

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Новосибирский государственный университет экономики и управления «НИНХ»**

**(ФГБОУ ВО «НГУЭУ», НГУЭУ)**

Кафедра информационных технологий

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

Дисциплина: Прикладные информационные технологии

Ф.И.О студента: Фабрициус Никита Евгеньевич

Направление: 09.03.02 Информационные системы и технологии

Номер группы: ИС101

Номер зачетной книжки: 212114

Номер варианта расчетно-графической работы: 4

Проверил: Ковригин Алексей Викторович, кандидат педагогических наук, доцент

Новосибирск 2023

**Описание проекта, общие положения.**

Исходя из темы проекта было принято решение создания небольшого консольного приложения, для удобного хранения и обработки информации, передаваемой внутри компании.

Структура проекта была сгенерирована с нуля. При создании проекта были учтены главные потребности компании о способах передачи, хранения и обработки информации. В прототип вошли такие данные как: фамилия заказчика, все возможные замеры и тип изготовляемого продукта.

Сама структура приложения составляет в себе множество функций выполняющих определенную задачу и главное меню в котором прописывается требуемое действие и вызов нужной функции для выполнения этого действия.

Описание функций

**Описание функционала.**

void Menu() – функция графически отображает функционал приложения (выводит список возможных действий).

void main() - функция вызывает функцию Menu() и DataInitialization(), а так же цикл в тором программа ожидает ввод одного из 7 чисел для выполнения операции.

Void DataInitialization() – функция создает временный файл, в котором будет хранится актуальная информация. Т.е. создает буффер в виде .txt файла.

Void DataEntry() – функция предназначена для ввода данных в приложение. В начале, спрашивается, как будут вводится данные: из файла или в ручную. После, в зависимости от выбора функция либо считывает данные из указанного файла или предоставляет возможность заполнения пустых ячеек данных.

Void DataReading() – функция будет считывать данные из указанного файла, подходящего под параметры считывания. Не все файлы могут быть считаны программой, только ранее созданные самой программой, либо оформленные по нужному шаблону.

Void Print() – функция нужна для вывода данных обрабатываемых в данный момент в приложении.

bool DataCleaning() – функция с начала спрашивает номер заказа, а после удаляет эту ячейку.

Void DataChange() – функция вызывает функцию Copy(), которая копирует данные в новый буфферный файл, затем спрашивает номер заказа, данные которого хотим изменить, и после дает возможность изменить все пункты выбранной ячейки информации.

Void DeleteData() – функция вызывает функцию Copy(), которая копирует данные в новый буфферный файл, но уже без выбранного ранее удаляемого заказа.

Void AddData() – функция получает на вход количество элементов в буфферном файле благодаря функции AmountOfData(), которая считает количество заказов в буфферном файле, после увеличивает это значение на 1 и спрашивает задаваемые параметры, записывая их в конец файла.

Void SaveData() – функция нужна для сохранения обработанной информации в виде файла. Делает она это путем считывания информации из буфферного файла в файл с заданным нами значением.

**Минусы проекта.**

1. Нет возможности изменить один из пунктов ячейки заказа. Нужно изменять все параметры, что является не совсем удобным.
2. Запись в файл будет не понятна без объяснений.
3. При каждом запуске приложения, нужно снова вводить интересующую нас информацию. Это связанно с очисткой буфферного файла при завершении работы приложения
4. При вводе некорректной информации программа ломается без возможности восстановить свою работоспособность.

**Планы на доработку проекта.**

1. Сделать ячейки с информацией о заказе более раздробленными (возможность брать и изменять определенный пункт в заказе, не трогая остальную информацию).
2. Сделать графический интерфейс
3. Изменить сохранение информации при помощи .txt, использовать более удобные методы, пример: SQL.
4. Реализовать структура проекта через классы.
5. Добавить обработку ошибок с вводом некорректной информации. (Тип вводимой информации)

**Скриншоты работы программы.**

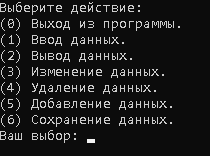
****

Рис. Меню



Рис. Ввод данных

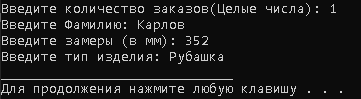


Рис. 2.1 Ввод данных (вручную)



Рис. 2.2 Ввод данных (из файла)

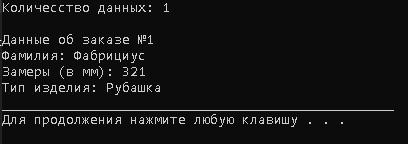


Рис. 3 Вывод данных



Рис. 4.1 Изменение данных

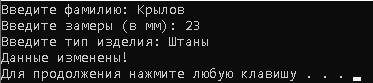


Рис. 4.2 Изменение данных



Рис. 5.1 Удаление данных



Рис. 5.2 Удаление данных

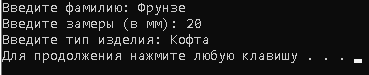


Рис. 6.1 Добавление данных

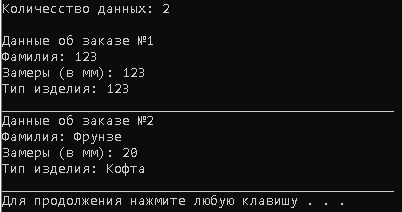


Рис. 6.2 Добавление данных



Рис. 7.1 Сохранение данных



Рис. 7.2 Сохранение данных

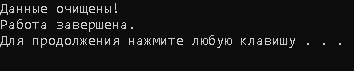


Рис. 8 Завершение работы приложения