

TD 2

Création de classes

Exercice 1 : le jeu de dominos

Le but est de simuler un jeu de dominos en solitaire.

Un domino est un pion rectangulaire, caractérisé par deux nombres (de 0 à 6). On modélise donc un domino par deux entiers.

Le jeu complet (28 dominos) est modélisé par un tableau.

Au fur et à mesure que le jeu avance, le tableau se remplit.

Question 1/ Ecrire et tester une classe Domino représentant un domino.

Les variables d'instances sont la valeur gauche et la valeur droite du domino (encapsulées).

Les méthodes sont :

- le constructeur Domino(int g, int d),
- int laGauche() et int laDroite(), getters, qui encapsulent l'accès aux variables d'instance,
- void afficher() qui affiche le domino,
- void inverser() qui « retourne » le domino : échange des valeurs gauche et droite.

Question 2/ Ecrire et tester une classe Table représentant la table de jeu.

Les variables d'instance de cette classe sont le tableau de dominos et le nombre de dominos posés. Tous les accès aux variables d'instances sont totalement encapsulés via la méthode : affTable().

Les méthodes sont :

- le constructeur Table (int n) : crée une table pouvant accueillir un jeu de n dominos,
- void initTable() : initialise le jeu avec 1 domino,
- void affTable() : affiche le jeu,
- int recherche(int g, int d) qui délivre :
 - ✓ -1 si la pose du domino (g, d) est possible à gauche du jeu,
 - ✓ 1 si elle est possible à droite,
 - ✓ 0 si elle est impossible,
- void poserAGauche(int g, int d) : pose le domino (g, d) à gauche,
- void poserADroite(int g, int d) : pose le domino (g, d) à droite.

Pour ces deux dernières méthodes, le domino est créé et rajouté (à gauche ou à droite) dans le tableau.

Remarque : le programme ne vérifiera pas si un domino saisi par l'utilisateur est déjà sur la table ou si un domino posé est posé au-delà de la 28^{ème} position.

Question 3/ Ecrire et tester une classe Jeu contenant le *main()*.

Le programme :

- réserve la table pour 28 dominos,
- initialise le jeu avec 1 domino, saisi par l'utilisateur (la vérification des valeurs de 0 à 6 ne sera réalisée que si vous en avez le temps)
- affiche l'état initial,
- lance le jeu proprement dit. L'utilisateur saisit les deux nombres caractérisant le domino qu'il veut poser.

Après chaque domino joué, le programme affiche l'état (provisoire) du jeu.

Pour le test de fin de partie, le jeu s'arrêtera quand l'utilisateur répond oui à la question « voulez-vous arrêter ? » (Utilisation de la méthode d'instance *compareTo* de la classe *String*).

COMPLEMENTS OPTIONNELS

Nous allons traiter ici des améliorations, afin d'intégrer des règles supplémentaires réelles du jeu de dominos.

Le jeu dispose d'une **pioche**, composée en début de partie de l'ensemble des dominos. Au début du jeu, chaque joueur pioche 7 dominos au hasard constituant sa main.

Pour démarrer le jeu, le joueur détenant le double le plus fort dans sa main le pose sur la table. A défaut d'un double, c'est le domino le plus fort qui est posé.

Lorsqu'un joueur ne peut placer aucun de ses dominos, il doit tirer 1 ou plusieurs dominos de la pioche jusqu'à ce qu'il puisse jouer. S'il n'y a plus de dominos à piocher, le joueur passera son tour.

Question 4/ Ecrire et tester une classe Pioche

La variable d'instance de cette classe est une ArrayList de dominos.

Les méthodes sont :

- le constructeur `Pioche(int n)` : crée une ArrayList pouvant accueillir un ensemble de n dominos
- `void initPioche()` : initialise l'ArrayList avec tous les dominos en début de partie
- `Domino piocherDomino()` : délivre 1 domino au hasard dans la pioche

Question 5/ Ecrire et tester une classe MainJoueur

La variable d'instance de cette classe est une ArrayList de dominos.

Les méthodes sont :

- le constructeur `MainJoueur()` : crée une ArrayList pouvant accueillir un ensemble de dominos.
- `void initMain(Pioche p)` : initialise l'ArrayList avec 7 dominos tirés au hasard dans la pioche p en début de partie.
- `int chercheIndiceDoubleDom()` : retourne l'indice du plus gros double d'une main d'un joueur ou -1 s'il n'y a pas de double.
- `int chercheIndiceGrosDom()` : retourne l'indice du plus gros domino d'une main
Cette méthode utilisera la méthode d'instance `int nbPointsDomino()`, à ajouter dans la classe Domino, qui retourne le nombre total de points d'un domino
- `void afficheMain()` : affiche la main d'un joueur

Question 6/ Modifier la classe Jeu

Intégrer dans cette classe les nouvelles règles en début de jeu, en intégrant 2 joueurs.

Poursuivre, sur votre temps libre, la modification du jeu pour intégrer l'ensemble des règles du domino.