

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное
учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники
Отделение информационных технологий
Направление информатика и вычислительная техника

Отчет
по лабораторной работе №6

по дисциплине
«ПРОГРАММИРОВАНИЕ»

Вариант № 26

**Структуры. Перегрузка операторов. Наследование. Классы. Методы
классов**

Выполнил:

Студент группы 8В32

Д.О.Карташов

Проверил:

Ассистент ОИТ ИШИТР

А.Ю.Малкин

Томск 2024

Цель работы

Получить навыки работы со структурами, перегрузками операторов, наследованием, классами, методами классов на языке программирования C++.

Задание

Задания, представленные ниже, реализовать с применением связки технологий CMake, GCC, VS Code. Проект выполнять как многофайловый: логика выполнения выносится в отдельный файл, в `main.cpp` остаётся линковка с нужными заголовочными файлами, определение параметров для классов и их методов, их вызов.

1. Реализовать класс "Матрица" - реализация математического объекта, таблицы целых чисел. Класс должен иметь:

- Конструктор по умолчанию, в котором создаётся нулевая матрица размерностью 2x2.
- Конструктор, принимающий два аргумента типа `int` - количество строк и столбцов матрицы. Элементам матрицы присваиваются случайные значения.
- Деструктор.
- Оператор `+` для сложения двух матриц.
- Оператор `-` для вычитания двух матриц.
- Оператор `*` для умножения матрицы на число.
- Оператор `*` для умножения матриц.
- Оператор `=` для присваивания одной матрице значений другой матрицы.
- Оператор `==` для определения равенства матриц (истинно, если они имеют одинаковые размеры и их соответствующие элементы равны).
- Оператор `<<` для отображения матрицы в консоль посредством `std::cout`.
- Оператор `>>` для ввода чисел в матрицу посредством `std::cin`.

- Метод, возвращающий максимальное значение среди элементов матрицы
- Конструктор по умолчанию, в котором создаётся нулевая матрица размерностью 2x2.
- Конструктор, принимающий два аргумента типа `int` - количество строк и столбцов матрицы. Элементам матрицы присваиваются случайные значения.
- Деструктор.
- Оператор `+` для сложения двух матриц.
- Оператор `-` для вычитания двух матриц.
- Оператор `*` для умножения матрицы на число.
- Оператор `*` для умножения матриц.
- Оператор `=` для присваивания одной матрице значений другой матрицы.
- Оператор `==` для определения равенства матриц (истинно, если они имеют одинаковые размеры и их соответствующие элементы равны).
- Оператор `<<` для отображения матрицы в консоль посредством `std::cout`.
- Оператор `>>` для ввода чисел в матрицу посредством `std::cin`.
- Метод, возвращающий максимальное значение среди элементов матрицы.

2. Создать класс "Свёртка", который наследуется от класса "Матрица". Реализовать метод получения матрицы в результате свёртки двух других матриц.

3. Создать класс "Субдискретизация", который наследуется от класса "Матрица". Реализовать метод получения матрицы в результате операции подвыборки исходной матрицы по максимальному значению.

Ход работы

На рисунке 1 демонстрируется сумма, разность, произведение и результат сравнения матриц $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ и $\begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ соответственно.

```
2 1 4
3 6 5
8 7 10

0 3 2
5 4 7
6 9 8

2 -2 2
5 -5 5
8 -8 8

false
```

Рисунок 1 – Сумма, разность, произведение и результат сравнения

На рисунке 2 демонстрируется результат умножения матрицы $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ на 2.

```
2 4 6
8 10 12
14 16 18
```

Рисунок 2 – Результат умножения на 2

На рисунке 3 демонстрируются результаты вызова конструкторов матрицы: без аргументов и с аргументами 3, 3 соответственно.

```
0 0
0 0

10391 12815 3051
11134 21889 19633
6438 10406 15036
```

Рисунок 3 – Результаты вызова конструкторов матрицы

На рисунке 4 демонстрируется результат применения свёртки к матрицам

$$\begin{pmatrix} 12 & 20 & 30 & 0 \\ 8 & 12 & 2 & 0 \\ 34 & 70 & 37 & 4 \\ 112 & 100 & 25 & 12 \end{pmatrix} \text{ и } \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
24 22 30
78 49 6
134 95 49
```

Рисунок 4 – Результат свёртки

На рисунке 5 демонстрируется результат применения субдискретизации к матрице

$$\begin{pmatrix} 12 & 20 & 30 & 0 \\ 8 & 12 & 2 & 0 \\ 34 & 70 & 37 & 4 \\ 112 & 100 & 25 & 12 \end{pmatrix} \text{ с ядром размера } 2 \times 2.$$

```
20 30
112 37
```

Рисунок 5 – Результат субдискретизации

Вывод

Получить навыки работы со структурами, перегрузками операторов, наследованием, классами, методами классов на языке программирования C++.

Приложение 1 – Ссылка на репозиторий с исходными кодами программ

Ссылка на репозиторий: *<https://github.com/The-Aozzi/Lab6>*