

ГУАП  
КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

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

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЫ №5 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 интерфейс `IMatrix` с несколькими реализациями --- `UsualMatrix` и расширяющий его `SquareMatrix` из предыдущих заданий и `SparseMatrix` для разреженных матриц. `SparseMatrix` должен быть реализован с помощью `LinkedList` (возможно, вам потребуется создать какие-то еще дополнительные классы, которые должны быть вложенными/внутренними). Все общие методы должны быть представлены в интерфейсе `IMatrix`.

Напишите программу, создающую 2 случайные матрицы размером 1000x1000 с 1000 ненулевых элементов в каждой двумя способами --- с помощью обычных и разреженных матриц. Проверьте, что сложение и умножение для разных видов матриц дает одинаковые результаты.

## Дополнительное задание

Добавить интерфейс `IAngle` и два класса которые его реализуют.

Первый класс `Radian` – класс для работы с углами в радианах.

Второй класс `Degree` – класс для работы с углами в градусах.

Эти два класса реализуют метод интерфейса `public IAngle sum(IAngle angle)`, который позволяет складывать значения двух углов.

## Инструкция

Данная программа позволяет пользователю работать с обычными, квадратными матрицами (наследник `Matrix`) и разреженными матрицами. Список доступных методов:

**`IMatrix product(IMatrix)`** – умножение матриц

**`IMatrix sum(IMatrix)`** – 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()`

## Тестирование

1. Тестирующая программа по заданию

Создаем 4 матрицы (матрицы заполняются случайным образом, с помощью класса `Random`) таким образом, что:

`SparseMatrix sm1 == UsualMatrix us1`

`SparseMatrix sm2 == UsualMatrix us2`

Размеры всех матриц 1000x1000

Проверим, что умножение для разных видов матриц дает одинаковый результат с помощью метода equals()

Выглядеть это будет так:

```
us1.product(us2).equals(sm1.product(sm2))
```

Результат: true

## Тестирование дополнительного задания

Init angles:

IAngle degreeAngle = new Degree(120) – создаем угол в градусах равный 120

IAngle radianAngle = new Radian(1.25) – создаем угол в радианах равный 1.25

### 1. Тест 1

Сложение угла в градусах с углом в радианах, возвращаем угол в градусах

degreeAngle.sum(radianAngle) = 191.62 градусов

### 2. Тест 2

Сложение угла в радианах с углом в градусах, возвращаем угол в радианах

radianAngle.sum(degreeAngle) = 3.34 радиан

### 3. Тест 3

Сложение угла в градусах с углом в градусах, возвращаем угол в градусах

degreeAngle.sum(degreeAngle) = 240 градусов