Tu Peux 1.2

4.3.1 Objectif

Dans la version 1.1, on a codé le jeu de telle sorte que l'utilisateur a 3 chances pour trouver le nombre sinon il perd.

Ce n'est pas très correcte, car s'il choisit le niveau 5 il aura 3 chances pour trouver un chiffre entre 1 et 5, et s'il choisit le niveau 100 il aura 3 chances pour trouver un nombre entre 1 et 100 !

On veut donc que si le niveau choisi est :

- entre 5 et 10 : il a 2 chances
- entre 11 et 30 : il a 3 chances
- entre 31 et 60 : il a 4 chances
- entre 61 et 100 : il a 5 chances

(On changera ces nombres par la suite si on constate que c'est trop simple ou trop compliqué)

5 Méthodes

5.1 Exercice 7

5.1.1 Objectif

Refaire l'exercice 1 en utilisant les méthodes :

- demanderPseudo()
- demanderAge()
- repondre()

5.1.2 Aide

- String demanderPseudo()
- int demanderAge()
- void repondre(String pseudo, int age)

5.2 Exercice 8

5.2.1 Objectif

Refaire l'exercice 2 en utilisant les méthodes de l'exercice 7 et :

- checkPseudo()
- checkAge()

5.2.2 Aide

- boolean checkPseudo(String pseudo)
- boolean checkAge(int age)

5.3 Exercice 9

5.3.1 Objectif

Refaire l'exercice 3 en utilisant des méthodes.

5.4 Exercice 10

5.4.1 Objectif

Refaire l'exercice 4 en utilisant les méthodes :

- askNumber()
- isNull()

5.4.2 Aide

- int askNumber()
- boolean isNull(int nbr)

6 Boss2

6.1 Trouve Moi Si Tu Peux 1.3

6.1.1 Objectif

Recoder la version 1.2 en utilisant un maximum de méthodes !

(Essaie d'optimiser un maximum et je t'aiderai à améliorer ce que tu as fait)

7 Tableaux

7.1 Exercice 11

7.1.1 Objectif

- Demander à l'utilisateur un entier entre 3 et 9 (lecture robuste)
- Créer un tableau de la taille que l'utilisateur a entré

- Y placer les chiffres de 0 à ... (chiffre que l'utilisateur a entré)
- Afficher la taille du tableau
- Afficher les valeurs du tableau

7.1.2 Exemple

```

1  Entrez un chiffre entre 3 et 9:
2  2
3  Entre 3 et 9 svp:
4  blbl
5  Un entier valide svp:
6  7
7  Taille du tableau: 7
8  0
9  1
10 2
11 3
12 4
13 5
14 6
15 Fin du programme!

```

7.1.3 Aide

- Un tableau est un type **objet**. Il faudra donc utiliser le mot clé **new** pour créer une nouvelle instance.
- La taille d'un tableau ne pourra **pas** changer.
- Un tableau de taille `tab[7]` ira de `tab[0]` à `tab[6]` car on commence à 0 en programmation.
- On peut connaître la taille d'un tableau grâce à `tab.length`.

Attention! ne pas confondre :

