

Projet 2 - Un jeu de cartes

1 Objectifs

L'objectif de ce projet est de réaliser un **jeu de bataille** comprenant 4 classes dans des 4 fichiers distincts et un programme principal utilisant ces classes.

Vous devrez rendre :

- ▷ un fichier présentant votre projet(sous *Word* ou *Google Doc*) ;
- ▷ un fichier *Python* pour chaque classe contenant la définition de la classe, des commentaires adaptés et la fonction de test de la classe ;
- ▷ le fichier principal du projet avec vos noms et prénoms ;
- ▷ présentation : une vidéo de présentation orale de 5 minutes incluant un diaporama.

Exemple de sortie en cours d'exécution :

```
# Dans la console PYTHON
Debut: 26 26
La carte du joueur Toto est: Dame de TREFLE
La carte du joueur Dupont est: 3 de COEUR
Le joueur Toto a gagne ce tour
Nombre de cartes en sortie du dernier coup:
NbCartes de Toto : 27
NbCartes de Dupont : 25
La carte du joueur Toto est: 2 de TREFLE
La carte du joueur Dupont est: As de TREFLE
Le joueur Dupont a gagne ce tour
Nombre de cartes en sortie du dernier coup:
NbCartes de Toto : 26
NbCartes de Dupont : 26
La carte du joueur Toto est: 10 de COEUR
La carte du joueur Dupont est: Roi de TREFLE
Le joueur Dupont a gagne ce tour
Nombre de cartes en sortie du dernier coup:
NbCartes de Toto : 25
NbCartes de Dupont : 27
```

2 Description de la classe Carte

2.1 Détails de l'implémentation

```
1 # Dans l'editeur PYTHON : fichier carte.py
2
3 # Variables globales
4 couleurs = ('CARREAU', 'COEUR', 'TREFLE', 'PIQUE')
5 noms = ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'Valet', 'Dame', 'Roi', 'As']
6 valeurs = {'2':2, '3':3, '4':4, '5':5, '6':6, '7':7, '8':8, '9':9, '10':10,
7            'Valet':11, 'Dame':12, 'Roi':13, 'As':14}
8
9 # Classe Carte
10 class Carte:
11     def __init__(self, nom, couleur):
12         # Affectation des attributs nom et couleur avec controle.
13         self.nom = nom
14         self.couleur = couleur
15         self.valeur = ...
```

```

16
17
18 ##### Definition des methodes d'instances avec controle #####
19     def setNom(self, nom): # setter
20         ...
21     def getNom(self): # getter
22         ...
23     def getCouleur(self): # getterDescription
24         ...
25     def getValeur(self): # getter
26         ...
27     def egalite(self, carte):
28         ''' Renvoie True si les cartes ont meme valeur, False sinon
29         carte: Objet de type Carte '''
30
31     def estSuperieureA(self, carte):
32         ''' Renvoie True si la valeur de self est superieure a celle de
33         carte, False sinon
34         carte: Objet de type Carte '''
35
36     def estInferieureeA(self, carte):
37         ''' Renvoie True si la valeur de self est inferieure a celle de
38         carte, False sinon
39         carte: Objet de type Carte '''

```

2.2 Fonction test et vérification de la classe Carte.

```

1 # Dans l'editeur PYTHON : fichier carte.py
2
3 ##### Test de la classe Carte #####
4 def testCarte():
5     valetCoeur = Carte('Valet', 'COEUR')
6     print('Nom:', valetCoeur.getNom())
7     print('Couleur:', valetCoeur.getCouleur())
8     print('Valeur:', valetCoeur.getValeur())
9     valetCoeur.setNom('Dame')
10    print('Nom modifie:', valetCoeur.getNom())
11    print('Valeur modifiee:', valetCoeur.getValeur())
12
13 # Essai des exceptions: cette instruction conduit a une erreur
14 dameCarreau = Carte('Dame', 'COooEUR')

```

Ce qui doit nous donner :

```

# Dans la console PYTHON
Nom: Valet
Couleur: COEUR
Valeur: 11
Nom de fichier image: COEURValet.png
Nom modifie: Dame
Valeur modifiee: 12
La couleur de la carte est incorrecte: COooEUR

