Les tableaux en Java

1 Mon tableau de caractères

Dans cet exercice, nous définissons une classe `MonTableauDeCaracteres` avec des méthodes classiques sur les tableaux de caractères et sur les mots en général. Nous n'utilisons pas la classe `String` car l'objectif est de vous faire manipuler les tableaux en Java. En parallèle à la classe `MonTableauDeCaracteres`, développez une classe `ProgrammeTableauDeCaracteres` pour tester vos méthodes au fur et à mesure.

1.1 Eléments de base de la classe

- Créez une classe `MonTableauDeCaracteres`.
- Donnez-lui un attribut `contenu` qui est un tableau de caractères.
- Prévoyez les constructeurs et accesseurs que vous jugez utiles.

1.2 Méthodes de gestion du tableau

- Ecrivez une méthode permettant d'afficher le contenu d'un tableau (prévoyez le cas où le tableau n'est pas créé).
- Ecrivez une méthode `void saisirEtCreer(Scanner c)` permettant de créer et saisir le contenu d'un tableau (vous demanderez la taille prévue pour le tableau puis les différents éléments qui le composent).

1.3 Méthodes de manipulation du tableau

- Ecrire une méthode qui compte le nombre d'occurrences d'un caractère dans un tableau.
- Ecrire une méthode qui inverse le contenu du tableau. Par exemple, si le tableau contient les caractères `bonjour`, après appel de la méthode, il contient `roujnob`.
- Ecrire une méthode qui retourne une autre instance de `MonTableauDeCaracteres` dont le contenu est l'inverse du contenu de `this`.
- Ecrire une méthode qui compare deux instances de `MonTableauDeCaracteres` et retourne vrai si elles ont même contenu, faux sinon.
- Ecrire une méthode qui retourne vrai si le contenu d'une instance de `MonTableauDeCaracteres` est un palindrome. Un palindrome est une chaîne de caractères identique qu'elle soit lue de gauche à droite ou de droite à gauche. Par exemple `bob`, `ENGAGELEJEUQUEJELEGAGNE`, ou `madam` sont des palindromes. Vous pouvez écrire cette méthode soit en analysant le contenu, soit en appelant les deux méthodes faites précédemment. Essayez les deux solutions.

2 Le jeu de la vie (à faire seulement si vous êtes en avance)

Le jeu de la vie[1] est un programme qui simule, sur un monde correspondant à une grille en deux dimensions, l'évolution d'une population de cellules avec des règles de vie et de mort sur ces cellules. A l'étape initiale, certaines cellules de la grille sont vivantes, d'autres sont mortes. Puis le système évolue d'étape en étape. L'évolution de l'état d'une cellule d'une étape à l'autre dépend de l'état de ses cellules voisines (huit sauf si on se trouve sur un bord du monde). Une cellule morte qui a exactement trois cellules voisines vivantes naît à l'étape suivante. Une cellule

1. Source : http ://fr.wikipedia.org/wiki/Jeu_de_la_vie

vivante possédant deux ou trois voisines vivantes reste vivante, sinon elle meurt (d'isolement ou d'étouffement).

Pour mettre en œuvre ce jeu, vous définirez trois classes. Quelques méthodes vous sont indiquées, et vous n'hésiterez pas à ajouter des méthodes auxiliaires pour faciliter la programmation.

La classe `Cellule` possède :

- un attribut `occupée` vrai si la cellule est vivante.
- un attribut `age` qui indique depuis combien de générations elle est vivante. L'âge est 0 si elle vient de naître, -1 quand elle est morte.
- un attribut `monMonde` qui indique dans quel monde se trouve la cellule.
- deux attributs `ligne` et `colonne` qui précisent à quel endroit du monde se trouve la cellule.
- des constructeurs et accesseurs appropriés.
- une méthode pour la visualisation, qui retourne par exemple le caractère 'O' quand elle est vivante et le caractère 'X' quand elle est morte.
- une méthode `toString`.
- une méthode calculant le prochain état suivant le voisinage.

La classe `Monde` possède :

- un tableau à deux dimensions, chaque case du tableau contient une cellule.
- un constructeur qui permet de créer correctement ce tableau et les cellules.
- une méthode qui retourne la cellule stockée dans une case dont les indices sont passés en paramètres.
- une méthode d'affichage qui visualise la population de cellules (et si elles sont vivantes ou non).
- une méthode d'initialisation qui attribue des statuts initiaux (vivante/morte) aux cellules.
- une méthode qui permet de passer à une nouvelle génération. Vous devez faire attention à calculer tous les voisinages en vous appuyant sur l'état précédent du monde.

La classe `Jeu` possède un `main` qui crée un monde et lance le jeu.