Пермский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

*Факультет социально-экономических и компьютерных наук*

Мальшаков Александр Аркадьевич

**ЛАБОРАТОРНАЯ РАБОТА №5**

*Функции, многомерные и рваные массивы, строки*

студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки 38.03.05-Бизнес - информатика

Руководитель

Старший преподаватель кафедры ИТБ \_\_\_\_\_\_\_\_\_\_\_\_

Волков С. А.

Пермь, 2024 год

**Оглавление**

[Постановка задачи 3](#_Toc185329683)

[Общая 3](#_Toc185329684)

[Вариант №9 3](#_Toc185329685)

[Анализ 4](#_Toc185329686)

[Матрица 4](#_Toc185329687)

[Рваный массив 4](#_Toc185329688)

[Строка 4](#_Toc185329689)

[Проектирование 5](#_Toc185329690)

[Матрица 5](#_Toc185329691)

[Рваный массив 5](#_Toc185329692)

[Строка 6](#_Toc185329693)

[Тестирование 7](#_Toc185329694)

[Матрица 7](#_Toc185329695)

[Рваный массив 8](#_Toc185329696)

[Строка 9](#_Toc185329697)

[Приложение А 10](#_Toc185329698)

# Постановка задачи

## Общая

В общем, задание состоит из 3 подзадач, работа с матрицей, работа с рваным массивом и работа со строкой.

1. Сформировать динамический двумерный массив, заполнить его.
2. Выполнить указанное в варианте задание.
3. Сформировать динамический рваный массив, заполнить его.
4. Выполнить указанное в варианте задание.
5. Ввести строку символов. Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит один знак препинания (.!?).
6. Выполнить обработку строки в соответствии с вариантом, используя по возможности, методы класса String.
7. Все результаты нужно выводить на экран.

## Вариант №9

Задачи для конкретного варианта:

1. Сформировать и заполнить динамический двумерный массив.
2. Добавить K строк в начало матрицы
3. Сформировать и заполнить динамический рваный массив.
4. Удалить строку с номером N
5. Ввести строку символов. Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит один знак препинания (.!?).
6. Перевернуть каждое слово, длина которого совпадает с его номером в предложении.
7. Все результаты нужно выводить на экран.

# Анализ

Контроль ввода чисел в матрице и рванном массив представлен ниже (см. Таблица 1). Для длин массивов таблицы таже, но числа не могут быть отрицательными.

Таблица 1 – Контроль ввода чисел

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Условие** | **Выходные данные** |
| Любые символы, не являющиеся целым числом | Невозможно вычислить | Сообщение об ошибке о том, что введенная строка не является целым числом |
| Целое число | Программа продолжается | Стандартные |

## Матрица

Для формирования матрицы нужно считать два числа – количество строк и количество столбцов, и затем просто заполнить ее построчно.

Чтобы добавить K строк в начало нужно создать новую матрицу для строк которые нужно вставить, потом создать матрицу в которую уже надо заполнить элементами исходной матрицы и матрицы из K строк, и присвоить эту матрицу текущей.

## Рваный массив

Для формирования рваного массива нужно создать массив и заполнить его другими массивами, для длины массивов можно при создании каждого массива вводить его длину, или можно использовать случайные числа.

Чтобы удалить строку нужно создать новый массив на один элемент меньше и скопировать в него все массивы, игнорируя тот. который удаляем.

## Строка

Для выполнения задания со строкой нужно один раз пройтись по строке, отделяя слова друг от друга, а также сохраняя знаки препинания, после вывести переворачивая те слова чья длина равна позиции в предложении.

# Проектирование

Весь код предоставлен в приложении (см. Приложение А), там находиться ссылка на github, код для этой лабораторной работы находиться в папке lab#5.

## Матрица

Для матриц можно создать класс MyMatrix в который будет хранить саму матрицу и методы добавления строк, заполнения и вывода на экран, в конструктор класса надо будет передать количество столбцов и строчек.

Метод для добавления строк будет перегружен чтобы принимать этот же класс или матрицу целых чисел. Для объединения этих матриц будет создаваться временная переменная result для хранения объединенной матрицы, в конце функции в текущую ссылку на массив присваиваем result. Чтобы объединить матрицы надо записать их по очереди в result с помощью двух циклов двойной вложенности. Первый для добавляемой матрицы, второй для исходной.

Метод для заполнения матрицы будет просто в каждый элемент матрицы присваивать либо случайное значение, либо ввод пользователя.

Метод для вывода на экран просто будет выводить все элементы, делая новую строку в конце цикла, отвечающего за строки.

## Рваный массив

Для рваных массивов можно создать класс массива MyCollection, который сможет хранить любые другие типы, это можно сделать с помощью обобщений. В нем должен быть массив для типа хранимых элементов и функция удаления элемента по индексу, в конструктор будет передаваться только длина массива.

