|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №3  по дисциплине «Языки программирования»  Вариант 4 | | |
|  | | Работу выполнил  студент группы ПМИ-3,4-2023 2 курса  Руденко П.М.  «\_16\_» \_\_\_марта \_2025 г. |
| Работу проверила  Ракина В.Д. «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Задание 1 3](#_Toc176204057)

[Текст задания 3](#_Toc176204058)

[Алгоритм решения 3](#_Toc176204059)

[Тестирование 5](#_Toc176204060)

[Код программы 5](#_Toc176204061)

[Задание 2 6](#_Toc176204062)

[Текст задания 6](#_Toc176204063)

[Алгоритм решения 6](#_Toc176204064)

[Тестирование 7](#_Toc176204065)

[Код программы 7](#_Toc176204066)

**Задания 1-3**   
Выполнить все задания в одном проекте. Задания 1, 2 и 3 выполнить в виде методов одного класса. Задание 1 реализовать в виде конструкторов (кроме них, могут быть и другие конструкторы). Класс содержит единственное поле – двумерный массив. В задании 3 перегрузить необходимые операторы и посчитать значение матричного выражения (решение без перегрузки не принимается). Перегрузить метод ToString() — сформировать строку из двумерного массива для отображения его на экране в виде таблицы. Приложить в отчет скрин-шот проверки на онлайн-калькуляторе, чтобы показать, что выражение посчитано верно.   
  
**Задание 1 «Заполнение двумерных массивов»**

## Текст задания

Первый массив, размерностью n х m , заполняется данными, вводимыми с клавиатуры, так что заполнение ведется по строкам от первых элементов строки к последним.   
Второй массив, размерностью n х n, заполняется так: элементы, лежащие выше побочной диагонали, являются случайными числами из интервала [-65; 120], а лежащие на побочной диагонали и ниже, являются случайными числами из интервала [-3.5; 10.75].   
Третий массив, размерностью n х n, заполняется для произвольного n так же, как для n=5:   
11 10 4 3 1   
0 12 9 5 2   
0 0 13 8 6   
0 0 0 14 7   
0 0 0 0 15

**Задание 2 «Работа с двумерными массивами»**   
В городе П. есть m банков. Известны величины задолженностей банков друг другу. Укажите банк с максимальным долгом   
**Задание 3. «Работа с двумерными массивами как с матрицами»**   
2\*А-В^(т)\*С

## Алгоритм решения

## Задание 1: Заполнение двумерных массивов

1. **Первый конструктор** (Matrix(int n, int m)):

Создаём массив n x m и запрашиваем с пользователя ввести элементы с клавиатуры по строкам. Если ввели не число, ставим 0.

1. **Второй конструктор** (Matrix(int n, bool isRandom)):

Для n x n заполняем случайными числами: выше побочной диагонали — от -65 до 120, на и ниже — от -3.5 до 10.75.

1. **Третий конструктор** (Matrix(int n)):

Заполняем как в примере для n=5. Формула 15 - j + i даёт нужные числа, а ниже диагонали ставлю 0.

**Задание 2: Работа с двумерными массивами**

Метод **GetMaxDebtBank()** считает сумму долгов каждого банка (по строкам) и возвращает номер банка с максимальной суммой. Нумерация с 0, но в выводе я добавил +1, чтобы было с 1.

**Задание 3: Работа с двумерными массивами как с матрицами**

Перегрузил операторы:

\* для скаляра (2 \* A),

\* для умножения матриц (B^T \* C),

- для вычитания (2\*A - B^T\*C).

Добавим метод **Transpose()** для транспонирования B.

Выражение 2\*A - B^T \* C считаем пошагово и выводим все промежуточные матрицы.

**Перегрузка ToString**

Выводим массив как таблицу с числами, выровненными (2 знака после запятой).  
  
и делаем тесты для всех заданий

## Тестирование и т.д., код успешно выолняется

## Код программы

[Ссылка на githab](https://github.com/dungeon-meowster/c-sharp-2-course-psu/blob/main/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%E2%84%963/ex1-3.cs)

**Задания 4 – 8** выполнить в виде статических методов одного класса, но отдельно от заданий 1-3. В задании 4 бинарные файлы, содержат числовые данные, исходный файл необходимо заполнить случайными данными, заполнение организовать отдельным методом. В задании 5 бинарные файлы содержат величины типа struct, заполнение исходного файла необходимо организовать отдельным методом, обязательно использовать xml сериализацию, решение без сериализации не принимается. В задании 6 в текстовом файле хранятся целые числа по одному в строке, исходный файл необходимо заполнить случайными данными, заполнение организовать отдельным методом. В задании 7 в текстовом файле хранятся целые числа по несколько в строке, исходный файл необходимо заполнить случайными данными, заполнение организовать отдельным методом. В задании 8 в текстовом файле хранится текст. Необходимо решить по 1 задаче из каждого задания согласно вашему варианту. Каждое задание оценивается по 1 баллу. Максимально за лабораторную работу можно получить 10 баллов (8 баллов за решение задач + 2 балла за оформление отчета).   
  
**Текст задания**  
  
**Задание 4**. Бинарные файлы Подсчитать количество пар противоположных чисел среди компонент исходного файла.   
**Задание 5**. Бинарные файлы и структуры Информация о багаже пассажира описывается массивом, где каждый элемент содержит название единицы багажа (чемодан, сумка, коробка и т.д.) и ее массу. Дан файл, содержащий сведения о багаже нескольких пассажиров. На сколько багаж максимальной массы отличается от багажа минимальной массы?   
**Задание 6.** Текстовые файлы Для заданного файла возвратить true, если он не содержит нуля, и false в противном случае   
**Задание 7**. Решить задачу с использованием структуры «текстовый файл» Вычислить максимальный элемент   
**Задание 8.** Текстовый файл Переписать в другой файл строки, оканчивающиеся на заданный символ

**Алгоритм решения**

Решим задачи 4–8, создав класс FileTasks с пятью статическими методами.   
**Для задачи 4** напишем метод заполнения бинарного файла случайными числами от -100 до 100, потом считал их в список и посчитал пары противоположных чисел, сравнивая каждый элемент с остальными.   
**В задаче 5** сделаем структуру Luggage (багаж на англ. языке) с именем и весом, заполнил бинарный файл случайными данными (имя из массива, вес от 0 до 50), добавим XML-сериализацию в отдельный файл, а затем находим разницу между максимальным и минимальным весом, читая только веса из бинарного файла.

**Для задачи 6** заполним текстовый файл случайными числами от -50 до 50 по одному в строке, потом проверим, есть ли ноль, читая строки и прерываясь, если найден.

**В задаче 7** заполним текстовый файл случайными числами от -100 до 100 по несколько в строке, затем находим максимум, разбивая строки на числа и сравнивая их.   
**Для задачи 8** создаём текстовый файл, вводим символ с клавиатуры и переписываем в новый файл только строки, заканчивающиеся на этот символ, проверяя через EndsWith. В Main для каждой задачи запросим нужные данные с клавиатуры, вызвав методы заполнения и решения. Выведём результаты.

## Тестирование

## Создаём файл text.txt перед выполнением кода

## Код программы

[Ссылка на githab](https://github.com/dungeon-meowster/c-sharp-2-course-psu/blob/main/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%E2%84%963/ex4-8.cs)