**Отчет по РГР**

**Тема «Заказы типографии»**

**Выполнил: Еремин Александр**

**СОДЕРЖАНИЕ**

[Функционал программы 3](#_Toc124765125)

[Структура программы 4](#_Toc124765126)

[Скриншот 1 — Все файлы проекта 4](#_Toc124765127)

[Скриншот 2 — Реализация меню программы 5](#_Toc124765128)

[Скриншот 3 — Объявление функций для работы меню 6](#_Toc124765129)

[Скриншот 4 — Класс «Srting» 7](#_Toc124765130)

[Скриншот 5 — Класс «Typography» 8](#_Toc124765131)

[Скриншот 6 — Класс «Book» 8](#_Toc124765132)

[Скриншот 7 — Класс «Poster» 9](#_Toc124765133)

[Скриншот 8 — Класс «BusinessCard» 9](#_Toc124765134)

[Скриншот 9 — Класс «Vector» 10](#_Toc124765135)

# Функционал программы

Тема данной расчетно-графической работы: «заказы типографии». Была написана программа, содержащая следующий функционал:

1. Оформление заказа. На выбор доступно 3 вида заказа: книга, плакат (постер) и визитка. При выборе книги, пользователю предлагается ввести название книги, имя и фамилию автора, количество книг. При заказе постера появляется срока для ввода типа постера (например, рекламный, учебный, политический и т.д.). При заказе визитки вводится имя и фамилия, предоставляемая услуга, номер телефона.
2. Удаление заказа. Также выбирается один из трех предложенных типов заказа, после чего вводятся соответствующие заказу данные.
3. Просмотр заказов. На экран выводятся оформленные пользователем заказы.
4. Прочитать заказы из файла. Заказы считываются и сохраняются в динамический массив (вектор) из текстового файла.
5. Сохранить заказы в файл. Заказы из динамического массива (вектора) записываются и сохраняются в текстовый файл.
6. Завершение работы программы.

# Структура программы

Проект состоит из 9 файлов: 8 содержат код программы и одна — это текстовый файл, куда будет сохраняться информация о заказах. Все файлы проекта представлены на скриншоте 1.

## Все файлы проекта

Файл «main.cpp» реализует меню программы. Он (файл) содержит в себе цикл и функции. Цикл позволяет выводить меню на экран каждый раз после выбора пользователя. Программа закончит свою работу, как только пользователь выберет «6». Реализация меню программы представлена на скриншоте 2.



## Реализация меню программы

Функции, использующиеся в цикле, такие как «mainMenu», «order», «deleteOrder» и т.д. объявлены в начале файла. Определены же они после основного цикла меню. Объявление функций для работы меню представлены на скриншоте 3.

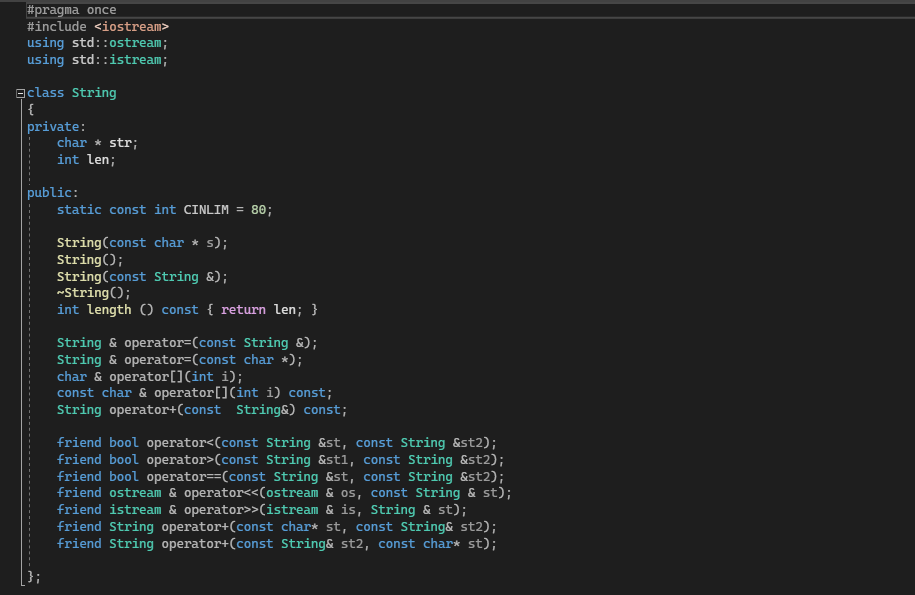


## Объявление функций для работы меню

Далее идут файлы «String.cpp» и «String.h». «String.h» - заголовочный файл, в нем описан класс «String»: определены данные класса (private) и объявлены методы класса (public). В файле «String.cpp» непосредственно определены методы класса. По точно такому же принципу разделены файлы «Typography.h» и «Typography.cpp», «TypographyOrder.h» и «TypographyOrder.cpp». Сделано это для более удобного чтения кода.

