111A Introduction to Computer and Computer Science

Homework Assignment #8

Due: 12/05 12:00:00

In this homework assignment, you will practice how to apply the concept of object-oriented programming to solve a practical issue. As you have learned all the related knowledge during the lectures, you should have the ability to accomplish this.

Problem #1: Create a Dice

There are several iconic games in casino, at least one of them is related to a simple, tiny, but elegant element - dice. In this homework, we are going to create a "**Dice**" object to represent a fair cubic dice, which has several instance and method attributes:

1. An instance attribute "**current_state**" to represent its current point. The initial state should be "None".
2. An instance attribute "**dice_id**" to represent its identity (btw, how do we check the identification of each dice during the game?).
3. A method attribute "**roll**" to randomly change the "**current_state**". The state should be 1, 2, …, or 6 with uniform probability density function.
4. A method attribute "**get_current_state**" to access the instance attribute **"current_state**".

You should get this result when you execute the corresponding code.

```
dice1 = Dice(1)

print(dice1)

dice1.roll()

print(dice1)
```

✓ 0.7s

```
Dice 1: None
Dice 1: 5
```

Here is the sample code:

```
from random import randint

class Dice:
    """A fare dice"""
    def __init__(self, id):
        self.current_state = ???
        self.dice_id = ???

    def __str__(self):
        ???

    def roll(self):
        ???

    def get_current_state(self):
        ???
```

Problem #2: A Person with a Dice box

Let us assume that we have a guest in our casino, holding a dice box with M dices. Please create a "**Person**" object to represent this person, which also has several instance and method attributes:

1. An instance attribute "**name**" to represent her/his identity.
2. An instance attribute "**dicebox**" to represent the dice box holding on her/his hand.
3. A method attribute "**put**" to put M dices into her/his dice box.
4. A method attribute "**check**" to check the status of her/his dice box.
5. A method attribute "**roll**" to represent the event of she/he rolling the dice box. This method should change the state of every dices in her/his dice box.
6. A method attribute "**check_sum**" to calculate and return you the summation of all dices in her/his dice box

Here is the sample code:

```
class Person:
    """A person holding a dice box with arbitrary dices"""
    def __init__(self, name, numDices):
        self.name = ???
        self.dicebox = ???
        self.put(numDices)

    def put(self, numDices):
        ???

    def check(self):
        ???

    def roll(self):
        ???

    def check_sum(self):
        ???
```

You should get this result when you execute the corresponding code.

```
A = Person("AAA", 6)

A.check()
A.roll()
print("Current sum = {}, type: {}".format(A.check_sum(), type(A.check_sum())))

B = Person("BBB", 10)

B.check()
B.roll()
print("Current sum = {}, type: {}".format(B.check_sum(), type(B.check_sum())))
```

✓ 0.3s

```
Current state: [5, 3, 1, 5, 5, 3]
Current sum = 18, type: <class 'int'>
Current state: [4, 2, 5, 6, 1, 6, 5, 5, 4, 5]
Current sum = 39, type: <class 'int'>
```

(Bonus) Problem #3: The Statistics of N Times Rolling

Our curious guest wants to do an experiment that can help her/him to understand the statistics of rolling M dices. Please write a function called "**Rolling_Statistic**" that counts the number of occurrences of summation after N times rolling. For example, your function should return a result that contains the information when rolling 1 dice for 1000 times as shown below:

- Case: 1 dice

Sum	Number of Occurrences
1	199
2	153
3	187
4	213
5	101
6	147

Here is the sample code:

```
def Rolling_Statistic(numDices, timesRoll):
    statistic_result = {}

    ???

    return statistic_result
```

!!!NOTICE!!!

In this homework, you are only allowed to
import the module to implement the uniform probability.
(the random module as shown in the first sample code)

Hand in procedure:

As we had mentioned in the lecture, you should list all your collaborators in your programs. Here is the template:

```
"""
Created on Sun Aug 7 01:23:45 2022

@author: Xi Winnie, student ID

@collaborators: Jane Doe, her student ID
                John Doe, his student ID
"""
```

In this homework, you can either choose packaging all your code in one ".py" file or separating your code into several ".py" file and then using a main script (main_hw8.py) to import them. Please save your code as a "**.zip**", "**.7z**", or "**.rar**" file if you choose the latter one. However, the file name should follow this format:

111A_hw#8_ID.py ..... if you choose the former.

111A_hw#8_ID.zip ..... if you choose the latter.

For example,

111A_hw#8_0123456789.py

111A_hw#8_0123456789.zip

Please be aware. **We are not going to accept any homework file with wrong file name or without signature**. Please double check the content of your file.

Once you have accomplished your works, you can upload your homework to the "E3@NYCU" system. There will be a section for uploading your homework.