Для того чтобы хранить числа в нем и заполнять, заполнять и выводить нужно наследоваться от MyCollection<int> и создать класс MyIntArray для массива из целых чисел, и уже в нем определить ввод и вывод чисел в рваный массив.

В методе удаления элемента нужно создать новый массив и сложить туда все элементы от 0 индекса до удаляемого и от удаляемого + 1 до конца массива, а поле присвоить получившийся массив в текущий.

## Строка

Для строк можно создать свой класс MyString, который будет наследоваться от MyCollection<char>, и в котором будет класс Sentence и метод возвращающий массив типа Sentence.

Класс Sentence будет хранить 2 массива – массив слов и массив знаков препинания, а также в нем будет метод, выводящий предложение на экран, в котором будет выполняться задание, слова, у которых длина равна позиции, будут перевернуты. Конструктор Sentence будет принимать 2 массива, слов и знаков препинания для этих слов.

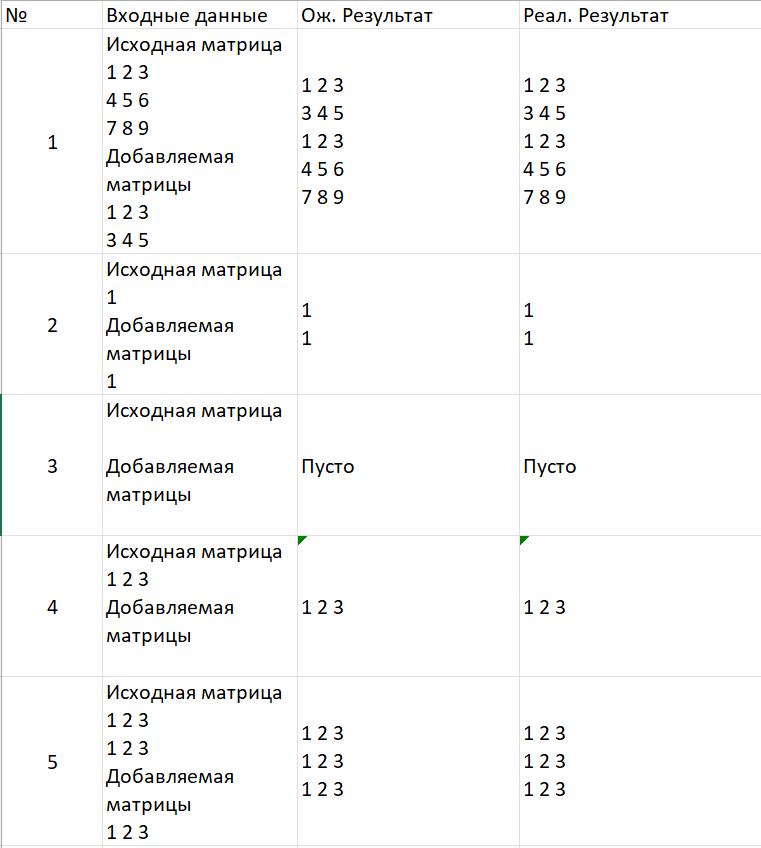
В методе, возвращающем массив предложений, нужно лишь объявить этот массив и пройтись по строке, разделяя её по предложениям, словам и знакам препинания, для этого нужно находить символы знаков препинания.

# Тестирование

Тестировать правильность ввода длин массивов и чисел внутри них не нужно, так как для этого используются те же функции и конструкции, как и в прошлой работе, а именно класс InputManager.

## Матрица

Тесты для задания с матрицей представлены ниже (см. Рисунок 1). Матрицы во входных данных это, получившиеся матрицы после ввода пользователя.



***Рисунок 1 – тесты для матрицы***

Далее проверка тестов по критериям черного ящика (см. Рисунок 2).



***Рисунок 2 – проверка тестов по ЧЯ для матрицы***

## Рваный массив

Тесты для задания с рваным массивом представлены ниже (см. Рисунок 3). Рваные массивы во входных данных, это получившиеся массивы после ввода пользователя.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

***Рисунок 3 – тесты для рваного массива***

Далее проверка тестов по критериям черного ящика (см. Рисунок 4). Для рваных массивов нет смысле проверять какие массивы находятся внутри, так как нет разницы какой массив удалять.

Изображение выглядит как текст, снимок экрана, число, линия

Автоматически созданное описание

***Рисунок 4 – проверка тестов по ЧЯ для рваного массива***

## Строка

Тесты для задания со строкой представлены ниже (см. Рисунок 5).

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

***Рисунок 5 – тесты для строки***

Далее проверка тестов по критериям черного ящика (см. Рисунок 6).



***Рисунок 6 – проверка тестов по ЧЯ для строки***

# Приложение А

Ссылка на github

https://github.com/Poruch/LabWorks.git