```

3 Description de la classe JeuCartes

3.1 Détail de l'implémentation de la classe JeuCartes

```

1  # Dans l'editeur PYTHON : fichier jeucartes.py
2
3  from carte import *    # Il faut importer la classe Carte et les
    variables globales
4  import random          # Necessary pour melanger le jeu
5
6  class JeuCartes():
7      def __init__(self, nbCartes=52):
8          # Le jeu doit comporter 32 ou 52 cartes, effectuer un controle
9
10         self.jeu = []    # self.jeu est une liste des self.nbCartes
11         ...              # a completer
12
13     ##### Definition des methodes d'instances #####
14
15     def getTailleJeu(self):
16         ''' Fonction qui retourne le nombre de cartes du jeu
17         Valeur retournee: type int '''
18     def creerJeu(self): # utilise des objet
19         ''' Cree la liste des cartes de l'attribut self.jeu '''
20     def getJeu(self):
21         ''' Renvoie la liste des cartes correspondant a l'attribut
22         self.jeu '''
23     def melanger(self): # utiliser le module random ...
24         ''' Melange sur place les cartes de la liste des cartes associee
25         au champ self.jeu '''
26     def distribuerCarte(self):
27         ''' Cette fonction permet de distribuer une carte a un joueur.
28         Elle retourne la carte
29         Valeur retournee: Objet de type Carte '''
30     def distribuerJeu(self, nbJoueurs, nbCartes):
31         ''' Cette methode distribue nbCartes a chacun des nbJoueurs,
32         ... '''

```

3.2 Fonction test et vérification de la classe JeuCartes

```

1  # Dans l'editeur PYTHON : fichier jeucartes.py
2  ##### Test de la classe JeuCartes #####
3
4  def testJeuCartes():
5      jeu52 = JeuCartes(52)
6      jeu52.melanger()
7
8      L=jeu52.getJeu()
9      carte= L[2]          # le 3e carte
10     print('Nom:', carte.getNom())
11     print('Couleur:', carte.getCouleur())
12     print('Valeur:', carte.getValeur())
13
14     # Distribution de 4 cartes a 3 joueurs
15     distribution_3j_4c = jeu52.distribuerJeu(3, 4)
16     for i in range(3):
17         print('Joueur', i+1, ':')
18         listeCartes = distribution_3j_4c[i]
19         for c in listeCartes:
20             print(c.getNom(), 'de', c.getCouleur())

```

```
21 # Distribution de 10 cartes a 6 joueurs pour geneer une exception
22 (6X10 > 52)
23 distribution_6_joueurs_10_cartes_par_joueur =
    jeu52.distribuerJeu(6, 10)
```

Ce qui doit nous donner :

```
# Dans la console PYTHON
Nom: Dame
Couleur: COEUR
Valeur: 12
Joueur 1 :
6 de TREFLE
Dame de CARREAU
6 de CARREAU
Dame de TREFLE
Joueur 2 :
3 de COEUR
8 de TREFLE
Valet de COEUR
5 de TREFLE
Joueur 3 :
Roi de TREFLE
As de COEUR
10 de PIQUE
9 de PIQUE
Pas assez de cartes dans le jeu.
```

4 Description de la classe Joueur

4.1 Détail de l'implémentation de la classe Joueur

Créer une classe **Joueur** ayant les attributs suivants :

1. **nom** : Nom du joueur ;
2. **nbCartes** : Correspond au nombre de cartes dans la main du joueur ;
3. **mainJoueur** : Liste des cartes(objets de type Carte) dans la main du joueur.

Cette classe devra implémenter les méthodes suivantes :

1. **setMain()** : Définit la main du joueur, donc la liste de ses cartes au début du jeu ;
2. **getNom()** : Accesseur de l'attribut nom ;
3. **getNbCartes()** : Accesseur du champ nbCartes ;
4. **jouerCarte()** : Enlève et renvoie la dernière carte (objet de type Carte) de la main du joueur pour la jouer, ou retourne None s'il n'y a plus de cartes dans la main du joueur ;
5. **insérerMain()** : Fonction qui insère les cartes de la liste des cartes gagnées dans la main du Joueur

4.2 Fonction test et vérification

A faire.

5 Description de la classe Bataille

La classe **Bataille** doit instancier un jeu de cartes, deux joueurs et implémenter la méthode jouer.

Tester votre classe.

A vous de faire le programme principal.