

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Лабораторная работа №1  
по дисциплине  
“Линейная алгебра и анализ данных”

Семестр I

Выполнили:  
студенты

Тиганов Вадим Игоревич  
гр. J3112  
ИСУ 467701

Вагин Арсений Антонович  
гр. J3112  
ИСУ 465339

Отчет сдан:  
XX.12.2024

Санкт-Петербург  
2024

## Цель лабораторной работы

Освоить основные концепции линейной алгебры и анализа данных по работе с матрицами. Научиться реализовывать и тестировать алгоритмы работы с матрицами в разреженно-строчном формате. Изучить и понять принципы работы алгоритмов, а также .tex верстания для создания отчета.

## Задачи лабораторной работы

1. Реализация хранения матриц в разреженно-строчном виде.
2. Реализация операций над матрицами.
3. Вычисление определителя и проверка существования обратной матрицы.
4. Тестирование и проверка правильности работы алгоритмов.
5. Верстка отчета в формате L<sup>A</sup>T<sub>E</sub>X

# Ход выполнения лабораторной работы

## Задача 1

Задача заключалась в реализации следующих функций в классе: (был выбран ЯП Python, полный листинг кода см. в приложении А)

- Ввод матрицы заданного размера пользователем.
- Подсчет следа матрицы.
- Поиск и вывод элемента матрицы по заданным индексам.
- Тестирование работы программы и создание консольного пользовательского интерфейса.

### Использованные библиотеки и инструменты языка

В ходе написания программы были использованы только стандартные средства языка. Также для удобства и лучшей читаемости кода была импортирована библиотека `typing`.

### Реализация функций и основные идеи

Все функции были реализованы в классе `MatrixKeeper`

Написаны функции:

`inputMatrix` - ввод матрицы пользователем,

`trace` - поиск следа матрицы,

`findByIndex` - поиск элемента по введенному индексу.

Суть работы алгоритмов:

- `inputMatrix`: Приглашение пользователя ко вводу. Вначале через пробел вводятся два числа типа `int` - размер матрицы. Вторым приглашением вводится матрица по строке, элементы в строке разделяются пробелом.
- `trace`: След матрицы - сумма элементов главной диагонали этой матрицы. Циклом, оставаясь в пределах матрицы, проходимся по элементам с индексами вида `[i][i]`, считаем сумму таких элементов. Можем так делать по той причине, что матрица имеет следующую структуру в классе:

```
1 self.matrix: Optional[List[List[float]]]
```

- то есть храним матрицу как список, каждый элемент которого является тоже списком.

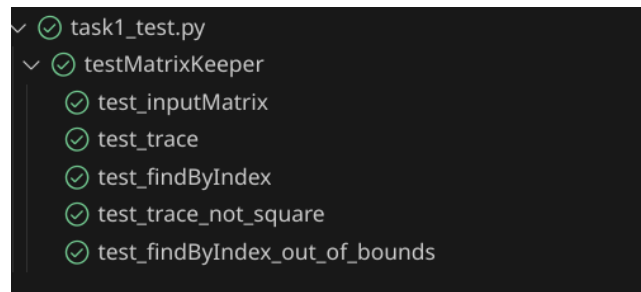
- `findByIndex`: возвращаем элемент из матрицы, отнимая от индексов по единице, т.к. в ЯП отсчет начинается с нуля.

```
1 return self.matrix[n-1][m-1]
```

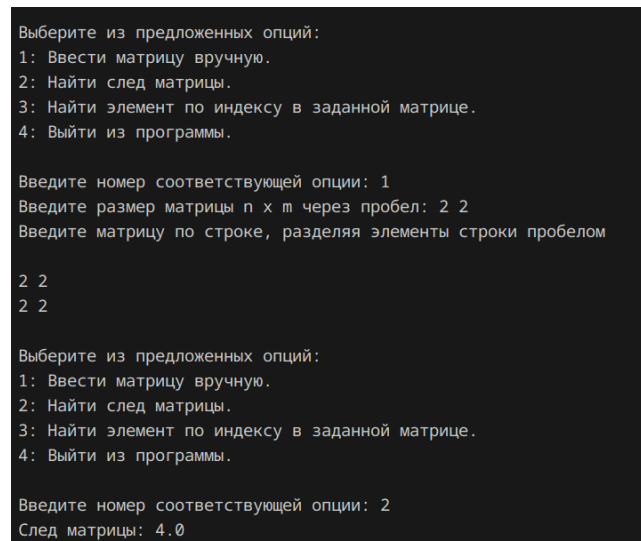
## Тестирование программы

Написаны юниттесты для каждой функции класса с помощью стандартной библиотеки `unittest`. Также протестировано вручную.

Листинг кода теста см. в приложении A-test.



## Успешное прохождение unittests



Ручной тест поиска следа матрицы 2x2 со всеми элементами, равными 2.

```
Введите номер соответствующей опции: 1
Введите размер матрицы n x m через пробел: 9 5
Введите матрицу по строке, разделяя элементы строки пробелом

1 2 3 5 1
1 6 9 2 4
3 5 1 2 5
1 2 5 7 4
-10000 1 6 32 1
0.0003 1 2 6 3
4 2 8 3 7
5 4 2 3 6
5 3 1 2 7

Выберите из предложенных опций:
1: Ввести матрицу вручную.
2: Найти след матрицы.
3: Найти элемент по индексу в заданной матрице.
4: Выйти из программы.

Введите номер соответствующей опции: 3
Введите номер строки: 5
Введите номер столбца: 1
Элемент матрицы [5][1] = -10000.0
```

Ручной тест поиска страшного элемента в страшной матрице 9x5.

Итак, справились с первым заданием.

## Задача 2

## Вывод

## Приложения

## Список использованной литературы