université PARIS DIDEROT PARIS 7

TYPES DE DONNÉES ET OBJETS (TO2)

TD5 : Listes chaînées simples

Exercice 1 On va dans cet exercice implémenter un jeu de cartes vu comme liste chainée. On considère pour cela la classe `Carte` suivante :

```
class Carte{
        private String couleur;     // couleur de la carte
        private int valeur;         // valeur de la carte
        /* A COMPLETER ... */
}
```

1. Écrire un constructeur qui utilise les deux valeurs passées en paramètres.
2. Écrire les getters et setters associés aux attributs `couleur` et `valeur`.
3. Écrire une méthode `description` qui renvoie une chaîne de caractères décrivant la carte de la façon suivante : `Cette carte est un(e) <valeur> de <couleur>.`
   Si l'attribut `valeur` vaut à 1, 11, 12 ou 13 alors on remplacera `<valeur>` par "as", "valet", "dame" ou "roi" respectivement.
4. Écrire une méthode `estEgalA` qui prend un argument `c` de type `Carte` et qui teste si `this` et `c` ont même valeur et même couleur.

Considérons à présent les deux classes suivantes :

```
class Cellule{
        Carte carte;
        Cellule precedente, suivante;
        /* A COMPLETER ... */
}

class Jeu{
        Cellule premiereCarte;     // la première carte du jeu
        /* A COMPLETER ... */
}
```

5. Écrire un constructeur pour la classe `Cellule` qui prend un argument `c` de type `Carte` et initialise les trois attributs.
6. Écrire un constructeur pour la classe `Jeu` qui construit le jeu ne comprenant aucune carte, puis une méthode `isEmpty` qui teste si `this` ne comprend aucune carte.
7. Dans la classe `Jeu` écrire une méthode `piocher` qui prend un argument `c` de type `Carte` et qui rajoute `c` dans `this` en en faisant la première carte.
8. Écrire une méthode `estPresent` qui teste si la carte `c` en argument apparaît dans `this`.
9. Écrire une méthode `defausser` qui retire `premiereCarte` de `this`.
10. Écrire une méthode `retirer` qui retire de `this` la carte `c` en argument si elle y apparaît.
11. Écrire une méthode `lesHabilles` qui renvoie le jeu composé des habillés (valet, dame, roi) apparaissant dans `this`.

Exercice 2 Le but de cet exercice est, étant donnée une liste d'entiers, de la décrire en donnant ses valeurs extrêmes et sa moyenne. Considérons pour cela les deux classes suivantes :

```
class Cellule {
        int valeur;
        Cellule suivante;
        /* A COMPLETER ... */
}

class ListeDEntiers {
        Cellule premier;
        /* A COMPLETER ... */
}
```

1. Écrire un constructeur pour la classe `Cellule` prenant un argument entier ainsi qu'un constructeur pour la classe `ListeDEntiers` qui construit la liste vide.
2. Écrire dans la classe `ListeDEntiers` une méthode `ajouter` qui rajoute l'argument entier `n`.
3. Écrire des méthodes `max` et `min` qui renvoient respectivement la plus grande et la plus petite valeur.
4. Écrire une méthode `description` qui renvoie une chaîne de caractères décrivant la liste d'entiers de la façon suivante :
   `Les elements de la liste ont une valeur moyenne de <la moyenne des elements>, le plus grand element de la liste est <le plus grand element> et le plus petit est <le plus petit element>.`