

ソフトウェア演習ⅡB〔課題2：クラスと多次元配列〕担当：青野雅樹

以下の内容を満たす Java プログラム Matrix.java、2つの入力ファイル、及び 実行結果を ZIP でまとめて Moodle にアップせよ。締切は 7 月 6 日の夜までとする。

$M \times N$ 行列を表す **Matrix** クラスを作成せよ。条件として、以下の 4 つを含むこと。

- (1) $M \times N$ の 2 次元配列で行列(名前 m)を private な double 型のメンバー変数として保持させ、行サイズ(名前 row)と列サイズ(名前 col)を private な int 型のメンバー変数としてクラスに保持する。行列の行と列を返すメンバー関数を public int 型で、それぞれ public int getRow(), public int getColumn()関数で用意せよ。
- (2) **Matrix** クラスのコンストラクタとして、無引数のもの (public Matrix()) と、(1)で述べた行と列を引数で与える 2 引数のもの(public Matrix(int M, int N))の 2 つのコンストラクタを作成せよ。
- (3) ある **Matrix** オブジェクト(サイズは $M \times K$ で行列 A と呼ぶ)と別の **Matrix** のオブジェクト (サイズは $K \times N$ で行列 B と呼ぶ)を与え、行列の乗算 ($A \cdot B$) を実行し、結果を別の **Matrix** クラスのオブジェクトとして返す multiply 関数 (プロトタイプは、**Matrix multiply(Matrix B)**) を用意する。 K が一致していない場合は、警告を出してプログラムを終了させること。 ($C = A \cdot B$ で行列の積を書くとき、 $C = A.multiply(B)$ のように呼び出すとする) **【注】multiply(Matrix A, Matrix B)としない**
- (4) **Matrix** クラスが保持する $M \times N$ 行列の中身の要素をプリントする void 型の print() 関数を用意する。
- (5) M 行 N 列の実数データ (スペース区切り) を記述したデータ用ファイルから読み込み、**Matrix** クラスのオブジェクトを生成する read(String fileName)関数。プロトタイプは、public Matrix read(String filename)とすること。なお、read 関数の中で例外処理を行うこと。ファイルが存在しない場合(FileNotFoundException の場合)は、その旨をプリントし、System.exit(1)で終了すること。整数や実数値を読もうとして、データの型が違う場合、すなわちフォーマットエラーの場合 (NumberFormatException の場合) も、その旨をプリントし、System.exit(1)で終了すること。
 - データファイルは先頭行に M N のように、行のサイズと列のサイズがあり、第 2 行目から m[0][0] m[0][1] ... m[0][$N-1$]のようにデータを記述し、最後の行は、m[$M-1$][0] m[$M-1$][1] ... m[$M-1$][$N-1$]のデータが並ぶものとする。

前準備：

2つの行列データファイル $M \times K$ 行列 A と $K \times N$ 行列 B を作成せよ。以降、 $A=(M,K)$, $B=(K,N)$ と書くことがある。ファイル名はそれぞれ $A.txt$, $B.txt$ とする。行列の中身は適当でよいが、プログラム 2-11 で紹介した乱数を用いて、 $[-1, +1]$ の間の実数乱数を、行列の要素生成に利用することを推奨する。なお、この前準備で、プログラミングを行った場合は、提出物には含まない。行列データだけで提出物に含めること。

上記の条件を変数や関数として含む **Matrix** クラスを定義し、**main** で以下の処理を行え。

- (A) **main** 関数から無引数の **Matrix** のコンストラクタを呼び出し、**Matrix** オブジェクト（ここでは x と呼称する）を生成せよ。
- (B) x に対して、前準備で述べた2つの行列データファイルをコマンドラインから2引数を ($args[0]$ と $args[1]$ とする) 与え、それぞれ、**Matrix** $A = x.read(args[0])$, **Matrix** $B = x.read(args[1])$ とし、2つのファイルを読み込み、個々の行列の中身をプリントせよ。ただし、引数の数が2でないとき、すなわち、**public static void main(String[] args)**において、 $args.length \neq 2$ のときは、「行列データファイルは2つ与えてください」と警告し、**System.exit(1)**で終了せよ。
- (C) 次に **Matrix** A と **Matrix** B の乗算を実行せよ。その前に、(1)で準備した **getRow()**, **getColumn()**関数で、 $A = (M,K)$, $B = (K,N)$ が得られたとする。**実行に当たって、 $3 \leq M, N, K \leq 10$ とせよ。更に追加条件として、 M, N, K の値のどれか一つは他と異なる値とすること**（つまり、すべて同じサイズの正方行列としないこと）。また、 A の列サイズと B の行サイズ(ここでは K としている)が一致しない場合は、一致しない旨をプリントし **System.exit(1)**で終了せよ。 M, N, K の設定が上記の不等号や追加条件を満たさない場合、その旨をプリントし **System.exit(1)**で終了すること。

コメントとヒント：

行列の積は、たとえば、 $C = AB \equiv (c_{i,j}) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{K-1} a_{i,k} b_{k,j}$ のように3重ループで計算して

結構です。データの読み込みの一部は、資料で紹介した **String** クラスの **split** 関数を使うといいかと思います。

以下にファイル例と実行結果のサンプルを載せます。（これはデバッグに使ってください。提出は各自、上の枠内の注意書きを読み、自作の行列ファイルで実行してください）

入力ファイルの例： $A * B$

A.txt:

```
3 3
0.408248 0.707107 1.0
0.816497 0.0 0.0
0.408248 -0.707107 1.0
```

B.txt

```
3 4
1.732050 0.0 1.0 0.0
0.0 1.0 1.0 0.0
1.0 0.0 0.0 1.0
```

実行結果例：(ファイル名は、java のコマンドラインから与えています！プログラムに埋め込まないでください)。この例では、Matrix.java, A.txt, B.txt, 及び出力結果を保持する Kadai2.txt を ZIP して Moodle LMS にアップロードしてください。提出したプログラムをチェックする都合上、必ず \$ java(u) Matrix A.txt(ファイル1) B.txt (ファイル2) で確認できるようにしてください。以下は、実行例で出力を Kadai2.txt としています。出力の切り替えができない場合、出力結果をコピー&ペーストして、Kadai2.txt ファイルを作成ください。

なお、チェック時には、上に述べた警告を出す機能があるかどうかのチェックを行います。チェック用のデータは、<https://www.kde.cs.tut.ac.jp/~aono/2021/matrix-good-bad-data/> においてありますので、個々の警告が正しく出るか、適宜これらのデータを使い、試しておいてください。

\$ javau Matrix A.txt B.txt > Kadai2.txt

```
*****
課題2：青野雅樹：01162069
日付：2021年6月16日 10:30:10 JST
入力ファイル名 1：A.txt
入力ファイル名 2：B.txt
ファイルから行列を読み込み行列クラスの積計算
*****

入力行列 A, サイズ: (3, 3)
| 0.408248  0.707107  1.000000  |
| 0.816497  0.000000  0.000000  |
| 0.408248  -0.707107  1.000000  |

入力行列 B, サイズ: (3, 4)
| 1.732050  0.000000  1.000000  0.000000  |
| 0.000000  1.000000  1.000000  0.000000  |
| 1.000000  0.000000  0.000000  1.000000  |

出力の積行列 C=A x B, サイズ: (3, 4)
| 1.707106  0.707107  1.115355  1.000000  |
| 1.414214  0.000000  0.816497  0.000000  |
| 1.707106  -0.707107  -0.298859  1.000000  |
```