

Python 과제

2020년 6월 15일

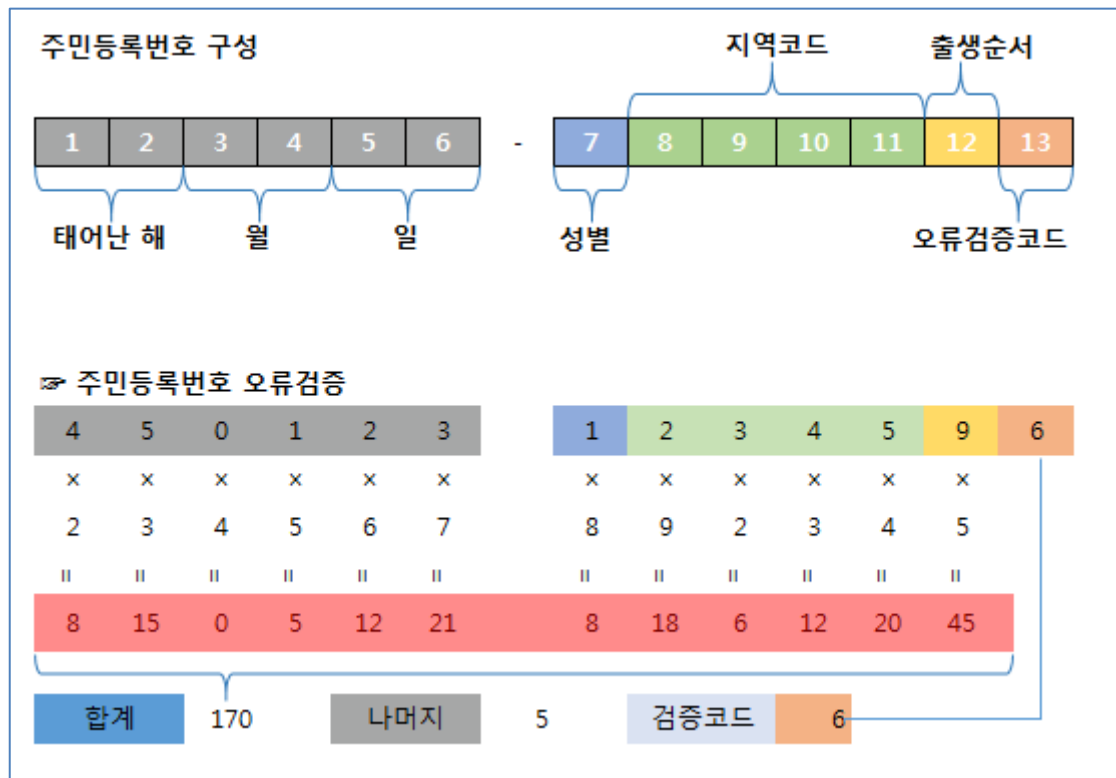
SWDO 3기 이름 : []

문제 1 :

주민번호를 입력 받아 검증하는 함수 validation()을 완성해서 제출하세요.

특별한 조건은 없지만 파이썬 함수와 연산자, 기능을 활용하여 “자바”처럼 코딩하지 말 것

- ✓ 제출 파일명 : juminVaridation_이름.py
- ✓ 주어진 코드를 이용해서 나머지 부분(pass 부분)을 완성 (나머지 부분은 고치지 말 것)
- ✓ 검증 내용
 - 길이가 - 포함 14자리 인지 체크
 - 반드시 생년월일뒤에는 “-”가 포함되어 입력해야 하며 다른 문자가 포함되었는지 체크
 - 검증코드 체크 : 검증 코드는 아래 계산식 이용
 - 문자열 slicing을 이용하여 생년, 생월, 생일을 얻어와야 함
 - 주민번호 유효성 검증 알고리즘



- 나머지 = 합계를 11로 나눈 나머지, 코드값 = 11-나머지

이렇게 계산된 코드값이 주민번호 마지막 오류검증 코드 자리와 같으면 주민번호는 올바른 것임.

✓

[실행 예]

입력(올바른 주민번호) : 111111-1111118

출력 : 당신은 11 년 11 월 11 일생 남자 입니다.

입력(잘못된 주민번호) : 111111*1111118

출력 : 잘못된 주민번호입니다

문제 2 :

2 차원 평면 상의 두 개의 좌표 값 (x_1, y_1) , (x_2, y_2) 를 입력하면 두 점 사이의 거리를 측정하는 함수를 작성하시오.

- ✓ 임의의 파일을 하나 만들어서 코드를 작성하여 제출
- ✓ math 모듈의 `sqrt()` 함수 등 관련 있는 함수 사용

```
def point(x1, y1, x2, y2) :  
    pass  
  
distance = point(10, -10, 0, 0) # x1 = 10, y1 = -10, x2 = 0, y2 = 0  
print(distance)
```

문제 3 : 사칙 연산이 가능한 SimpleCalc 클래스를 생성하시오

- ✓ 임의의 파일을 하나 만들어서 코드를 작성하여 제출

- 1) 멤버변수 : `first`, `second` 이 있음
- 2) 기본생성자 : 기본생성자에서는 멤버변수 `first`, `second` 를 단위 값 1로 초기화 한다.
- 3) 멤버메소드 : 데이터 초기화 하는 메서드와 연산자 `+` `-` `*` `/` `%` 다섯 개를 이용하여 두 수의 연산을 실행하는 메소드 완성

`setData()` - 첫 번째 수, 두 번째 수를 전달받아 멤버 초기화, `first`, `second`값 모두 0이 들어올 수 있음

`add()`-덧셈,

`sub()`-뺄셈,

`mul()`-곱셈,

`div()`-나눗셈: 나누는 수(`second` 값)가 0 일 경우 try ~ Exception 처리,

`mod()`-나머지: 나누는 수(`second` 값)가 0 일 경우 try ~ Exception 처리

위의 메소드는 전달인자 0 개 이상 받을 수 있으며, 만약 2 개 이상의 전달인자가 전달될 경우 에러 없이 앞 2 개의 수만 이용하여 연산 실시

만약 전달된 값이 0 개일 경우 `setData`에 의해 초기화 된 값 혹은 기본생성자에 의해 초기화 된 단위 값을 이용하여 연산 실시

* 중복된 코드가 많을 경우 다른 방법을 사용해도 됨

[실행 결과]

```
# 아래의 모든 테스트 코드가 실행되어야 함

a = SimpleCalc()
print(a.add())          # 2
print(a.add(10))        # 11.0
print(a.add(20, "a"))   # 정확한 데이터를 입력해주세요
print(a.add(10, 15))     # 25.0

b = SimpleCalc()
print(b.sub())           # 0
print(b.sub(10))         # 9.0
print(b.sub(20, "a"))    # 정확한 데이터를 입력해주세요
print(b.sub(10, 15, 4))  # -5.0

c = SimpleCalc()
print(c.mul())           # 1
print(c.mul(10))         # 10.0
print(c.mul(20, "a"))    # 정확한 데이터를 입력해주세요
print(c.mul(10, 15, 4))  # 150.0

d = SimpleCalc()
print(d.div())           # 1.0
print(d.div(10, 0))      # 정확한 데이터를 입력해주세요
print(d.div(20, "a"))    # 정확한 데이터를 입력해주세요
print(d.div(10, 15, 4))  # 0.6666666666666666

e = SimpleCalc()
print(e.mod())           # 0
print(e.mod(10, 0))      # 정확한 데이터를 입력해주세요
print(e.mod(20, "a"))    # 정확한 데이터를 입력해주세요
print(e.mod(10, 15, 4))  # 10.0
```