

2019년-Spring

객체지향프로그래밍

Assignment #5

학번: 20180085

학과: 무은재학부

이름:송수민

POVIS ID: songsm921

명예서약(Honor code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

Assignment #5 : Calculator based on Python

I. 프로그램 개요

본 과제는 파이썬을 기반으로 하는 간단한 계산기를 만드는 과제이며, 이 과제를 수행하기 위해서는 파이썬의 기본문법, Operator Overloading 개념을 알고 있어야 한다. 이 계산기는 스칼라, 행렬, 문자열을 계산할 수 있다.

II. 프로그램 구조 및 알고리즘

이 프로그램에서는 어떠한 식을 프로그램에 주어질 때, 그 식의 피연산자의 나열 순서가 중요하다. 예를 들어, `myScalar + myMatrix` 이 순으로 주어진다면, `myScalar.py`에 있는 `__add__` magic method에 if문 중 `myMatrix`에 해당한다. 역으로, `myMatrix + myScalar` 이 식으로 주어진다면, `myMatrix.py`에 `__add__` magic method에 if문 중 `myScalar`에 해당한다. 이러한 과정이 수행되는 이유는 magic method의 argument가 `self`와 `op`로 구성 되어있기 때문이다.

아래는 각 해당하는 case의 구현 방법이다.

1. __add__

```
def addScalar(self,op):
    res = myMatrix(self.row,self.col)
    for i in range(len(self.val)):
        for j in range(len(self.val[i])):
            res.val[i][j] = self.val[i][j] + op.val
    return res
def addMatrix(self,op):
    res = myMatrix(self.row,self.col)
    for i in range(len(self.val)):
        for j in range(len(self.val[i])):
            res.val[i][j] = self.val[i][j] +op.val[i][j]
def addStr(self,op):
    res = myString.myString('')
    return myString.myString(str(self.val)+op.val)

if type(op) is myScalar.myScalar:
    return self.val+str(op.val)
elif type(op) is myString:
    return self.val+op.val
```

2. __sub__

```
def subScalar(self,op):
    res = myMatrix(self.row,self.col)
    for i in range(len(self.val)):
        for j in range(len(self.val[i])):
            res.val[i][j] = self.val[i][j] - op.val
    return res
def subMatrix(self,op):
    res = myMatrix(self.row,self.col)
```

```

        for i in range(len(self.val)):
            for j in range(len(self.val[i])):
                res.val[i][j] = self.val[i][j] - op.val[i][j]
def subMatrix(self,op):
    res = myMatrix.myMatrix(op.row,op.col)
    for i in range(len(op.val)):
        for j in range(len(op.val[i])):
            res.val[i][j] = self.val - op.val[i][j]
if type(op) is myScalar:
    return myScalar(self.val-op.val)

```

3. __mul__

```

def mulScalar(self,op):
    res = myMatrix(self.row,self.col)
    for i in range(len(self.val)):
        for j in range(len(self.val[i])):
            res.val[i][j] = self.val[i][j] *op.val
    return res
def mulMatrix(self,op):
    res = myMatrix(self.row,op.col)
    for i in range(res.row):
        for j in range(res.col):
            for k in range(self.col):
                res.val[i][j]+=self.val[i][k]*op.val[k][j]
    return res
if type(op) is myScalar:
    return myScalar(self.val*op.val)
def mulString(self,op):
    res = myString.myString('')
    if self.val>0:
        for i in range(int(self.val)):
            res.val+=op.val
    elif self.val<0:
        for i in range(int(self.val)*(-1)):
            temp = ''.join(reversed(op.val))
            res.val+=temp
    elif self.val==0:
        return
    return res

```

4. __truediv__

```

def divScalar(self,op):
    res = myMatrix(self.row,self.col)
    for i in range(len(self.val)):
        for j in range(len(self.val[i])):
            res.val[i][j] = self.val[i][j] /op.val
    return res
if type(op) is myScalar:
    return myScalar(self.val/op.val)

```

위 코드에서 if문을 사용하여, 해당하는 코드를 수행한다.

해당하는 case가 없는 경우는 try - catch문에 의해 Not defined를 에러 메시지로 보내고, 행렬 계산에서 더하기, 빼기, 곱하기 연산 중, 행렬 사이즈가 올바르게 맞지 않을 때는 Dimension mismatch를 에러 메시지로 보낸다.

III. 프로그램 실행 방법 및 실행 결과

```
7.2
[[5.0, 6.0, 7.0], [8.0, 9.0, 10.0]]
CSED

10.2
-0.75
-12.809999999999999
0.6
-0.6
[[3.0, 3.0, 4.0], [3.0, 5.0, 3.0]]
[[-2.0, -3.0, -4.0], [-5.0, -6.0, -7.0]]
[[3.0, 6.0], [9.0, 12.0]]
Not defined
None
3.75abc
Not defined
None
CSEDCSEDCSED
DESCDESCDESCDESCDESC
Not defined
None
[[7.2, 7.2, 8.2], [7.2, 9.2, 7.2]]
[[-3.75, -3.75, -2.75], [-3.75, -1.75, -3.75]]
[[-0.0, -0.0, -4.27], [-0.0, -8.54, -0.0]]
[[0.0, 0.0, 0.2], [0.0, 0.4, 0.0]]
[[-0.0, -0.0, -0.2], [-0.0, -0.4, -0.0]]
[[5.0, 6.0, 8.0], [8.0, 11.0, 10.0]]
Dimension mismatch
None
[[-5.0, -6.0, -6.0], [-8.0, -7.0, -10.0]]
Dimension mismatch
None
[[21.0, 24.0, 27.0], [47.0, 54.0, 61.0]]
Dimension mismatch
None
Not defined
None
Not defined
None
Not defined
None
Not defined
None
Not defined
None
HelloWorld
Not defined
None
abc7.2
Not defined
None
WorldWorldWorld
None
```

raise NameError를 print로 바꾸고 계속 진행되게 한 후 실행시킨 결과이다.

IV. 토론

Python을 사용하여 수행한 첫 과제이었다. Python은 이전에 배웠던 C++,C랑 매우 비슷하여 활용하는 부분에서는 어려운 점이 없었다. 다만, 조금 불편했던 점이 있었는데, 이전에 배웠던 언어는 중괄호를 사용하여 구분할 것을 구분하였는데, 이것은 스페이스바 4자리, 또는 탭 한자리로 구분하게 되는데, 이것이 혼용될 경우 컴파일이 되지않아 꽤 불편하였다.

V. 결론

Python을 사용하여 magic method, module 사용법, try – raise 예외 처리에 대해 배웠다. Python이라는 언어는 언어를 잘 모르는 사람이 보아도 비교적 쉽게 이해 할 수 있는 언어라고 생각했다.

VI. 개선사항

이번 과제에서 개선사항을 뽑자면 처음부터 Indentation을 스페이스바 4자리나, Tab 1자리로 통일을 하고 시작하여야 한다는 것이다. 이를 처음에 정하지 않아, 나중에는 error가 발생하여 스페이스바와 탭이 섞여 있는 곳을 찾는 것 및 통일과정이 매우 번거로웠다.