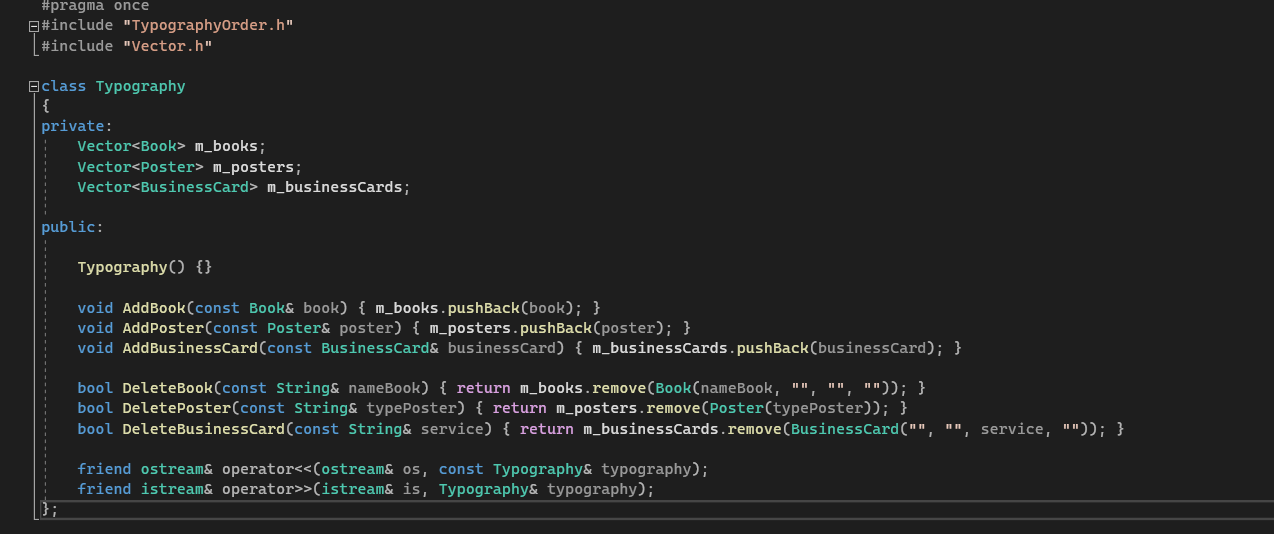
Класс «String» нужен для работы со строками (хранение строк, их длины, присвоение строк и т.д.). Данный класс был позаимствован и немного видоизменен из книги Стивена Прата «Язык программирования C++. Лекции и упражнения, 6-е изд.».

Код класса «String» представлен на скриншоте 4.



## Класс «Srting»

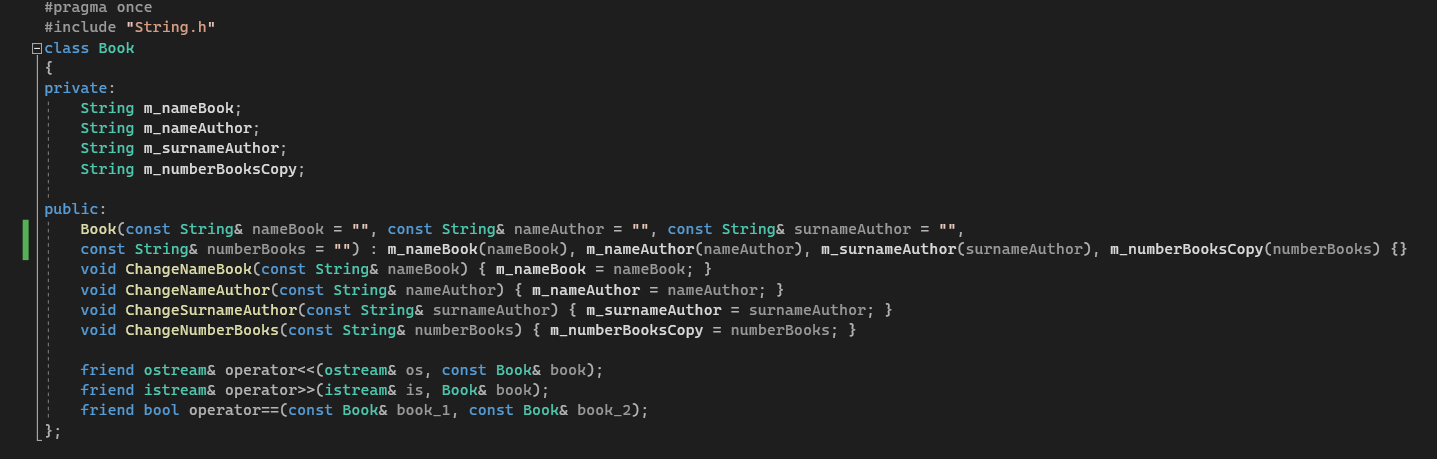
Класс «Typography» содержит три динамических массива, в которых по отдельности хранятся заказы книг, плакатов и визиток. Также он позволяет добавлять и удалять эти заказы. Класс «Typography» представлен на скриншоте 5.



