

Java Dragon

Sm!le42

27 juin 2021

Table des matières

0.1	Règles	1
0.1.1	Aide autorisée	1
0.1.2	Temps imposé	1
0.1.3	Remise des exercices	1
0.1.4	Ce que tu dois payer	2
1	Base	2
1.1	Exercice 1	2
1.1.1	Objectif	2
1.1.2	Contraintes	2
1.1.3	Exemples	2
1.2	Exercice 2	2
1.2.1	Objectif	2
1.2.2	Exemples	2
1.2.3	Aide	2
1.2.4	Variante	3
2	Boucles (itérations)	3
2.1	Exercice 3	3
2.1.1	Objectif	3
2.1.2	Aide	3
2.1.3	Variantes	3

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 deux possibilités.
Imaginons que je ne tape pas de pseudo, et que j'entre 142 pour l'âge, alors :

```
1 //Le programme pourrait afficher:
2 Pseudo invalide
3
4 //Ou alors, le programme pourrait afficher:
5 Pseudo invalide
6 Tu es sans doute dédécé...
```

Essaie de trouver la différence entre ces deux codes !;)

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);`