

ソフトウェア演習Ⅲ〔課題 5: 演算子オーバーローディング・デコレータ〕 青野雅樹

行列の演算に関して、Python 言語で演算子のオーバーローディングを加えた以下の条件をみたすプログラム `kadai5.py`, 及び(2 パターンの)実行結果 (を結合した) `kadai5.txt` を ZIP にまとめ Moodle LMS にアップロードせよ。メ切は 11 月 15 日 (火) の夜までとする。

[1] M 行 N 列の行列を表す `Matrix` クラスを独自に作成せよ。`Matrix` クラスには、`row`, `col`, `mat` のメンバー変数をもたせるものとする。ここで `row` は行を表す整数、`col` は列を表す整数、`mat` は M 行 N 列の実数行列を NumPy 配列で保持するものとする。以下の(ア)～(ウ)に述べる 4 つの演算子を `Matrix` クラス内に演算子のオーバーローディングとして定義し、さらに、`__main__` から実行する際は、[2]の指示に従うこと。

(ア) 同一サイズの行列同士の加算(`__add__`)と減算(`__sub__`)を行い、結果の行列を返す+ 演算子ならびに- 演算子。(サイズが一致しない場合は、警告を出し、演算子の左辺の行列だけ返すこと)

(イ) 同一サイズの行列同士の乗算(`__mul__`) (通常の行列乗算ではなく、対応する要素ごとの積とする) を計算し、結果の行列を返す* 演算子。

(ウ) 同一サイズの行列同士の除算(`__truediv__`) (要素ごとの除算) を行う / 演算子。ただし、 A / B において、 B の要素に 1 つでも 0 があれば、警告を出し終了するような関数デコレータ (`@divzero_warning`) を (`Matrix` クラスの外に) 作成すること。

[2] `__main__` 部分では行列クラスのオブジェクトを作成し、以下の計算を行え。

(ア) 3 つの整数 $M, N, flag$ をコマンドラインで指定 ($3 \leq M, N \leq 5, flag$ は 0 または 1) し、 M 行 N 列の行列 A と同じ行数と列数を有する行列 B を生成し、 M, N および行列中身をプリントせよ。`flag` は 1 または 0 とする。プリントの際、行列の各要素は、小数点以下は 4 桁表示とする。このような行列を少なくとも 2 パターン (1 パターン目 (`flag==0`) は、 A も B も要素は $[-10, 10]$ の間の実数でランダムに発生させるパターン、2 パターン目 (`flag==1`) は、 A と B をランダムに $[-10, 10]$ の実数で生成したあと、 $B.mat[0, 0]=0.0$ とする。

(イ) [1] の (ア) で作成した加算、減算、乗算演算子で $C=A+B$, $C=A-B$, $C=A*B$ をそれぞれ実行し、結果の行列 C をプリントせよ。

(ウ) [1] の (ウ) で作成した除算演算子で $C=A/B$ を実行し、正しく要素ごとに除算ができる場合 ((ア) で述べた 1 パターン目) と、`@divzero_warning` が出る場合 ((ア) で述べた 2 パターン目) で、結果の行列 C のプリント (1 パターン目)、もしくは、警告終了させること (2 パターン目)。なお、デコレータ関数である `divzero_warning` は、ゼロ割り算のときは、`None` を返し、 C が `None` でないときのみ、結果の行列 C をプリントさせること。

【コメントとヒント】

デコレータ関数の `divzero_warning` 関数は `def divzero_warning(func)` のようなプロトタイプで、内部で定義する関数内関数は、`return func(A,B)` のような `return` 文を持つこととなります。ここで、`A,B` は、本課題で定義する `Matrix` クラスの変数です。2つの数値の割り算での警告を出す、デコレータ関数例は、たとえば

<https://www.educative.io/edpresso/how-to-use-decorators-in-python>

<https://prabhupant.medium.com/python-decorators-tutorial-e577888ba1ac>

あたりが参考になります。本課題では計算対象は数値でなく、行列ですのでご注意ください。また、今回は2パターン目で `B.mat[0,0]=0.0` としましたが、`B` のどの要素がゼロであっても同様のふるまいになるように実装してください。

実行結果例は以下のようです。

```
$ python kadai5.py 3 3 0
```

```
*****
```

青野雅樹, 01162069

日付: 2022-10-31 17:59:59.499032

内容: 行列演算 (演算子のオーバーローディング)

およびデコレータ関数利用例

```
*****
```

入力された値 M=3,N=3,flag=0

A:

```
| 1.3867 3.6909 3.6438 |  
| -4.4542 8.7177 -3.0589 |  
| -7.4401 -8.0816 -6.5842 |
```

B:

```
| 6.3598 -7.6003 -8.2997 |  
| 2.5985 -3.9202 -9.8748 |  
| -3.7552 7.8640 6.7632 |
```

C=A+B:

```
| 7.7465 -3.9095 -4.6559 |  
| -1.8556 4.7975 -12.9337 |  
| -11.1953 -0.2175 0.1790 |
```

C=A-B:

```
| -4.9730 11.2912 11.9435 |  
| -7.0527 12.6379 6.8160 |  
| -3.6849 -15.9456 -13.3474 |
```

C=A*B:

```
| 8.8192 -28.0519 -30.2423 |  
| -11.5742 -34.1751 30.2057 |  
| 27.9390 -63.5537 -44.5305 |
```

C=A/B:

```
| 0.2180 -0.4856 -0.4390 |  
| -1.7141 -2.2238 0.3098 |  
| 1.9813 -1.0277 -0.9735 |
```

\$ python kadai5.py 3 3 1

青野雅樹, 01162069

日付: 2022-10-31 18:00:01.349035

内容:行列演算(演算子のオーバーローディング)

およびデコレータ関数利用例

入力された値 M=3,N=3,flag=1

A:

```
| 1.2190 -7.1695 6.6765 |  
| -6.0564 7.6849 4.8047 |  
| -2.6855 -8.2021 4.1663 |
```

B:

```
| 0.0000 -1.5065 8.9342 |  
| 8.3449 -5.8674 9.8271 |  
| 0.1548 2.9059 -9.0222 |
```

C=A+B:

```
| 1.2190 -8.6759 15.6107 |  
| 2.2885 1.8174 14.6319 |  
| -2.5307 -5.2962 -4.8559 |
```

C=A-B:

```
| 1.2190 -5.6630 -2.2576 |  
| -14.4012 13.5523 -5.0224 |  
| -2.8403 -11.1079 13.1885 |
```

C=A*B:

```
| 0.0000 10.8006 59.6495 |  
| -50.5397 -45.0905 47.2167 |  
| -0.4157 -23.8341 -37.5895 |
```

C=A/B:

ゼロ割り算が発生しました