## Класс «Typography»

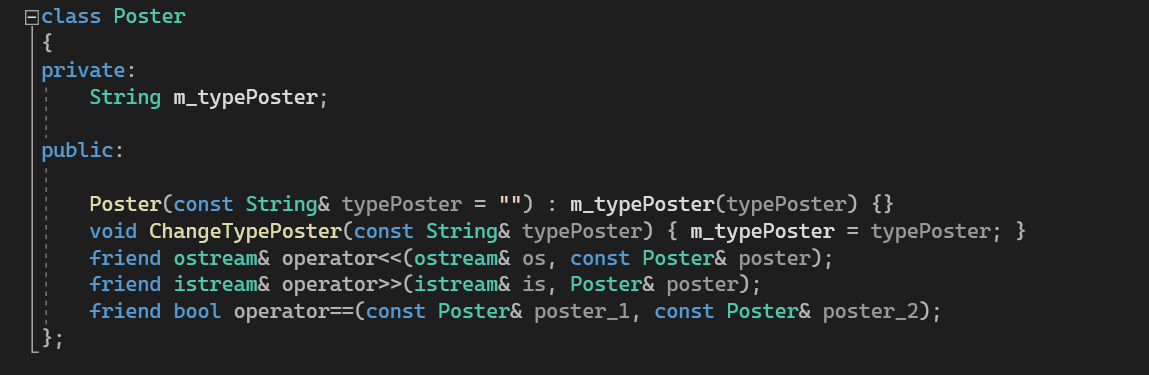
Файл «TypographyOrder.h» содержит в себе три класса: «Book», «Poster», «BusinessCard». Данные классы позволяют вводить новые заказы для книг, постеров и визиток.

Класс «Book» представлен на скриншоте 6.



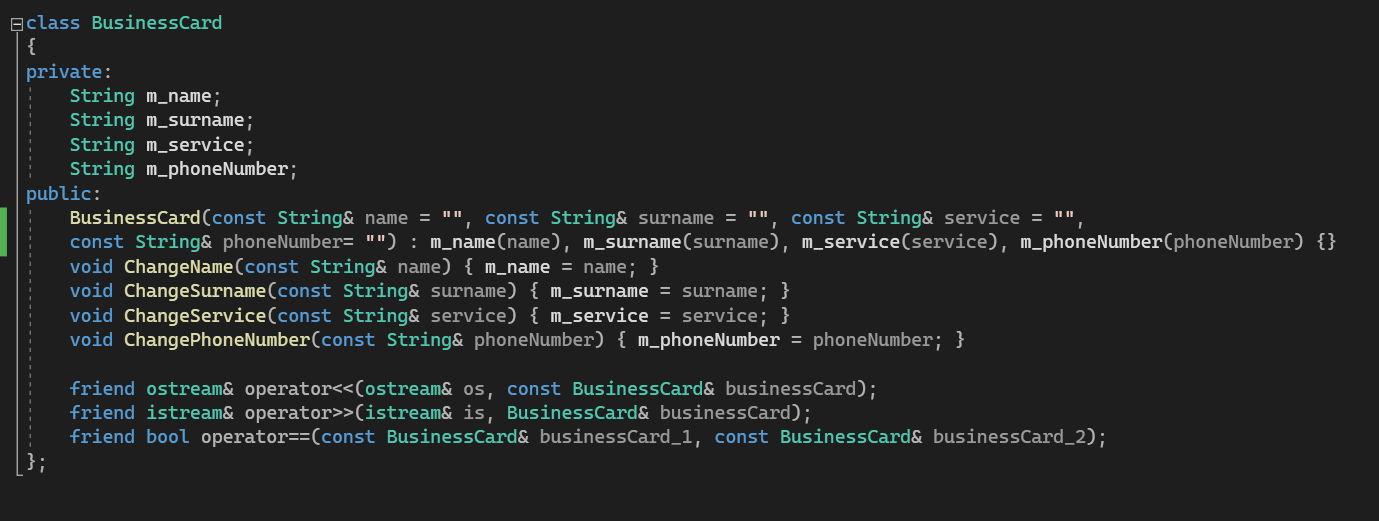
## Класс «Book»

Класс «Poster» представлен на скриншоте 7.



## Класс «Poster»

Класс «BusinessCard» представлен на скриншоте 8.



## Класс «BusinessCard»

Заголовочный файл «Vector.h» содержит шаблон вектора, который нужен для создания динамических массивов, в которых будут храниться разные типы заказов. Преимущество данного шаблона в том, что можно создавать массивы, длина которых устанавливается во время выполнения, без необходимости явно выделять и освобождать память с помощью операторов «new» и «delete». Также такой динамический массив может хранить вместе разные типы данных, например, строки и целые числа. Класс вектор представлен на скриншоте 9.



## Класс «Vector»