Master Data Science 1

modifié le 22 octobre 2022 a u8:14

Poker

Contents

Il s'agit de créer des classes pour modéliser une partie de pocker.

1. Une carte

Compléter la classe `Carte` suivante :

PYTHON

```
from random import randrange

class Carte():
  valeurs = list(range(7,11)) + ['Valet','Dame','Roi','As']
  couleurs = ['Coeur','Carreau','Pique','Trèfle']

  def __init__(self,v=None,c=None):
    if v:
      if not v in Carte.valeurs:
        raise ValueError(f"{v}: valeur incorrecte")
    else:
      v = Carte.valeurs[randrange(0,len(Carte.valeurs))]

    if c:
      if not c in Carte.couleurs:
        raise ValueError(f"{c}: couleur incorrecte")
    else:
      c = Carte.couleurs[randrange(0,len(Carte.couleurs))]

    self.couleur = c
    self.valeur = v
```

de façon à pouvoir écrire :

PYTHON

```
print(Carte())                    # -> Valet de Trèfle (par exemple)
print(Carte())                    # -> Dame de Coeur (par exemple)
print(Carte())                    # -> 8 de Carreau (par exemple)
```

```
carte1 = Carte(7,"Coeur")
print(carte1)                       # -> 7 de Coeur
carte2 = Carte("Valet","pic")       # -> ValueError: pic: couleur incorrecte
carte3 = Carte("Valet","Pique")
print(carte3)                       # -> Valet de Pique
```

2. Des cartes

Compléter la classe `Cartes` suivante :

PYTHON

```python
class Cartes():
  def __str__(self):
    return ", ".join([str(c) for c in self.cartes])

  def ajoute(self,c):
    self.cartes += [c]
```

de façon à pouvoir écrire :

PYTHON

```python
# 3 cartes
carte1 = Carte("As","Trèfle")
carte2 = Carte(7,"Coeur")
carte3 = Carte("Valet","Pique")

# un ensemble de cartes (vide au départ)
des_cartes = Cartes()

# on y ajoute les 3 cartes
for une_carte in [carte1,carte2,carte3]:
  des_cartes.ajoute(une_carte)

print(f"{des_cartes=}")

# on le clone
les_memes = Cartes(des_cartes)
print(f"{les_memes=}")

# on pioche dedans tant que l'on peut
try:
  while True:
    print(f"{des_cartes.pioche()=}")
    print(f"{des_cartes=}")
    print(f"{les_memes=}")
except ValueError as e:
  # traceback.print_exc(file=sys.stdout)
```

```
  print(e)
print("fin de programme")
```

pour obtenir :

```
des_cartes=As de Trèfle, 7 de Coeur, Valet de Pique
les_memes=As de Trèfle, 7 de Coeur, Valet de Pique
des_cartes.pioche()=As de Trèfle
des_cartes=7 de Coeur, Valet de Pique
les_memes=As de Trèfle, 7 de Coeur, Valet de Pique
des_cartes.pioche()=Valet de Pique
des_cartes=7 de Coeur
des_cartes.pioche()=7 de Coeur
des_cartes=
les_memes=As de Trèfle, 7 de Coeur, Valet de Pique
pioche vide !
fin de programme
```

3. Le jeu

Créer une classe `Jeu` qui, utilisée comme :

PYTHON

```
un_jeu = Jeu()
print(f"{un_jeu=}")
un_jeu -= Carte('Valet','Coeur')
un_jeu -= Carte('As','Pique')
un_jeu -= Carte(10,'Trèfle')
print(f"{un_jeu=})
```

donne :

```
un_jeu=7 de Coeur, 7 de Carreau, 7 de Pique, 7 de Trèfle, 8 de Coeur, 8 de
Carreau, 8 de Pique, 8 de Trèfle, 9 de Coeur, 9 de Carreau, 9 de Pique, 9 de
Trèfle, 10 de Coeur, 10 de Carreau, 10 de Pique, 10 de Trèfle, Valet de Coeur,
Valet de Carreau, Valet de Pique, Valet de Trèfle, Dame de Coeur, Dame de
Carreau, Dame de Pique, Dame de Trèfle, Roi de Coeur, Roi de Carreau, Roi de
Pique, Roi de Trèfle, As de Coeur, As de Carreau, As de Pique, As de Trèfle
un_jeu=7 de Coeur, 7 de Carreau, 7 de Pique, 7 de Trèfle, 8 de Coeur, 8 de
Carreau, 8 de Pique, 8 de Trèfle, 9 de Coeur, 9 de Carreau, 9 de Pique, 9 de
Trèfle, 10 de Coeur, 10 de Carreau, 10 de Pique, Valet de Carreau, Valet de
Pique, Valet de Trèfle, Dame de Coeur, Dame de Carreau, Dame de Pique, Dame de
Trèfle, Roi de Coeur, Roi de Carreau, Roi de Pique, Roi de Trèfle, As de Coeur,
As de Carreau, As de Trèfle
```

(il est possible d'ajuster les classes `Carte` et `Cartes` si besoin)

