Cours MOOC EPFL d'initiation à la programmation illustré en Java

Second devoir noté
Branchements conditionnels

J. Sam & J.-C. Chappelier

1 Exercice 1 — Location de vélos

Le but de cet exercice est de permettre à un service de location de vélos (online, tournant 24 heures sur 24) de facturer ses clients.

Le programme demandera à l'utilisateur d'entrer les heures de début et de fin de location sous la forme d'entiers (on ne se préoccupe pas des minutes pour simplifier).

Les tarifs de location sont définis comme suit :

— 1 franc par heure si le vélo est loué entre 0h et 7h ou entre 17h et 24h ;

— 2 francs par heure si le vélo est loué entre 7h et 17h.

Votre programme demandera à l'utilisateur de quelle heure à quelle heure se fait la location (partie du code fournie) et calculera le prix de la location en conséquence.

Vous adoterez les simplifications suivantes :

— les heures de début et fin de location sont des entiers (pas de demi ni de quart, toute heure entamée est due) ;

— l'heure du début de la location est toujours inférieure à l'heure de la fin de la location ;
cela implique que la location ne peut pas se faire sur plus de 24 heures ; elle doit se faire dans la même journée.

Si les données introduites sont correctes, votre programme affichera simplement le coût de la location en respectant *strictement* les formats donnés dans les exemples de déroulement ci-dessous.

En cas de donnée incorrecte, votre programme devra afficher un message d'erreur et s'arrêter. Utilisez *strictement* les messages suivants :

— « `Les heures doivent être comprises entre 0 et 24 !` » suivi d'un saut de ligne, si une des heures introduites par l'utilisateur n'est pas comprise entre 0 et 24 (inclus) ;

— « `Bizarre, vous n'avez pas loué votre vélo bien longtemps !` » suivi d'un saut de ligne, si les heures de début et fin de location sont identiques ;

— et « `Bizarre, le début de la location est après la fin ...` » suivi d'un saut de ligne si l'heure de début de la location est supérieure à l'heure de fin.

Le code fourni contient des instructions à utiliser telles quelles (en les plaçant aux bons endroits) pour produire les affichages souhaités.

Vous n'utiliserez pas l'instruction `return;` pour ce programme.

1.1 Donnée

Télécharger le programme `Velo.java` fourni sur le site du courset le compléter suivant les instructions données ci-dessous.

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 `Velo.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 `Velo.java`) dans « OUTPUT submission » (et non pas dans « Additional ! »).

1.2 Exemples de déroulement

Il est impératif que votre code respecte le format de réponse suivant :

1) Exemple où la durée de location implique les deux tarifs :

```
Donnez l'heure de début de la location (un entier) : 10
Donnez l'heure de fin de la location (un entier) : 19
Vous avez loué votre vélo pendant
2 heure(s) au tarif horaire de 1.0 franc(s)
7 heure(s) au tarif horaire de 2.0 franc(s)
Le montant total à payer est de 16.0 franc(s).
```

2) Exemple où la durée de location n'implique qu'un seul tarif :

```
Donnez l'heure de début de la location (un entier) : 18
Donnez l'heure de fin de la location (un entier) : 20
Vous avez loué votre vélo pendant
2 heure(s) au tarif horaire de 1.0 franc(s)
Le montant total à payer est de 2.0 franc(s).
```

2 Exercice 2 – Identification de champignons

2.1 Introduction

Le but de cet exercice est d'écrire un programme Java posant des questions à l'utilisateur pour deviner (parmi une liste connue à l'avance) à quel champignon pense l'utilisateur.

Pour deviner un champignon, le programme ne peut poser que **trois** questions **au maximum** [1], dont la réponse est soit oui, soit non (l'utilisateur répondra aux questions du programme par `false` pour non, et par `true` pour oui ; voir l'exemple de déroulement fourni plus bas).

Les 6 champignons possibles sont :

— l'agaric jaunissant ;
— l'amanite tue-mouches ;
— le cèpe de Bordeaux ;
— le coprin chevelu ;
— la girolle ;
— et le pied bleu.

Seul le cèpe de Bordeaux possède des tubes, les autres champignons ayant des lamelles.

Le coprin chevelu et l'agaric jaunissant poussent dans les prés, les autres dans la forêt.

Les seuls à avoir un chapeau convexe sont l'agaric jaunissant, l'amanite tue-mouches et le pied bleu.

Enfin, les seuls à avoir un anneau sont l'agaric jaunissant, l'amanite tue-mouches et le coprin chevelu.

Commentaire : `clavier.nextBoolean()` permet de lire les booléens.

2.2 Instructions

Pour ce devoir, nous ne vous imposons pas de code au départ, mais simplement le format des questions et les noms des champignons.

Pour vous faciliter leur écriture, téléchargez le programme `Champi.java` fourni sur le site du courset utilisez le code fourni à votre guise, mais sans modifier les affichages.

Ce qu'il vous faut faire, c'est écrire tout le programme (en utilisant les `System.out.print` fournis) de sorte à ce qu'il puisse trouver, en **trois** questions **maximum** le champignon auquel pense l'utilisateur (dans le cadre décrit plus haut).

1. Mais ce ne sont pas forcément les trois mêmes questions à chaque fois !

Faites simplement attention à ne pas modifier le texte des questions (mais déplacez les pour changer l'ordre si nécessaire) ; notre correcteur automatique s'appuie sur le texte de ces questions pour évaluer votre programme.

Une des difficultés de cet exercice consiste à trouver quelles questions poser et dans quel ordre. Tous les ordres ne sont pas équivalents et ne conduisent pas à la solution en trois questions maximum.

Note : On suppose que l'utilisateur respecte les règles. Si les réponses de l'utilisateur sont incohérentes ou incorrectes, l'affichage du programme n'est pas spécifié, c.-à-d. qu'il peut être n'importe quoi suivant votre choix.

Nous ne testerons pas ces cas là. Notre correcteur ne fournira que des réponses correctes et cohérentes à votre programme.

2.3 Exemples de déroulement

Attention, ces exemples ne sont là que pour donner une idée de ce qui se passe. Il n'est pas dit que les questions posées ici soient pertinentes ni dans le bon ordre pour garantir trois questions maximum !

Il n'est pas dit non plus que ces deux exemples proviennent du même programme. Ce sont juste des exemples possibles d'interactions avec l'utilisateur.

Exemple 1 :

```
Pensez à un champignon : amanite tue mouches, pied bleu, girolle,
cèpe de Bordeaux, coprin chevelu ou agaric jaunissant.

Est-ce que votre champignon a des lamelles (true : oui, false : non) ? true
Est-ce que votre champignon a un anneau (true : oui, false : non) ? false
Est-ce que votre champignon a un chapeau convexe (true : oui, false : non) ? false
==> Le champignon auquel vous pensez est la girolle.
```

Exemple 2 :

```
Pensez à un champignon : amanite tue mouches, pied bleu, girolle,
cèpe de Bordeaux, coprin chevelu ou agaric jaunissant.

Est-ce que votre champignon vit en forêt (true : oui, false : non) ? true
Est-ce que votre champignon a des lamelles (true : oui, false : non) ? false
==> Le champignon auquel vous pensez est le cèpe de Bordeaux.
```