Cours MOOC EPFL d'introduction à la programmation orientée objet, illustré en Java

Premier devoir
Classes et objets

J. Sam & J.-C. Chappelier

du 30 octobre au 17 novembre 2014

1 Exercice 1 — IMC

Le but de cet exercice est de créer des « patients » qui ont un poids et une taille, et de calculer leur « Indice de Masse Corporelle » (IMC).

1.1 Description

Télécharger le programme fourni sur le site du cours [1] et le compléter.

ATTENTION : vous ne devez modifier ni le début ni la fin du programme, juste ajouter vos propres lignes à l'endroit indiqué. Il est donc primordial de respecter la procédure suivante (les points 1 et 3 concernent spécifiquement les utilisateurs d'Eclipse) :

1. désactiver le formatage automatique dans Eclipse :
   `Window > Preferences > Java > Editor > Save Actions`
   (et décocher l'option de reformatage si elle est cochée)
2. sauvegarder le fichier téléchargé sous le nom `Imc.java` (avec une majuscule, notamment). Si vous travaillez avec Eclipse vous ferez cette sauvegarde à l'emplacement `[dossierDuProjetPourCetExercice]/src/` ;
3. rafraîchir le projet Eclipse où est stocké le fichier (clic droit sur le projet > refresh) pour qu'il le prenne en compte ;

1. `https://d396qusza40orc.cloudfront.net/intropoojava/assignments-data/Imc.java`

4. écrire le code à fournir entre ces deux commentaires :

```
/*****************************************
 * Completez le programme a partir d'ici.
 *****************************************/

/*****************************************
 * Ne rien modifier apres cette ligne.
 *****************************************/
```

5. sauvegarder et tester son programme pour être sûr(e) qu'il fonctionne correctement, par exemple avec les valeurs données plus bas ;
6. rendre le fichier modifié (toujours `Imc.java`) dans « OUTPUT submission » (et non pas dans « Additional ! »).

Le code fourni

— crée un patient,
— affiche les données du patient ainsi que son IMC.

Ces deux étapes sont répétées deux fois avec des valeurs différentes.

La définition de la classe `Patient` manque et c'est ce qu'il vous est demandé d'écrire.

Un patient est caractérisé par un poids et une taille. Vous nommerez les attributs correspondants respectivement `masse` et `hauteur` (veuillez respecter strictement ces consignes).

Par ailleurs, les méthodes spécifiques à un patient sont :

— une méthode `init` prenant en paramètre deux `double`, le premier pour initialiser le poids du patient et le second pour initialiser sa taille ;
ces données ne seront affectées aux attributs du patient que si elles sont toutes les deux positives ; dans le cas contraire, la taille et le poids du patient seront **tous deux initialisés à zéro** ; pour simplifier, il n'y a pas d'autre contrôle à faire sur ces données ;

— une méthode `afficher` permettant d'afficher sur le terminal les caractéristiques du patient en respectant ***strictement*** le format suivant :
`Patient : <poids> kg pour <taille> m`
où `<poids>` est à remplacer par le poids du patient et `<taille>` par sa taille ; **cet affichage sera terminé par un saut de ligne**. Vous n'afficherez qu'un chiffre après la virgule pour le poids et la taille. Vous vous inspirerez pour cela du code suivant qui affiche les deux doubles avec un chiffre après la virgule :

```
double x = 1.2456;
double y = 1.4568;
```

```
// affiche : "1.2 et 1.5"
System.out.printf("j'affiche deux doubles formattés %.1f et %.1f
   x, y);
```

— une méthode `poids` retournant le poids du patient ;

— une méthode `taille` retournant la taille du patient ;

— une méthode `imc` retournant l'IMC du patient : son poids divisé par le carré de sa taille ; **en cas de taille nulle, cette méthode retournera zéro**.

Ces méthodes feront partie de l'interface d'utilisation de la classe.

Un exemple de déroulement possible est fourni plus bas.

1.2 Exemple de déroulement

```
Patient : 74.5 kg pour 1.8 m
IMC : 24.3265306122449
Patient : 0.0 kg pour 0.0 m
```

2 Exercice 2 — Tirelire

Le but de cet exercice est de simuler une tirelire dans laquelle on stocke et retire de l'argent et que l'on souhaite utiliser pour payer un certain budget (de vacances, par exemple).

2.1 Description

Télécharger le programme fourni sur le site du cours [2] et le compléter.

