td1a_correction_session6

December 23, 2020

1 1A.2 - Classes, héritage (correction)

Correction.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.0.1 Exercice 1 : pièce normale

[2]:
```
import random
class Piece :
    def tirage_aleatoire(self, precedent) :
        return random.randint(0,1)
    def moyenne_tirage(self, n):
        tirage = [ ]
        for i in range (n) :
            precedent = tirage[-1] if i > 0 else None
            tirage.append( self.tirage_aleatoire (precedent) )
        s = sum(tirage)
        return s * 1.0 / len(tirage)

p = Piece()
print (p.moyenne_tirage(100))
```

```
0.48
```

1.0.2 Exercice 2 : pièce truquée

[3]:
```
class PieceTruquee (Piece) :
    def tirage_aleatoire(self, precedent) :
        if precedent == None or precedent == 1 :
            return random.randint(0,1)
        else :
            return 1 if random.randint(0,9) >= 3 else 0

p = PieceTruquee()
print (p.moyenne_tirage(100))
```

```
0.58
```

1.0.3 Exercice 3 : Pièce mixte

[4]:
```
class PieceTruqueeMix (PieceTruquee) :
    def tirage_aleatoire(self, precedent) :
        if random.randint(0,1) == 0 :
            return Piece.tirage_aleatoire(self, precedent)
        else :
            return PieceTruquee.tirage_aleatoire(self, precedent)

p = PieceTruqueeMix()
print (p.moyenne_tirage(100))
```

```
0.67
```

1.0.4 Exercice 4 : pièce mixte avec des fonctions

[5]:
```
# ce qui vient de l'énoncé
def moyenne_tirage(n, fonction):
    """
    cette fonction fait la moyenne des résultats produits par la fonction passée en␣
 ↪argument
    """
    tirage = [ ]
    for i in range (n) :
        precedent = tirage[-1] if i > 0 else None
        tirage.append( fonction (precedent) )
    s = sum(tirage)
    return s * 1.0 / len(tirage)

def truquee (precedent) :
    if precedent == None or precedent == 1 :
        return random.randint(0,1)
    else :
        return 1 if random.randint(0,9) >= 3 else 0

# la partie ajoutée pour la correction
print (moyenne_tirage(100, lambda v : random.randint(0,1) if  random.randint(0,1) == 0␣
 ↪\
                      else truquee(v)))
```

```
0.51
```