

ГУАП
КАФЕДРА № 52

ПРЕПОДАВАТЕЛЬ

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 СОЗДАНИЕ ПРОГРАММЫ НА ЯЗЫКЕ JAVA

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5022		К.А. Агаширинов
		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

Сделать реализацию `SparseMatrix` на `Array<LinkedList<Element>>`, каждая строка храниться списком `"class Element { int j; int val }"`. Все списки записаны в массив, ускоряется доступ. Сравнить в `main` две реализации с помощью `System.currentTimeMillis()` (замерить время до вызова умножения, потом после вызова умножения, вычесть).

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

метод `toString()`

метод `equals()`

Тестирование

1. Основное задание:

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
спомощью метода equals()

Выглядеть это будет так:

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

Результат: true

2. Дополнительное задание:

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

```
SparseMatrix sm == UsualMatrix us1
```

```
SparseArrayMatrix sam == UsualMatrix us2
```

Размеры всех матриц 1000x1000

После создания SparseMatrix sm, UsualMatrix us1, замеряем время с помощью System.currentTimeMillis(), после создаем SparseArrayMatrix sam, UsualMatrix us2, вызываем метод **IMatrix** product(**IMatrix**) и также замеряем время с помощью System.currentTimeMillis() и вычитаем.

Тест 1:

before multiply: 1651097622187

after multiply: 1651097622367

result: 180

Тест 2:

before multiply: 1651097685415

after multiply: 1651097685599

result: 184

Тест 3:

before multiply: 1651097711190

after multiply: 1651097711376

result: 186