Les exceptions dans le langage Java – TP 1

Le but du TP est de coder différentes méthodes qui s'appliqueront à des durées exprimées en heures et minutes. Pour être valides, les heures devront être positives ou nulles. Les minutes quant à elles devront être comprises entre 0 et 59. Les durées seront stockées dans des tableaux de 2 cases : à l'indice 0, on trouvera les heures et à l'indice 1 les minutes.

Pour les différentes méthodes de la classe, le principe sera le suivant : si l'une des durées passées en argument est invalide, la méthode propagera vers son appelant l'exception prédéfinie ***IllegalArgumentException***.

Au fur et à mesure de l'écriture des méthodes, il vous sera demandé d'écrire un programme de test. Le but est de tester le bon fonctionnement dans le cas nominal (celui qui ne génère pas la levée de l'exception ***IllegalArgumentException***) et aussi dans le cas d'erreur (donc lorsque l'exception est levée).

Préalable

Récupérer les fichiers ***OutilDureeHMACompleter.java*** et ***TestOutilDureeACompleter.java***. Modifier les noms des fichiers et les paquetages pour les adapter à votre organisation.

La classe ***OutilDureeHM*** contient des méthodes outils sur les durées exprimées en heures et minutes. La classe ***TestOutilDuree*** est la classe de test de la précédente : elle contient des méthodes permettant de tester le bon fonctionnement des méthodes de la classe ***OutilDureeHM***.

Exercice 1 – Test validité d'une durée

1) Ajouter à la classe ***OutilDureeHM*** une méthode nommée ***estValide*** qui aura en paramètre une durée, sous la forme d'un tableau de 2 entiers. La méthode renverra un booléen, vrai si la durée argument est valide (voir condition dans le 1er paragraphe), faux dans le cas contraire

2) Tester la méthode ***estValide*** à l'aide de la méthode nommée ***testEstValide*** et présente dans la classe de test. Si besoin modifier le code de ***estValide*** pour la mettre au point.

3) Une fois que la méthode ***estValide*** est correcte, mettre en commentaire, dans la fonction *main*, l'appel à la méthode de test.

Exercice 2 – Affichage d'une durée et test simple

1) Dans la classe ***OutilDureeHM***, modifier la méthode ***afficher*** afin qu'elle lève l'exception ***IllegalArgumentException*** si la durée argument n'est pas valide. En même que l'exception, elle propagera le message d'erreur « La durée argument est invalide ».

2) Tester la méthode ***afficher*** à l'aide de la méthode de test nommée ***testAfficherSimplifie***. Vous constaterez que l'exception n'est pas interceptée dans la méthode de test et qu'en conséquence le programme s'arrête en état d'erreur.

3) Modifier la méthode de test ***testAfficherSimplifie*** pour qu'elle intercepte l'exception ***IllegalArgumentException***. Il faudra donc utiliser un bloc ***try-catch***. L'objectif est d'obtenir l'affichage suivant :

```
Test de la méthode de afficher (test préalable très simple)
-----------------------------------------------------------

Exemple d'affichage d'une durée valide :  12 heure(s) 25 minute(s)

Exemple d'affichage d'une durée invalide :
Test correct pour l'affichage d'une durée invalide.
Le message d'erreur est le suivant : La durée à afficher est invalide
```

Exercice 3 – Test de la méthode d'affichage à partir d'un tableau de durées

1) Ecrire une nouvelle méthode de test de la méthode ***afficher*** qui se nommera ***testAfficher*** (le début de la méthode est déjà codé). Lorsque c'est possible, elle devra afficher les durées présentes dans le tableau constant nommé ***DUREE***. Dans cette première question, les durées invalides ne seront pas affichées.

```
Exemples d'affichage de durées valides :
  10 heure(s) 25 minute(s)
  9 heure(s) 0 minute(s)
  21 heure(s) 59 minute(s)
  23 heure(s) 12 minute(s)
  23 heure(s) 59 minute(s)
  0 heure(s) 45 minute(s)
  0 heure(s) 0 minute(s)
  24 heure(s) 25 minute(s)
  56 heure(s) 15 minute(s)
  2000 heure(s) 0 minute(s)
```

