

ГУАП
КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н.		Линский Е. М.
должность , уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЫ № 3 СОЗДАНИЕ ПРОГРАММЫ НА ЯЗЫКЕ JAVA

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5022		С.А.Баландюк
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2022

Задание

Напишите класс `Matrix`, реализующий матрицы и расширяющий его класс `SquareMatrix`, реализующий квадратные матрицы. В классах должны быть определены:

- конструкторы с параметрами размерами матриц, создающие нулевую матрицу для `Matrix` и единичную для `SquareMatrix`;
- методы **`Matrix`** `sum(Matrix)` и **`Matrix`** `product(Matrix)`, вычисляющие сумму и произведение матриц; метод `sum` должен быть переопределен в `SquareMatrix`;
- методы `setElement(int row, int column, int value)` и `getElement(int row, int column)`, для обращения к элементам матрицы;
- метод `toString()`.

Напишите собственный класс исключения, расширяющий (*наследующий*) класс `RuntimeException`. Во всех конструкторах и методах должны бросаться исключения в тех случаях, когда соответствующая операция невозможна (например, при сложении матриц разных размеров). Исключения должны содержать информацию о том, какая именно проблема возникла. Достаточно хранить эту информацию в виде строки, возвращаемой методом `getMessage()`.

Примените к написанной программе:

1. Разложите классы по пакетам.
2. Напишите слово **`final`** в тех случаях, где оно разумно.
3. Реализуйте для матриц метод `equals()`.

Дополнительное задание

Создать класс `PartialMatrix`(наследник `Matrix`). Матрица такого класса состоит из равных кусочков (размер кусочков `rows x 2`), т.е. элементы в первом и втором столбце равны элементам в третьем и четвертом столбце соответственно. Хранится только один кусочек матрицы. Этот класс должен работать со всем методами класс `Matrix`.

Инструкция

Данная программа позволяет пользователю работать с обычными и квадратными матрицами(наследник `Matrix`)

Список доступных методов:

`Matrix` `product(Matrix)` – умножение матриц

`Matrix` `sum(Matrix)` – сумма матриц

`setElement(int row, int column, int value)` – установить элемент в определенную позицию

`getElement(int row, int column)` – получить элемент на определенной позиции

`inputMatrix()` – метод для пользовательского ввода матриц

метод `toString()`

метод `equals()`

SquareMatrix() – конструктор, создающий единичную квадратную матрицу

Тестирование

Init matrix:

1 2 3

4 5 6

7 8 9

Init SquareMatrix:

1 0 0

0 1 0

0 0 1

1. Тест 1 (сложение обычной матрицы с квадратной)

Result Matrix:

2 2 3

4 6 6

7 8 10

2. Тест 2(Умножение ResultMatrix саму на себя)

Result after product:

33 40 48

74 92 108

116 142 169

3. Тест 3(setElement(1, 2, -10))

Result after setElement:

33 40 48

74 92 -10

116 142 169

Тестирование дополнительного задания

Init PartialMatrix(4, 4):

1 2 1 2

3 4 3 4

5 6 5 6

7 8 7 8

1. Тест 1 (Сложение матрицы с самой собой)

Result Matrix:

```
2  4  2  4
6  8  6  8
10 12 10 12
14 16 14 16
```

2. Тест 2(Умножение изначальной матрицы саму на себя)

Result Matrix:

```
26  32  26  32
58  72  58  72
90  112  90  112
122 152 122 152
```

3. Тест 3(Установить элементы в изначальную матрицу)

```
setElement(0, 4, 22);
setElement(2, 3, 80);
setElement(1, 2, -10);
```

Result Matrix:

```
22  2  22  2
-10 4  -10  4
5   80  5   80
7   8   7   8
```