

## CSED232 Object-Oriented Programming (Spring 2019)

### Programming Assignment #5

#### Python

Due data: 2019년 5월 27일 월요일 23:59

담당조교: 정진웅 (jinwoong.jung@postech.ac.kr)

#### 목표

본 과제에서는 python 코딩을 학습하는 것을 목적으로 한다. 특히, python의 특징인 module, magic method들을 알아보고, class, operator overloading을 사용하는 것을 학습한다.

#### 개요

본 과제에서는 행렬과 문자열을 계산하는 가상의 계산기를 만든다. 이미 Assignment #4에서 Qt를 활용한 UI를 만드는 것을 해 보았기 때문에, 본 과제에서는 UI를 제외하고 가상계산기에 쓰이는 가장 중요한 data structure class를 구현한다.

#### 채점 방식

조교가 작성한 test.py 코드에서 각 class를 import하여 실행한다.

실행 환경은 Msys2, Python 3.7.3이다.

Msys에서의 실행 명령어는 `python test.py` 이다.

print 함수를 이용하여 출력된 것이 정답과 맞는지 확인한다.

실행이 안될 경우 기능 항목의 점수가 0점이 된다.

#### 제출 방식

제출 코드는 Assn5 폴더를 압축하여 학번.zip으로 만든다.

myScalar.py, myMatrix.py, myString.py의 3가지 파일이 꼭 포함되도록 제출할 것

제출 보고서는 학번.pdf로 만든다.

압축 파일과 보고서를 따로 제출한다.

## 1. 데이터 타입

### 기본 데이터 타입

python에서 기본으로 제공하는 float, str을 사용한다.

### 사용자 지정 데이터 타입

본 과제에서는 사용자 지정 데이터 타입으로 myScalar, myMatrix, myString의 class들을 구현하여야 한다. 각 데이터 타입에 대한 설명은 아래와 같다.

(각 데이터 타입의 연산에 대해서는 [2. 사용자 지정 데이터 타입 간 연산] 에서 설명한다.

### 연산

연산은 [(피연산자1) (연산자) (피연산자2)] 형식으로 2가지 피연산자를 가지며, 피연산자들의 타입 조합별로 연산자가 수행하는 연산이 각각 다르다.

### myScalar

myScalar은 1개의 float 변수를 val이라는 member variable에 저장한다.

### myMatrix

myMatrix은 행렬을 저장하는 타입이다.

row, col이라는 member variable에 행, 열의 수를 저장하고, val이라는 member variable에 list of list를 통해 row x col 개의 float 변수를 저장한다.

val은 col개의 변수를 가진 list를 row개 가진 list 변수이다. (c++에서의 2차원 배열과 유사)

### myString

myString은 1개의 str 변수를 val이라는 member variable에 저장한다.

## 2. 사용자 지정 데이터 타입 간 연산

연산은 총 4가지 ADD, SUB, MUL, DIV (+, -, \*, /)를 지원한다.

해당 연산을 구현하기 위해 class의 magic method 중 `__add__`, `__sub__`, `__mul__`, `__truediv__`를 overloading하여야 한다.

각 연산자로 나오는 결과는 아래 Table들을 통해 확인할 수 있다.

Table의 예시는 간단한 숫자들을 이용하였다. (실제로는 float형 변수)

<div>피연산자2</div> <div>피연산자1</div>	myScalar	myMatrix	myString
myScalar	<b>ADD(+):</b> 두 변수의 값을 더함 $3+5=7$	<b>ADD(+):</b> 변수의 값에서 행렬의 모든 원소 값을 더함 $2 + \begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 4 & 5 \\ 3 & 7 \end{bmatrix}$	<b>ADD(+):</b> 변수를 문자열로 바꾼 뒤 이어 붙임 $2.0 + 'bcd' = '2.0bcd'$ <b>str(val)을 사용</b>
	<b>SUB(-):</b> 두 변수의 값을 뺌 $3-5=-2$	<b>SUB(-):</b> 변수의 값에서 행렬의 모든 원소 값을 뺌 $2 - \begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & -3 \end{bmatrix}$	<b>SUB(-):</b> <b>불가능</b>
	<b>MUL(*):</b> 두 변수의 값을 곱함 $3*5=15$	<b>MUL(*):</b> 행렬의 모든 원소에 변수 값을 곱함 $2 * \begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 2 & 10 \end{bmatrix}$	<b>MUL(*):</b> <b>아래 참조</b>
	<b>DIV(/):</b> 두 변수의 값을 나눔 $3/5=0.6$	<b>DIV(/):</b> <b>불가능</b>	<b>DIV(/):</b> <b>불가능</b>

피연산자2 피연산자1	myScalar	myMatrix	myString
myMatrix	<b>ADD(+):</b> 행렬의 모든 원소에 변 수 값을 더함 $\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} + 2 = \begin{bmatrix} 4 & 5 \\ 3 & 7 \end{bmatrix}$	<b>ADD(+):</b> 2차원 행렬 덧셈 연산 $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 7 & 3 \end{bmatrix}$	<b>ADD(+):</b> <b>불가능</b>
	<b>SUB(-):</b> 행렬의 모든 원소에 변 수 값을 뺌 $\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} - 2 = \begin{bmatrix} 0 & 1 \\ -1 & 3 \end{bmatrix}$	<b>SUB(-):</b> 2차원 행렬 뺄셈 연산 $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} - \begin{bmatrix} -1 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 1 & 3 \end{bmatrix}$	<b>SUB(-):</b> <b>불가능</b>
	<b>MUL(*):</b> 행렬의 모든 원소에 변 수 값을 곱함 $\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} * 2 = \begin{bmatrix} 4 & 6 \\ 2 & 10 \end{bmatrix}$	<b>MUL(*):</b> 2차원 행렬 곱셈 연산 $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 7 & -4 \\ 1 & 0 \end{bmatrix}$	<b>MUL(*):</b> <b>불가능</b>
	<b>DIV(/):</b> 행렬의 모든 원소에 변 수 값을 나눔 $\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} / 2 = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$	<b>DIV(/):</b> <b>불가능</b>	<b>DIV(/):</b> <b>불가능</b>