2) Modifier la méthode de test précédente pour que, dans le cas où la durée située à l'indice ***i*** ne peut pas être affichée, car invalide, un message s'affiche et indique l'indice de la durée qui n'a pas pu être affichée. Il sera nécessaire d'utiliser un bloc ***try-catch***.

```
Tentative d'affichage de toutes les durées :

10 heure(s) 25 minute(s)
9 heure(s) 0 minute(s)
 ==> Impossible d'afficher la durée située à l'indice 2. Elle n'est pas valide.
21 heure(s) 59 minute(s)
23 heure(s) 12 minute(s)
 ==> Impossible d'afficher la durée située à l'indice 5. Elle n'est pas valide.
23 heure(s) 59 minute(s)
0 heure(s) 45 minute(s)
 ==> Impossible d'afficher la durée située à l'indice 8. Elle n'est pas valide.
0 heure(s) 0 minute(s)
24 heure(s) 25 minute(s)
 ==> Impossible d'afficher la durée située à l'indice 11. Elle n'est pas valide.
56 heure(s) 15 minute(s)
2000 heure(s) 0 minute(s)
```

Exercice 4 – Somme de 2 durées et test

1) Ajouter à la classe ***OutilDureeHM*** une méthode qui aura en paramètre 2 durées, sous la forme de 2 tableaux d'entiers, et qui renverra la somme des 2 durées (aussi sous la forme d'un tableau de 2 entiers). Si l'une des durées argument n'est pas valide, la méthode lèvera l'exception ***IllegalArgumentException***. En même temps que l'exception sera propagée, un message d'erreur sera communiqué au programme appelant : « Calcul de la somme impossible. L'une des 2 durées est invalide ».

2) Ajouter à la classe de test une méthode pour tester la méthode précédente. On procèdera en 3 étapes.

a) Ajouter la durée nommée ***REFERENCE*** à chacune des durées du tableau ***DUREE_VALIDE***. Normalement l'exception ne doit pas se produire. Si toutefois elle se produisait, un message d'erreur sera affiché.

```
==> TEST 1 - Ajout de la durée 2 heure(s) 40 minute(s)  à des durées valides:

  2 heure(s) 40 minute(s)  + 10 heure(s) 25 minute(s)  = 13 heure(s) 5 minute(s)
  2 heure(s) 40 minute(s)  + 9 heure(s) 0 minute(s)  = 11 heure(s) 40 minute(s)
  2 heure(s) 40 minute(s)  + 21 heure(s) 59 minute(s)  = 24 heure(s) 39 minute(s)
  2 heure(s) 40 minute(s)  + 23 heure(s) 12 minute(s)  = 25 heure(s) 52 minute(s)
  2 heure(s) 40 minute(s)  + 23 heure(s) 59 minute(s)  = 26 heure(s) 39 minute(s)
  2 heure(s) 40 minute(s)  + 0 heure(s) 45 minute(s)  = 3 heure(s) 25 minute(s)
  2 heure(s) 40 minute(s)  + 0 heure(s) 0 minute(s)  = 2 heure(s) 40 minute(s)
  2 heure(s) 40 minute(s)  + 24 heure(s) 25 minute(s)  = 27 heure(s) 5 minute(s)
  2 heure(s) 40 minute(s)  + 56 heure(s) 15 minute(s)  = 58 heure(s) 55 minute(s)
```

b) Ajouter la durée nommée ***REFERENCE*** à chacune des durées du tableau ***DUREE_INVALIDE***. Normalement l'exception doit se produire.

```
==> TEST 2 -  Ajout de la durée 2 heure(s) 40 minute(s)  à des durées invalides:

  Ajout à la durée d'indice 0 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 1 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 2 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 3 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 4 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 5 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 6 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 7 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 8 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
  Ajout à la durée d'indice 9 :   Calcul de la somme impossible. L'une des 2 durées est invalide.
```

