

클래스 연습문제

연습문제1

■ 봇과 가위바위보 게임하기

- 컴퓨터와 대결하여 가위바위보 게임을 3선승하는 사람이 이기는 게임프로그램

- random 모듈을 탑재한다.
- Game 이라는 클래스를 생성한다
 - 객체를 생성할 때 3개의 인스턴스 변수를 생성한다.
 - '가위', '바위', '보' 를 저장하는 튜플 변수 : items
 - 사람이 이긴 횟수를 저장하기 위한 변수 : human_count
 - 컴퓨터가 이긴 횟수를 저장하기 위한 변수 : bot_count
- start() 함수 생성
 - while문을 사용하여 양측 모두 3선승을 하기 전까지 무한 반복
 - 사용자에게 가위, 바위, 보를 표준 입력으로 받아 변수 human에 저장. 만약 입력을 잘못하면 에러 메시지를 출력
 - 봇은 난수 0,1,2를 발생시켜 앞에서 선언한 items에서 값을 무작위로 추출하여 변수 bot에 저장. 이때 봇이 선택한 아이템을 확인하기 위해 출력
 - 가위바위보 게임의 결과에 따라 human_count 또는 bot_count가 올라간다.
 - 둘 중 누구라도 3번 이상 이기면 게임 결과를 출력하고 종료한다.

연습문제1

■ 실행결과

컴퓨터와의 가위-바위-보 게임을 시작하겠습니다

=====

가위, 바위, 보 중 하나를 입력하세요! 가위

봇이 [보]를 냈습니다.

당신이 이겼습니다!! :)

=====

가위, 바위, 보 중 하나를 입력하세요! 보

봇이 [바위]를 냈습니다.

당신이 이겼습니다!! :)

=====

가위, 바위, 보 중 하나를 입력하세요! 보

봇이 [바위]를 냈습니다.

당신이 이겼습니다!! :)

축하드립니다!! 봇을 이기셨습니다!! :)

연습문제 2

■ 봇이 고른 숫자 맞추기

- 봇이 발생시킨 정수를 사람이 맞추는 게임

- random 모듈을 탑재한다.
- Game 이라는 클래스를 생성한다
 - 객체를 생성할 때 2개의 인스턴스 변수를 생성한다.
 - 사용자가 입력한 숫자를 저장하기 위한 변수 : human
 - 컴퓨터가 선택한 숫자를 저장하기 위한 변수 : bot
- start() 함수 생성
 - 게임 안내메세지를 출력한 후, 바로 난수를 발생시켜 0부터 9까지의 정수 중 하나를 bot에 저장
 - while문을 사용하여 human 과 bot이 일치할 때 까지 무한 반복한다.
 - 사용자가 0~9까지의 정수 중 하나를 입력하게 유도한다.
 - 입력한 숫자와 봇이 선택한 숫자를 비교하여 결과를 알려주고 다시 반복문을 수행.

연습문제2

■ 실행결과

0~9 정수 중 컴퓨터가 선택한 무작위 숫자를 맞춰 보세요.

=====

0 부터 9까지의 정수 중 하나 선택하세요.5

입력하신 숫자가 봇이 고른 숫자보다 작습니다. 다시 시도하세요.

=====

0 부터 9까지의 정수 중 하나 선택하세요.7

축하드립니다!! 봇이 고른 숫자를 맞추셨습니다!! :)

연습문제3

■ 원을 나타내는 클래스 circle

- 반지름 : radius
- calPerimeter() : 원이 둘레 계산
- calArea() : 원의 면적 계산

■ 실행결과

반지름 : 100 원의 면적 :314125.92 원의 둘레 :628.3184

> > >

연습문제 4

- Cat 클래스 생성 → cat.py 저장

```
class Cat:
    def speak(self):
        print("야옹!")

    def drink(self):
        print("고양이가 우유를 마십니다.")
        print("고양이가 낮잠을 잡니다.")
```

- cat 모듈 импорт
- romeo 객체 : cat 클래스
- speak 와 drink 호출

연습문제 5

■ Cat 클래스에 생성자 정의 및 메소드 수정 :cat2.py로 저장

- 생성자에서 cat 객체의 이름을 저장 : name
- speak와 drink를 수정
- 두개의 cat 객체 생성 : Romeo, Juliet
- 실행해보기

■ 실행결과

```
>>> romeo= cat2.Cat("Romeo")
>>> Juliet = cat2.Cat("Juliet")
>>> Romeo.speak()
Romeo가 야옹합니다.
```


연습문제6

■ 사칙연산 클래스 만들기

■ 사칙연산을 하는 클래스 생성

- 클래스 이름 : FourCal()
- 메서드
 - setdata : 두 숫자를 입력받는 메서드
 - sum : +
 - sub : -
 - mul : *
 - div : /
- 필드
 - First / second : 두개의 숫자

연습문제6

■ 실행결과

```
>>> a = FourCal()
>>> a.setdata(4, 2)
>>> print(a.add())
6
>>> print(a.mul())
8
>>> print(a.sub())
2
>>> print(a.div())
2
```

연습문제 7

■ FourCal 클래스를 상속받아 MoreFourCal 클래스 생성

- a의 b제곱(a^b)을 계산하는 MoreFourCal 클래스

- 두 수의 거듭제곱을 구할 수 있는 pow 메서드를 추가

■ 실행결과

```
>>> a = MoreFourCal(4, 2)
```

```
>>> a.pow()
```

```
16
```

연습문제 8

■ FourCal 클래스를 상속받아 SafeFourCal 클래스 생성

```
>>> a = FourCal(4, 0)
>>> a.div()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    result = self.first / self.second
ZeroDivisionError: division by zero
```

- 0으로 나눌 때 오류가 아닌 0을 돌려주도록 만든다

- SafeFourCal 클래스에 오버라이딩한 div 메서드는 나누는 값이 0인 경우에는 0을 리턴

■ 실행결과

```
>>> a = SafeFourCal(4, 0)
>>> a.div()
0
```

연습문제 9

- 다음과 같이 동작하는 클래스 Calculator를 작성하시오.

```
>>> cal1 = Calculator([1,2,3,4,5])
>>> cal1.sum() # 합계
15
>>> cal1.avg() # 평균
3.0
>>> cal2 = Calculator([6,7,8,9,10])
>>> cal2.sum() # 합계
40
>>> cal2.avg() # 평균
8.0
```

-생성자 구현

-필드 : 숫자 리스트