**Техническое задание.**

**Тема РГР:** Заказы фирмы разработки ПО

**Исполнитель:** Иванова Дана Алексеевна

**Группа:** ИС101

**Номер зачетной книжки:** 212105

**Направление:** 09.03.02 «Информационные системы и технологии»

**Профиль:** «Информационные системы и технологии»

**Описание проекта, общие положения.**

Создать программу по управлению объектами данных по указанной теме с использованием ООП и реализовывать изученные паттерны.

Необходимо создать минимум 3 объекта сущности разных видов и указать взаимодействие между объектами. Реализовать интерфейсы ввод, вывода и редактирования сущностей. Реализовать ручной ввод сущностей, загрузку из файла и сохранения в файл, удаление выбранных сущностей.

Данные между запусками программы хранить в структурированном текстовом файле.0

Исходя из темы проекта было принято решение создания некоторого аналога проектной программы, которая будет использоваться внутри компании.

Примерная структура была взята из FirePlanetStudio, которой я пользуюсь на своей работе. Там существуют базы: инциденты, претензии. По их образу был разработан прототип данной РГР.

Программа будет:

1. Считывать данные из файла или вводить их вручную

2. Делать определенные действия с данными (менять, добавлять, выводить в консоль и т.д.)

3. Записывать данные в файл

Это программа, которая работает с данными и выполняет все действия благодаря временному файлу. Этот файл выступает в роли буфера, куда записывается информация, с которой непосредственно работает программа. Поэтому не используются никакие массивы, только файлы и функции.

**Описание функций.**

void DataInitialization() – инициализация данных. Создание временного файла, в котором будет храниться актуальная информация, т.е. создание буфера.

Это сделано с помощью потока ofstream –это поток для работы с файлами в режиме записи

void DataEntry() – ввод данных вручную. Сначала вводим количество данных, затем идет открытие буферного файла. С помощью цикла записываем данные в файл.

void DataReading(string fileName) – чтение данных из файла. Тут требуются два потока: ifstream – читает данные из нужного файла; ofstream – записывает данные в буферный файл. Считывает данные из файла, название которого введено.

void Print() – вывод данных. Считывает количество данных, затем проходит по циклу и считывает данные. После выводит данные в консоль.

bool DataCleaning() – очистка данных. Очищает буферный файл до закрытия программы.

void DataChange() – изменение данных. Сохраняет данные в еще один буферный файл, очищает текущий файл (выбранный элемент), и записывает данные в него. Используется функция Copy. После дополнительно считываем данные. Удаляет дополнительный буферный файл.

void Copy() – копирование данных. Создает еще один буферный файл. Записывает из старого буферного файла данные в новый.

int AmountOfData() – функция, которая выводит количество данных.

void DeleteData() – удаление данных. Снова используется Copy. Открывает дополнительный буфер для чтения. Считывает количество данных и предлагает выбрать элемент для удаления. Затем удаляет этот элемент.

void AddData() – добавление данных. Открывает буферный файл, изменяет количество данных. Затем записывает нужные данные в конец файла.

void SaveData(string fileName) – сохранение данных. Вводим название файла, куда хотим сохранить данные. Считывает данные из буферного файла и записывает в новый, введенный нами файл.

**Минусы проекта.**

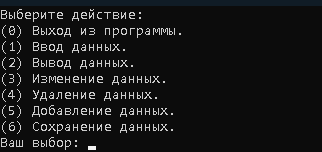
1. Оптимизация. Данный код является шаблонным/черновым, который выполняет поставленный функционал, но его можно оптимизировать.
2. Если вводим что-то не то, программа выдает "Invalid number", но при этом объект все равно удаляется из файла и программа по сути ломается. Нужно доработать этот момент.

**Планы на доработку проекта.**

1. Отредактировать данный код так, чтобы убрать повторяющиеся строчки кода и т.д.
2. Убрать запись данных в .txt, использовать таблицы, либо SQL.

**Скриншоты работы программы.**

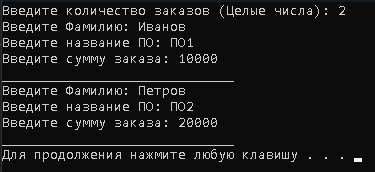
**Меню**

****

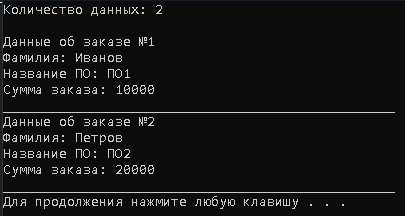
**Ввод данных (выбор способа добавления данных)**

****

**Добавление данных вручную**

****

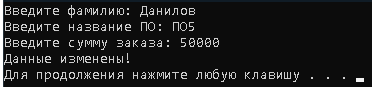
**Вывод данных**

****

**Изменение данных (выбор элемента)**

****

**Изменение данных**

****

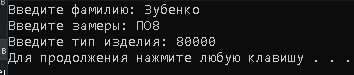
**Удаление данных (выбор элемента)**

****

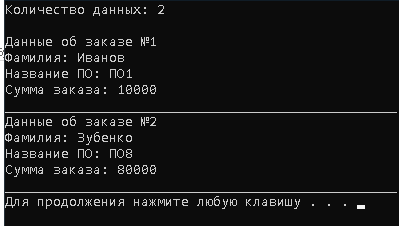
**Удаление данных**

****

**Добавление данных**

****

**Вывод данных после добавления новых**

****

**Сохранение данных (ввод названия файла, куда сохранится база данных)**

****

**Сохранение данных**

****