c) Ajouter la durée nommée ***REFERENCE*** à chacune des durées du tableau ***DUREE***. Normalement l'exception doit se produire pour certains calculs. Le programme de test devra afficher le nombre de calculs impossibles.

```
==> TEST 3 - Ajout de la durée 2 heure(s) 40 minute(s)  à différentes durées valides ou invalides:

   Au total 4 calculs n'ont pas été possibles.
```

Exercice 5 – Somme d'un tableau de durées et test

1) Ajouter à la classe ***OutilDureeHM*** une méthode qui aura en paramètre un tableau contenant des durées en heures et minutes (donc un tableau à 2 dimensions). La méthode calculera et renverra la somme de toutes les durées du tableau. Si l'une d'elle n'est pas valide, elle ne sera pas prise en compte dans le calcul. Aucune exception ne sera propagée vers l'appelant.

2) Ajouter à la classe de test une méthode pour tester la méthode précédente. On pourra additionner les durées des 3 tableaux constants.

```
Test de la méthode de sommeTableDuree
-------------------------------------

==> TEST 1 - Somme de durées valides :
   La somme est égale à 170 heure(s) 0 minute(s)

==> TEST 2 -  Somme de durées invalides :
   La somme est égale à 0 heure(s) 0 minute(s)

==> TEST 3 -  Somme de durées valides ou invalides :
   La somme est égale à 2170 heure(s) 0 minute(s)
```

Exercice 6 – Egalité de 2 durées et test

1) Ajouter à la classe ***OutilDureeHM*** une méthode qui aura en paramètre 2 durées exprimées en heures et minutes. La méthode déterminera si les 2 durées sont identiques ou pas. Si l'une des 2 durées est invalide, l'exception ***IllegalArgumentException*** sera propagée vers l'appelant accompagnée d'un message d'erreur : « Comparaison impossible. L'une des 2 durées est invalides. ».

2) Ajouter à la classe de test une méthode pour tester la méthode précédente. On comparer la durée constante ***A_COMPARER*** avec chacune des durées du tableau ***DUREE***. Si une durée comparée est invalide, le résultat de la comparaison ne sera pas affiché. Mais cette comparaison impossible sera comptabilisée dans un total affiché à la fin du test.

```
Test de la méthode Egalite
--------------------------

Comparaison de la durée 21 heure(s) 59 minute(s)  à différentes durées valides ou invalides:

  Difference avec 10 heure(s) 25 minute(s)
  Difference avec 9 heure(s) 0 minute(s)
  Egalité avec 21 heure(s) 59 minute(s)
  Difference avec 23 heure(s) 12 minute(s)
  Difference avec 23 heure(s) 59 minute(s)
  Difference avec 0 heure(s) 45 minute(s)
  Difference avec 0 heure(s) 0 minute(s)
  Difference avec 24 heure(s) 25 minute(s)
  Difference avec 56 heure(s) 15 minute(s)
  Difference avec 2000 heure(s) 0 minute(s)

  Au total 4 comparaisons n'ont pas été possibles.
```

Exercice 5 – Combien de durées différentes parmi 3 et test

1) Ajouter à la classe ***OutilDureeHM*** une méthode qui aura en paramètre 3 durées exprimées en heures et minutes. La méthode déterminera combien de durées parmi les 3 sont différentes. Le résultat renvoyé pourra être 1 (si toutes les durées sont égales), 2 ou 3 (si elles sont toutes différentes). La méthode propagera l'exception ***IllegalArgumentException*** si l'une des durées argument est invalide.

2) Ajouter à la classe de test une méthode pour tester la méthode précédente. On envisagera différents cas, y compris des cas où l'une des durées à traiter est invalide.