4. Une main

Créer une classe `Main` qui, utilisée comme :

```python
le_jeu = Jeu()

# on crée 2 mains vides
ma_main = Main(le_jeu)
ta_main = Main(le_jeu)
print(f"{ma_main=}")
print(f"{ta_main=}")

# on y ajoute 3 cartes
for i in range(3):
  ma_main.complete()
  print(f"{ma_main=}")
  ta_main.complete()
  print(f"{ta_main=}")
print(f"{le_jeu=}")

# on tente d'ajouter 25 cartes à la première
try:
  for i in range(25):
    ma_main.complete()
except ValueError as e:
  print(e)

# on tente dajouter 25 cartes à la seconde
try:
  for i in range(25):
    ta_main.complete()
except ValueError as e:
  # traceback.print_exc(file=sys.stdout)
  print(e)

print("fin de programme")
```

donne :

```
ma_main=
ta_main=
ma_main=As de Pique
ta_main=Dame de Carreau
ma_main=As de Pique, 10 de Pique
ta_main=Dame de Carreau, Roi de Carreau
ma_main=As de Pique, 10 de Pique, Dame de Trèfle
ta_main=Dame de Carreau, Roi de Carreau, 9 de Trèfle
le_jeu=7 de Coeur, 7 de Carreau, 7 de Pique, 7 de Trèfle, 8 de Coeur, 8 de
Carreau, 8 de Pique, 8 de Trèfle, 9 de Coeur, 9 de Carreau, 9 de Pique, 10 de
```

```
Coeur, 10 de Carreau, 10 de Trèfle, Valet de Coeur, Valet de Carreau, Valet de
Pique, Valet de Trèfle, Dame de Coeur, Dame de Pique, Roi de Coeur, Roi de
Pique, Roi de Trèfle, As de Coeur, As de Carreau, As de Trèfle
plus de carte pour compléter la main [Dame de Carreau, Roi de Carreau, 9 de
Trèfle]
fin de programme
```

5. Carré

Un carré est 4 cartes de valeurs identiques. Créer la classe `Carre` dont une instance se crée à partir d'une main, et qui s'initialise avec un carré contenu dans la main s'il y en a un, ou lance une exception sinon. Faire en sorte qu'on puisse retirer un carré d'une main. Exemple :

PYTHON

```python
le_jeu = Jeu()

# une main de 25 cartes
une_main = Main(le_jeu)
for i in range(25):
  une_main.complete()
print(f"{une_main=}")

# recheche tous les carrés contenus dans la main
while True:
  try:
    un_carre = Carre(une_main)              # essai de créer un carré
    # si on est là, c'est qu'un carré a été créé
    print(f"{un_carre=}, {une_main=}")
    une_main -= un_carre                    # on l'enlève de la main
  except RuntimeError as e:
    # si on est là, c'est qu'un carré n'a pas pu être créé
    print(e)
    break

print("fin de programme")
```

pourrait donner :

```
une_main=As de Trèfle, 9 de Coeur, Valet de Coeur, 7 de Coeur, 10 de Carreau, 8
de Trèfle, Dame de Coeur, Dame de Pique, Roi de Trèfle, 10 de Trèfle, 9 de
Pique, 7 de Trèfle, 10 de Pique, 9 de Trèfle, 9 de Carreau, 7 de Pique, 8 de
Pique, 7 de Carreau, Valet de Trèfle, Roi de Pique, 8 de Carreau, Roi de
Carreau, 10 de Coeur, 8 de Coeur, Valet de Carreau
un_carre=carré de 7, une_main=As de Trèfle, 9 de Coeur, Valet de Coeur, 7 de
Coeur, 10 de Carreau, 8 de Trèfle, Dame de Coeur, Dame de Pique, Roi de Trèfle,
10 de Trèfle, 9 de Pique, 7 de Trèfle, 10 de Pique, 9 de Trèfle, 9 de Carreau, 7
de Pique, 8 de Pique, 7 de Carreau, Valet de Trèfle, Roi de Pique, 8 de Carreau,
Roi de Carreau, 10 de Coeur, 8 de Coeur, Valet de Carreau
```

```
un_carre=carré de 8, une_main=As de Trèfle, 9 de Coeur, Valet de Coeur, 10 de Carreau, 8 de Trèfle, Dame de Coeur, Dame de Pique, Roi de Trèfle, 10 de Trèfle, 9 de Pique, 10 de Pique, 9 de Trèfle, 9 de Carreau, 8 de Pique, Valet de Trèfle, Roi de Pique, 8 de Carreau, Roi de Carreau, 10 de Coeur, 8 de Coeur, Valet de Carreau
un_carre=carré de 9, une_main=As de Trèfle, 9 de Coeur, Valet de Coeur, 10 de Carreau, Dame de Coeur, Dame de Pique, Roi de Trèfle, 10 de Trèfle, 9 de Pique, 10 de Pique, 9 de Trèfle, 9 de Carreau, Valet de Trèfle, Roi de Pique, Roi de Carreau, 10 de Coeur, Valet de Carreau
un_carre=carré de 10, une_main=As de Trèfle, Valet de Coeur, 10 de Carreau, Dame de Coeur, Dame de Pique, Roi de Trèfle, 10 de Trèfle, 10 de Pique, Valet de Trèfle, Roi de Pique, Roi de Carreau, 10 de Coeur, Valet de Carreau
pas de carré dans [As de Trèfle, Valet de Coeur, Dame de Coeur, Dame de Pique, Roi de Trèfle, Valet de Trèfle, Roi de Pique, Roi de Carreau, Valet de Carreau]
fin de programme
```

6. Quinte

Une quinte est une suite de 5 cartes d'une même couleur. Créer la classe Quinte dont une instance se crée à partir d'une main, et qui s'initialise avec une quinte contenue dans la main s'il y en a une, ou lance une exception sinon. Faire en sorte qu'on puisse retirer une quinte d'une main.

Contents