피연산자2 피연산자1	myScalar	myMatrix	myString
myString	<b>ADD(+):</b> 변수를 문자열로 바꾼 뒤 이어 붙임 'hello'+4.5='hello4.5' <b>str(val)을 사용</b>	<b>ADD(+):</b> <b>불가능</b>	<b>ADD(+):</b> 두 문자열을 이어 붙임 'hello'+ 'world' ='helloworld'
	<b>SUB(-):</b> <b>불가능</b>	<b>SUB(-):</b> <b>불가능</b>	<b>SUB(-):</b> <b>불가능</b>
	<b>MUL(*):</b> <b>아래 참조</b>	<b>MUL(*):</b> <b>불가능</b>	<b>MUL(*):</b> <b>불가능</b>
	<b>DIV(/):</b> <b>불가능</b>	<b>DIV(/):</b> <b>불가능</b>	<b>DIV(/):</b> <b>불가능</b>

### **myScalar \* myString과 myString \* myScalar**

myScalar가 저장하고 변수의 정수 부분만큼 myString의 val을 반복한다.

float는 소수점 아래는 버림하여 정수를 만든다.

만약 정수가 음수일 경우에는 문자열이 반대로 뒤집혀서 반복된다.

myScalar와 myString의 곱셈에 대해서는 교환법칙이 성립한다고 보고 구현할 것

예시 (양수)	예시 (음수)
$3.75 * 'ab' = 'ababab'$	$-4.2 * 'abcd' = 'dcbadcbadcbadcba'$

### **myMatrix 간의 연산**

행렬 간 연산에 대해 dimension이 달라 연산이 불가능한 경우에는 `NameError('Dimension mismatch')`를 발생시켜야 한다.

(ex. 3x4 행렬과 2x4 행렬의 덧셈 및 뺄셈, 2x3 행렬과 2x3 행렬의 곱셈 등)

### 3. 실행 예시 및 구현 관련

Skeleton 코드는 각 class들의 initialization 및 출력에 관한 부분과, ADD 연산 중 몇가지를 구현하여 배포한다. 과제 문서와 함께 첨부한 skeleton 코드 중 test.py를 실행하면 아래와 같은 결과가 나오게 된다.

```
owner@DESKTOP-F7K6CUH MINGW64 ~/pyTest
$ python test.py
[[0.0, 0.0, 1.0], [0.0, 2.0, 0.0]]
[[3.0, 3.0, 4.0], [3.0, 5.0, 3.0]]

3.0abc
10.2

owner@DESKTOP-F7K6CUH MINGW64 ~/pyTest
$ |
```

이번 과제에서는 주어진 skeleton file중 myScalar.py, myMatrix.py, myString.py의 3가지 파일을 완성하여 제출해야 한다.

현재는 module 및 class의 circular import problem이 일어나지 않도록 작성한 skeleton 코드이며, 이를 잘못 건드릴 경우 실행이 되지 않을 수 있으니 주의해야 한다. 해당 문제에 대해서는 학생이 자유롭게 수정하여 .py 파일 등을 더 추가하고 import하는 것이 가능하다. 하지만 채점 시에는 test.py에서 main부분이 수정되어 채점 할 예정이다.

코드에는 주석으로, 과제 문서에는 1p에 명시 하였듯이 print함수를 이용하여 채점을 할 예정이므로 각 .py 파일에 대해 선언 부분인 \_\_init\_\_ 함수와 \_\_str\_\_ 함수 부분은 수정을 하지 않기를 바라며, 수정하였을 경우 기존에 작성되어 있는 코드와 동일한 결과가 나와야 한다.

또한, skeleton 코드에서도 볼 수 있듯이 정의 되지 않은 사용자 지정 데이터 타입 간 연산을 하려 한다면 NameError('Not defined')를 발생시켜야 한다. (myScalar.py의 line 30 참고)

python에서 기본적으로 제공하는 library만을 사용하여 과제를 수행하여야 한다.

(numpy같은 추가 라이브러리 사용을 금지한다.)

## 4. 채점 기준

### 기능 (50)

skeleton 코드의 test.py와 비슷하게 모든 가능한 연산을 실행해 보고, 제대로 실행되는 부분에 대한 점수로 채점한다. (Msys 환경, 부분 점수 있음)

Msys 환경에서 실행하다가 들여쓰기가 사라지는 등의 실수로 인해 프로그램이 제대로 실행되지 않을 경우 0점을 받을 수도 있으니 주의해야 한다.

\* 올바른 입력만이 주어진다.

### 설계 및 구현 (35)

함수 구조가 잘 나누어져 설계되어 있는지에 대해 평가한다.

예외 처리가 제대로 되어 있는지에 대해 평가한다. (skeleton code에 있는 Initialization Fail 및 Not defined와 행렬 간 연산에서 발생하는 Dimension mismatch에 대해서만 평가한다.)

### 코드의 가독성 (5)

프로그램의 중요한 함수나 분기 등에 적절한 주석이 작성되어 있는지를 평가한다.

(한글/영어 상관없음, 다만 모든 줄에 주석을 다는 것은 가독성을 해치는 것으로 본다.)

변수 이름, 함수 이름 등이 쉽게 뜻을 알 수 있도록 잘 명명되어 있는지에 대해 평가한다.

### 보고서 (10)

AssnReadMe.pdf를 기준으로 평가