1. `".length()"` pour connaître la taille d'un String
2. `".length"` pour connaître la taille d'un tableau

```

1  int variable           //Est une variable de type entier
2  int[] tableau1D       //Est un tableau à une dimension avec des valeurs de type entier
3  int[] [] tableau2D    //Est un tableau à deux dimensions avec des valeurs de type entier
4
5  //Il existe plusieurs façons de créer un tableau:
6
7  //1 Si on ne connaît pas encore les valeurs à mettre dans le tableau:
8  int tab1[];           //On prévient Java qu'on va créer un tableau "tab1"
9  tab1 = new int[7];    //On créé le tableau de taille 7 (valeurs non définies)
10 //2 On peut faire la même chose en une seule ligne:
11 int[] tab1 = new int[7];
12
13 //3 Si on connaît les valeurs à mettre dans le tableau:
14 int tab3[];
15 tab3 = new int[] {0, 1, 2, 3, 4, 5, 6};
16 //4 On peut faire la même chose en une seule ligne:
17 int tab4[] = new int[] {0, 1, 2, 3, 4, 5, 6};
18 //5 Si on déclare un tableau AVEC les valeurs et EN UNE SEULE LIGNE
19 //Alors on n'est pas obligé de mettre "new int[]"
20 int tab5[] = {0, 1, 2, 3, 4, 5, 6}; //Ne fonctionne qu'en une seule ligne!
21
22 //Exemple taille de tableau
23 int[] tab6 = new int[7];
24 System.out.println("Taille de tab6: " + tab6.length);
25
26 //Exemple de parcours d'un tableau 1D:
27 int[] tab6 = {2, 9, 9, 7, 9, 2, 4, 5, 8};
28 for (int i = 0, i < tab6.length; i++) {
29     System.out.println("tab6[" + i + "] = " + tab6[i]);

```



```

30 }
31 Ce code affichera:
32 tab6[0] = 2
33 tab6[1] = 9
34 tab6[2] = 9
35 tab6[3] = 7
36 tab6[4] = 9
37 tab6[5] = 2
38 tab6[6] = 4
39 tab6[7] = 5
40 tab6[8] = 8

```

7.2 Exercice 12

7.2.1 Objectif

- Demander à l'utilisateur un nombre entre 5 et 15
- Créer un tableau de la taille que l'utilisateur a entré
- Demander un nombre à l'utilisateur et le stocker dans le tableau
- Recommencer le nombre de fois que l'utilisateur a choisi au début du logiciel
- Une fois que tous les nombres ont été demandés, afficher le tableau

7.2.2 Exemple

```

1  Entrez un nombre entre 5 et 15:
2  2
3  Entre 5 et 15 svp:
4  blbl
5  Un entier valide svp:
6  5
7  1) Entrez un nombre:
8  7
9  2) Entrez un nombre:
10 14
11 3) Entrez un nombre:
12 21
13 4) Entrez un nombre:
14 28
15 5) Entrez un nombre:
16 35
17 Vous avez entre les nombres:
18 [7, 14, 21, 28, 35]
19 Au revoir!

```

7.3 Exercice 13

ATTENTION cet exercice est compliqué ! Ne reste pas bloqué dessus 2heures...

7.3.1 Objectif

Pareil que l'exercice 12, mais cette fois-ci trier le tableau par ordre croissant avant de l'afficher à la fin.

7.3.2 Exemple

```

1  Entrez un nombre entre 5 et 15:
2  2
3  Entre 5 et 15 svp:
4  blbl
5  Un entier valide svp:

```

```

6 5
7 1) Entrez un nombre:
8 21
9 2) Entrez un nombre:
10 7
11 3) Entrez un nombre:
12 14
13 4) Entrez un nombre:
14 35
15 5) Entrez un nombre:
16 28
17 Vous avez entre les nombres:
18 [7, 14, 21, 28, 35]
19 Au revoir!

```

7.3.3 Aide

Cherche sur Google et regarde dans le cours de l'ESI ! Non mais... Quel chacal celui-là j'te jure

8 Solutions

8.1 Exercice 1

```

1 public class Exercice1 {
2     public static void main(String[] args) {
3         Scanner kb = new Scanner(System.in);
4
5         System.out.println("Entre ton pseudo: ");
6         String pseudo = kb.nextLine();
7
8         System.out.println("Entre ton âge: ");
9         int age = kb.nextInt();
10
11         if(age < 18) {
12             System.out.println("Bonjour " + pseudo + ", tu es mineur!");
13         } else {
14             System.out.println("Bonjour " + pseudo + ", tu es majeur!");
15         }
16     }
17 }

