Jeu de 52 cartes

Le jeu de cartes français ou jeu de 52 cartes est un jeu de cartes organisées en deux couleurs : noir et rouge et en quatre enseignes françaises : trèfle, carreau, cœur, pique Les « enseignes » sont plus généralement appelées couleurs. Il comporte 54 cartes à jouer, dont deux jokers. Le jeu de cartes français est originaire de Rouen et Lyon au XVe siècle.

Il s'agit de simuler un jeu de 52 cartes sans joker à l'aide de la programmation POO.

Les questions suivantes utilisent le programme « `carte.py` » à compléter.

1 Initialisation de la classe « `JeuDeCartes` »

Compléter la méthode `def __init__(self):` permettant d'initialiser une liste de tuple contenant la valeur de la carte dans l'ordre croissant et le numéro de la couleur.

Indice	0	1	2	3	4	5	6	7	8	9	10	11	12
Valeur	2	3	4	5	6	7	8	9	10	Valet	Dame	Roi	As

Indice	0	1	2	3
Couleur	Pique	Trèfle	Carreau	Cœur

Vérification :

```
jeux = JeuDeCartes()
print(jeux.carte)
```

```
#résultat dans la console
[(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10, 0), (11, 0), (12, 0),
 (0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1),
 (0, 2), (1, 2), (2, 2), (3, 2), (4, 2), (5, 2), (6, 2), (7, 2), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2),
 (0, 3), (1, 3), (2, 3), (3, 3), (4, 3), (5, 3), (6, 3), (7, 3), (8, 3), (9, 3), (10, 3), (11, 3), (12, 3)]
```

2 Affichage du nom de la carte

Compléter la méthode `def nomCarte(self, c):` affichant le nom de la carte à partir du tuple.

Vérification :

`jeux = JeuDeCartes()`
`jeux.nomCarte((1, 2))`
`jeux.nomCarte((12, 1))`
	`#résultat dans la console`
	`3 de Carreau`
	`as de Trèfle`

3 Battre les cartes

Compléter la méthode `def battre(self):` permettant de mélanger les cartes.

Aide : utiliser la méthode shuffle de la classe random

Vérification :

`jeux = JeuDeCartes()`
`jeux.battre()`
`print(jeux.carte)`	

`#résultat dans la console` `[(12, 3), (4, 1), (11, 2), (3, 2), (8, 0), (8, 1), (1, 0), (10, 0), (9, 3), (2, 1), (0, 2), (9, 2), (11, 3), (10, 2), (1, 1), (12, 0), (2, 0), (5, 0), (12, 2), (7, 0), (0, 0), (11, 1), (7, 2), (9, 1), (12, 1), (3, 0), (1, 3), (6, 1), (3, 1), (6, 2), (4, 2), (0, 1), (7, 1), (5, 1), (9, 0), (2, 3), (2, 2), (4, 3), (1, 2), (7, 3), (10, 3), (0, 3), (10, 1), (6, 0), (5, 3), (8, 3), (8, 2), (3, 3), (11, 0), (5, 2), (6, 3), (4, 0)]`	

4 Tirer une carte

4.1 Compléter la méthode `def tirer(self):` permettant de tirer la carte à l'indice 0 et de la supprimer de la liste `self.carte`.
La méthode retourne le tuple correspondant à la carte. Si toutes les cartes sont tirées, il faudra retourner « `none` ».

Pour supprimer la carte à l'indice 0 dans une liste, utiliser la fonction suivante :
`del(self.carte[0])`

Vérification :

```
jeux = JeuDeCartes()
jeux.battre()
print(jeux.carte)
c = jeux.tirer()
print(jeux.nomCarte(c))
print()
print(jeux.carte)
```

Le tuple (8,3) au début de la liste est retiré après avoir faire un tirage.

```
#résultat dans la console
[(8, 3), (6, 0), (3, 3), (1, 0), (5, 2), (6, 3), (10, 2), (0, 3), (4, 0), (2, 3), (8, 0), (0, 1), (1, 1),
(10, 0), (8, 2), (0, 2), (5, 1), (3, 2), (12, 2), (11, 3), (8, 1), (11, 1), (6, 2), (10, 3), (12, 0), (0, 0),
(5, 3), (12, 1), (11, 2), (11, 0), (3, 1), (9, 3), (9, 0), (4, 3), (7, 3), (10, 1), (2, 0), (3, 0), (4, 2),
(7, 2), (9, 1), (1, 3), (7, 1), (2, 1), (1, 2), (4, 1), (2, 2), (6, 1), (12, 3), (9, 2), (7, 0), (5, 0)]
10 de Coeur
None

[(6, 0), (3, 3), (1, 0), (5, 2), (6, 3), (10, 2), (0, 3), (4, 0), (2, 3), (8, 0), (0, 1), (1, 1), (10, 0),
(8, 2), (0, 2), (5, 1), (3, 2), (12, 2), (11, 3), (8, 1), (11, 1), (6, 2), (10, 3), (12, 0), (0, 0), (5, 3),
(12, 1), (11, 2), (11, 0), (3, 1), (9, 3), (9, 0), (4, 3), (7, 3), (10, 1), (2, 0), (3, 0), (4, 2), (7, 2),
(9, 1), (1, 3), (7, 1), (2, 1), (1, 2), (4, 1), (2, 2), (6, 1), (12, 3), (9, 2), (7, 0), (5, 0)]
```

4.2 Vérifier la méthode « `tirer` » en retirant toutes les cartes de la liste une par une avec une boucle pour.

```
jeux = JeuDeCartes()
for n in range(53):
    c = jeux.tirer()
#A compléter
```

```
#résultat dans la console

dame de Pique
valet de Coeur
3 de Trèfle
roi de Trèfle
valet de Carreau
...
...
7 de Pique
5 de Trèfle
2 de Trèfle
valet de Trèfle
10 de Pique
Plus de cartes
```

5 Simuler un jeu de bataille entre 2 joueurs (robots)

Le joueur remporte un point lorsqu'il remporte le pli.

```
jeuA = JeuDeCartes()
jeuB = JeuDeCartes()
jeuA.battre()
jeuB.battre()
comptA = 0
comptB = 0
#A compléter
```

```
#résultat dans la console
Score :
Joueur A = 23 -------- Joueur B = 27
```

6 Affichage des cartes

En utilisant les programmes de l'année de 1ere et les images png (fichier « data.zip ») représentant les différentes cartes, effectuer un rendu graphique de la simulation de la bataille.

7 Adaptation en réseau

En utilisant le programme de chat textuel réalisé en 1ere, effectuer une partie de bataille en réseau local entre 2 PC.

8 Défi

Ajouter une classe fille « Belote » héritant de la classe « **`JeuDeCartes`** » permettant de gérer le jeu de la belote ou un autre jeu similaire.