

Section	SN2
Date	30/11/2023
Enseignant	M.MALDONADO
Matière	Langage PYTHON

PYTHON

L'objectif de ce devoir est de mettre en place les éléments permettant la programmation du jeu Domino.

Le sujet est volontairement plus long que ce qu'un étudiant moyen peut arriver à faire en 2h.

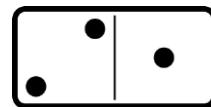
Vous rendrez (sur myLearningBox) ce que vous aurez fait, et essaiez de poursuivre à la maison.



CLASSE

Un domino est défini par deux marques (chiffres de 0 à 6). Sur l'exemple, 2-1

Pour l'exercice, ces deux marques seront deux champs `_marqueGauche` et `_marqueDroite` de la classe Domino. Sur l'exemple, `_marqueGauche` vaut 2, et `_marqueDroite` vaut 1.



Deux accesseurs en lecture `getMarqueGauche()` et `getMarqueDroite()` permettent de renvoyer la valeur du champ.

Le **constructeur** de la classe prend en paramètre deux entiers et initialise les deux champs avec.

Un second constructeur (**constructeur par défaut**) sera à envisager : celui-ci initialisera les deux marques à 0.

La valeur d'un domino est la somme de ses deux marques (3 sur l'exemple). La méthode **valeur()** a pour objectif de retourner ce calcul.

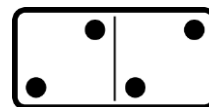
Bien évidemment, si l'on inverse les deux marques d'un domino, on obtient un domino équivalent.

La méthode **inverser()** inversera les deux marques du domino (sur l'exemple, 1-2).

La méthode **estEquivalent()** renverra vrai si le domino courant est équivalent au domino passé en paramètre.

Un double est un domino pour lequel les deux marques sont égales.

La méthode **estDouble()** permettra de tester si c'est le cas et renverra vrai dans l'affirmative, faux sinon.



Pour la méthode **afficher()**, on affichera les deux marques du domino entre crochets, séparés par des espaces et le caractère pipe (|). Dans le cas d'un zéro, un espace sera affiché à la place du chiffre.

Sur l'exemple, cela donnerait [2 | 1], ou [6 |] pour le domino 6-0.

Question 1:

Déclarez la classe Domino.

Vous munirez cette classe des méthodes décrites dans le préambule.

Question 2:

Complétez la classe Domino en écrivant le corps des méthodes :

- les 2 constructeurs
- les 2 accesseurs
- les 5 autres méthodes

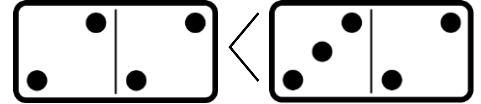
Question 3:

Ecrivez dans un fichier **Main.py** un main qui :

- crée le domino 2-1, affiche ce domino et sa valeur
- l'inverse et l'affiche à nouveau
-

OPERATEURS

On souhaite pouvoir comparer deux dominos sur la base de leur valeur :
 Sur l'exemple, on a $2-2 < 3-2$ car $\text{valeur}(2-2) = 4 < \text{valeur}(3-2) = 5$



On va pour cela définir les opérateurs $<$, $>$ et $==$

Question 4:

Ajoutez dans la classe les trois **opérateurs** $<$, $>$ et $==$

Ces trois opérateurs prennent en paramètre un deuxième domino, testent les valeurs des deux dominos, et retournent le booléen correspondant au test.

EXCEPTIONS

Un domino ne peut correspondre qu'à deux chiffres compris entre 0 et 6.

Question 5:

Modifier le constructeur de la classe Domino pour qu'il renvoie une **exception** si l'un des deux chiffres fournis en paramètres n'est pas valide (non compris entre 0 et 6).

Question 6:

Ecrivez un main qui saisit les deux marques d'un domino et crée le domino correspondant.
 Vous testerez la saisie de marques possibles (ex : 2-1), et la saisie de marques incorrectes (ex : 7-1)

HERITAGE

Une chaîne de dominos (classe **ChaineDominos**) est définie par la mise bout à bout de plusieurs dominos.

Une chaîne de domino est définie par un **vecteur de dominos** (champ `_chaine` de type `vector`) :

`vector<Domino> _chaine ;`



Une chaîne de domino est initialisée :

- Soit à une chaîne vide (**constructeur par défaut**).
- Soit à une chaîne possédant un seul domino (**constructeur prenant en paramètre un domino**)

Par de nombreux aspects, une chaîne de dominos peut être vue comme un « mega » domino :

- Comme un domino, il a une **marque gauche** (respectivement une **marque droite**)
il s'agit de la marque gauche (respectivement droite) de son premier (respectivement dernier) domino.
Sur l'exemple, la marque gauche de la chaîne de dominos vaut 6. Sa marque droite vaut 2
- Comme un domino, il a une **valeur**
il s'agit de la somme des valeurs des dominos qui le composent.
Sur l'exemple, la valeur de la chaîne de dominos est 42.
- Comme pour un domino, on peut tester si la chaîne est **double** (ses marques gauche et droite sont égales)
Sur l'exemple, elle n'est pas double
- Comme un domino, on peut **l'afficher**
ça consiste à afficher chaque domino de la chaîne, en les séparant par un trait d'union.
Sur l'exemple, cela donne [6 | 3] - [3 | 5] - [5 | 2] - [2 | 4] - [4 | 3] - [3 | 2]

Remarque : on pourrait également imaginer

- une méthode `inverser()` qui inverse tous les dominos de la chaînes

- une méthode `estEquivalent()` qui regarde si deux chaînes contiennent les mêmes dominos et ont les mêmes marques

Ces deux méthodes ne sont pas demandées.

Ces similitudes incitent à créer la classe **ChaineDominos** par **héritage** de la classe **Domino**.

Question 7:

Déclarer dans un fichier `ChaineDominos` la classe **ChaineDominos**, sous-classe de la classe **Domino**

Cette classe intègre un vecteur de dominos, et les méthodes :

- les **deux constructeurs**
- accesseurs `getMarqueGauche()` et `getMarqueDroite()`
- `valeur()`, `estDouble()` et `afficher()`

Question 8:

Ecrivez le corps de ces méthodes

Bonus Track

On souhaite pouvoir ajouter un domino à une chaîne, de la manière suivante :

ChaîneDomino uneChaîne ;
Domino unDomino ;

uneChaîne = uneChaîne + unDomino ; // ajoute le domino à droite de la chaîne

ou

uneChaîne = unDomino + uneChaîne; // ajoute le domino à gauche de la chaîne

Question 9:

Ajouter dans la classe ChaîneDomino deux fonction amies :

opérateur + qui prend en paramètre un **domino** et une **chaîne de domino**, et retourne la chaîne de domino

opérateur + qui prend en paramètre une **chaîne de domino** et un **domino**, et retourne la chaîne de domino

Si l'ajout n'est pas possible (marques différentes) une exception sera levée.

Question 10:

Ecrivez un main qui crée la chaîne de dominos donnée en exemple sur la page précédente, et qui l'affiche.