```

8.2 Exercice 2

```

1 public class Exercice2 {
2     public static void main(String[] args) {
3         Scanner kb = new Scanner(System.in);
4
5         System.out.println("Entre ton pseudo: ");
6         String pseudo = kb.nextLine();
7
8         System.out.println("Entre ton âge: ");
9         int age = kb.nextInt();
10
11         if(pseudo.equals("")) {
12             System.out.println("Pseudo invalide...");
13         }
14         else if(age < 0) {
15             System.out.println("Tu n'es pas encore né...");
16         }
17     }
18 }

```

```

17     else if(age > 120) {
18         System.out.println("Tu es sans doute décédé...");
19     }
20     else if(age < 18) {
21         System.out.println("Bonjour " + pseudo + ", tu es mineur!");
22     } else {
23         System.out.println("Bonjour " + pseudo + ", tu es majeur!");
24     }
25 }
26 }

```

8.2.1 Variante

```

1 //Variante:
2 //...
3 if(pseudo.equals("") && (age < 0 || age > 120)) {
4     System.out.println("T'es bourré ou quoi?");
5 }
6 else if(pseudo.equals("")) {
7     //...

```

8.3 Exercice 3

```

1 public class Exercice1 {
2     public static void main(String[] args) {
3         Scanner kb = new Scanner(System.in);
4
5         System.out.println("Entre ton pseudo: ");
6         String pseudo = kb.nextLine();
7
8         while(pseudo.equals("")) {
9             System.out.println("Pseudo invalide! Recommence: ");
10            pseudo = kb.nextLine();
11        }
12
13        System.out.println("Entre ton âge: ");
14        int age = kb.nextInt();
15
16        while(age < 0 || age > 120) {
17            System.out.println("Âge invalide! Recommence: ");
18            age = kb.nextInt();
19        }
20
21        if(age < 18) {
22            System.out.println("Bonjour " + pseudo + ", tu es mineur!");
23        } else {
24            System.out.println("Bonjour " + pseudo + ", tu es majeur!");
25        }
26    }
27 }

```

8.3.1 Variante 1

```

1 //Variante 1
2 //...
3 System.out.println("Entre ton pseudo: ");
4 String pseudo = kb.nextLine();
5

```

```

6  int i = 0;
7  while(pseudo.equals("")) {
8      if(i >= 2) {
9          System.out.println("Tu te fous de moi? Allez ciao.");
10         System.exit(0);
11     }
12     System.out.println("Pseudo invalide! Recommence: ");
13     pseudo = kb.nextLine();
14     i++;
15 }
16
17 System.out.println("Entre ton âge: ");
18 int age = kb.nextInt();
19
20 i = 0; //Ne pas déclarer i une deuxième fois!
21 while(age < 0 || age > 120) {
22     if(i >= 2) {
23         System.out.println("Tu te fous de moi? Allez ciao.");
24         System.exit(0);
25     }
26     System.out.println("Âge invalide! Recommence: ");
27     age = kb.nextInt();
28     i++;
29 }
30 //...

```

8.3.2 Variante 2

```

1  //Variante 2
2  //...
3  System.out.println("Entre ton pseudo: ");
4  String pseudo = kb.nextLine();
5
6  int i = 0;
7  while(pseudo.equals("") && i < 2) {
8      System.out.println("Pseudo invalide! Recommence: ");
9      pseudo = kb.nextLine();
10     i++;
11 }
12
13 if(pseudo.equals("")) {
14     pseudo = "invité";
15 }
16
17 System.out.println("Entre ton âge: ");
18 int age = kb.nextInt();
19
20 i = 0; //Ne pas déclarer i une deuxième fois!
21 while((age < 0 || age > 120) && i < 2) {
22     System.out.println("Âge invalide! Recommence: ");
23     age = kb.nextInt();
24     i++;
25 }
26
27 if(age < 0 || age > 120) {
28     age = 0;
29 }
30 //...

```