ATTENTION : vous ne devez modifier ni le début ni la fin du programme, juste ajouter vos propres lignes à l'endroit indiqué. Il est donc primordial de respecter la procédure suivante (les points 1 et 3 concernent spécifiquement les utilisateurs d'Eclipse) :

1. désactiver le formatage automatique dans Eclipse :
   `Window > Preferences > Java > Editor > Save Actions`
   (et décocher l'option de reformatage si elle est cochée)

2. `https://d396qusza40orc.cloudfront.net/intropoojava/assignments-data/TestTirelire.java`

2. sauvegarder le fichier téléchargé sous le nom `TestTirelire.java` (avec une majuscule, notamment). Si vous travaillez avec Eclipse vous ferez cette sauvegarde à l'emplacement `[dossierDuProjetPourCetExercice]/src/` ;
3. rafraîchir le projet Eclipse où est stocké le fichier (clic droit sur le projet > refresh) pour qu'il le prenne en compte ;
4. écrire le code à fournir entre ces deux commentaires :

```
/*****************************************
 * Completez le programme a partir d'ici.
 *****************************************/

/*****************************************
 * Ne rien modifier apres cette ligne.
 *****************************************/
```

5. sauvegarder et tester son programme pour être sûr(e) qu'il fonctionne correctement, par exemple avec les valeurs données plus bas ;
6. rendre le fichier modifié (toujours `TestTirelire.java`) dans « OUTPUT submission » (et non pas dans « Additional ! »).

Le code fourni crée une tirelire et lui fait subir divers manipulations (la vider, la secouer, en afficher le contenu etc.).

Ce programme demande aussi à l'utilisateur quel budget il aimerait consacrer à ses vacances.

Si la tirelire contient suffisamment d'argent (ce budget ou plus), il indique combien d'argent il resterait dans la tirelire après les vacances. Dans la cas contraire, il indique quel montant manque pour partir en vacances avec le budget souhaité.

La définition de la classe `Tirelire` manque et il vous est demandé de la fournir.

Une tirelire est caractérisée par le *montant* qu'elle contient. Vous nommerez cet attribut avec ce nom dans votre programme.

Les traitements qui lui sont spécifiques sont :

- une méthode `getMontant` retournant le montant de la tirelire ;
- une méthode `afficher` affichant les données de la tirelire sous le format suivant :
  - `Vous etes sans le sous.`
    si la tirelire ne contient pas d'argent (les accents on été délibérément omis pour éviter les problèmes d'encodage)
  - `Vous avez : <montant> euros dans votre tirelire.`
    dans le cas contraire (où `<montant>` est le montant de la tirelire, affiché sans formattage particulier)

— une méthode `secouer` affichant sur le terminal le message `Bing bing`, suivi d'un saut de ligne, dans le cas où la tirelire contient de l'argent, et qui n'affiche rien sinon ;
— la méthode `remplir` mettant un montant donné en paramètre (`double`) dans la tirelire. Seuls les montant positifs seront acceptés (dans le cas contraire on ne fait rien) ;
— une méthode `vider` (re)intialisant le montant de la tirelire à zéro ;
— une méthode `puiser` permettant de puiser dans la tirelire un montant donné en paramètre. Si le montant est négatif il sera ignoré. Si le montant en argument est plus grand que le montant disponible, la tirelire est alors vidée. La méthode `puiser` ne retourne rien.
— une méthode `calculerSolde` qui retourne la différence entre le montant de la tirelire et le budget que l'on souhaite dépenser (un `double`). Si le budget est négatif (ou nul), la méthode `calculerSolde` doit retourner le montant de la tirelire.

Ces méthodes feront partie de l'interface d'utilisation de la classe.

Deux exemples de déroulement possibles sont fournis plus bas.

2.2 Exemples de déroulement

```
Vous etes sans le sous.
Vous etes sans le sous.
Bing bing
Vous avez : 550.0 euros dans votre tirelire.
Vous avez : 535.0 euros dans votre tirelire.

Donnez le budget de vos vacances :
450
Vous etes assez riche pour partir en vacances !
il vous restera 85.0 euros a la rentree
```

ou

```
Vous etes sans le sous.
Vous etes sans le sous.
Bing bing
Vous avez : 550.0 euros dans votre tirelire.
Vous avez : 535.0 euros dans votre tirelire.

Donnez le budget de vos vacances :
```

```
1250.0
Il vous manque 715.0 euros pour partir en vacances !
```