Licence 3 Informatique

Deuxième Session

11/06/2020

- Le barème est donné à titre indicatif.
- Les réponses sont attendues sous forme d'un unique document pdf par personne. Vous pouvez utiliser votre traitement de texte favori, mais pensez bien à exporter le document en pdf avant de l'envoyer (pas de docx, odt,...).
- 3 points seront dédiés à la qualité générale de votre code (indentation, respect des conventions de nommage, choix des noms de variables/méthodes/classes, utilisation de commentaires Javadoc pertinents, encapsulation,...).

Exercice I (2 points)

(1) 1. Dans une classe UtilTime, définissez :

- une méthode qui prend en entrée trois nombres entiers représentant respectivement des heures, minutes et secondes, et qui retourne le nombre de secondes correspondant.
- une méthode qui prend en entrée un nombre entier représentant des secondes, et qui retourne une chaîne de caractères correspondant à ce temps au format heures/minutes/secondes.

(1) 2. On suppose qu'on a une liste de nombres entiers qui représentent des temps en secondes. Définissez une méthode qui permet d'obtenir le temps le plus long, sous forme de chaîne de caractères (au format heures/minutes/secondes).

Exercice II (6 points)

Une liste chaînée est une structure de données représentant une collection ordonnée et de taille arbitraire d'éléments du même type. Par "ordonnée", on veut dire que chaque élément a une position déterminée par un indice précis, contrairement à un ensemble (mais cela ne veut pas forcément dire que la liste est triée). On représente un de ces éléments sous forme d'un "maillon", c'est-à-dire une cellule qui contient une valeur et une référence vers le maillon suivant (ou **null** si c'est le dernier maillon de la chaîne). On peut donc manipuler une liste chaînée juste en connaissant son premier maillon.
Pour les questions suivantes, vous êtes invités à ajouter les éventuels attributs ou méthodes qui vous semblent nécessaires.

(1) 1. Définissez la classe Maillon qui contient une donnée stockée sous forme de chaîne de caractères, et une référence vers le Maillon suivant.

(1) 2. Définissez la classe ListeChainee qui permet de représenter une liste de chaines de caractères. Cette classe doit avoir au moins deux constructeurs :

- un constructeur qui crée une liste vide ;
- un constructeur qui prend en entrée un tableau de String, et construit la liste chainée correspondante.

(1½) 3. Définissez une méthode qui permet d'ajouter une chaîne de caractères à une position donnée. Si cette position n'est pas possible (valeur négative, ou plus grande que le nombre d'éléments dans la liste), la méthode doit lever une exception d'un type particulier que vous aurez créé dans ce but.

(½) 4. Créez une méthode qui ajoute un élément à la première position de la liste, et une méthode qui ajoute un élément à la dernière position de la liste.

(2) 5. Créez une méthode qui enregistre dans un fichier (dont le chemin est passé en entrée) l'ensemble des éléments de la liste, avec un élément par ligne.

Exercice III (6 points)

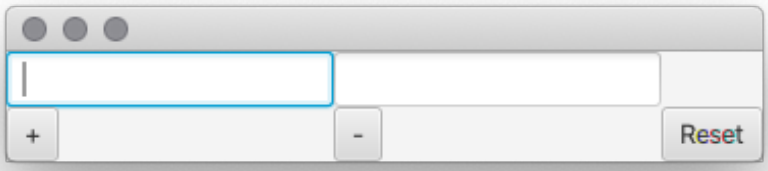
On peut représenter une formule logique (comme celles qui sont évaluées dans des expressions **if** ou **while**) sous forme d'un arbre, dont les noeuds internes sont les connecteurs logiques OR (||), AND (&&) et NOT (!), et dont les feuilles sont les valeurs booléennes (**true** ou **false**). On souhaite pouvoir représenter ces expressions et évaluer leur valeur.

(4) 1. Implémentez les classes nécessaires à cette représentation.

(2) 2. Donnez des tests unitaires pour la méthode qui permet d'évaluer un noeud AND.

Exercice IV (3 points)

(1½) 1. Donnez le code Java qui permet de produire l'interface JavaFX suivante :



Remarque : La zone au dessus du bouton "Reset" est un label (qui est vide lors de l'initialisation de la fenêtre). Par contre, il n'y a rien entre le bouton "+" et le bouton "-", ni entre le bouton "-" et le bouton "Reset".

(1½) 2. Indiquez les modifications à apporter au code précédent pour gérer les événements :

- lors d'un click sur le bouton "+", les valeurs présentes dans les deux champs de texte sont additionnées et affichées dans le label.
- lors d'un click sur le bouton "-", les valeurs présentes dans les deux champs de texte sont soustraites et affichées dans le label.
- lors d'un click sur le bouton "Reset", le contenu du label est effacé.

Pour cet exercice, on suppose que les valeurs présentes dans les champs de texte sont bien des nombres (il n'est pas nécessaire de le vérifier, ou de gérer les erreurs de l'utilisateur)