Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

Рег. № \_\_\_\_\_\_\_\_\_\_ Кафедра информационных систем и технологий

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_\_\_\_Название\_\_\_Разработать программный модуль «Картотека недвижимости»\_\_\_\_\_\_\_\_

по дисциплине \_Название\_\_\_\_Основы конструирования программ\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Урбанович Владислав Сергеевич (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_1920\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc70793473)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc70793474)

[**1.1. Общие положения 4**](#_Toc70793475)

[**1.2. Постановка задачи 4**](#_Toc70793476)

[**1.3. Спецификация входных и выходных данных 4**](#_Toc70793477)

[**1.4. Выбор средства 5**](#_Toc70793478)

[**1.5.Описывается требуемое оборудование и программное обеспечение 6**](#_Toc70793479)

[2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 7](#_Toc70793480)

[**2.1. Алгоритм решения 7**](#_Toc70793481)

[**2.2. Описание программы 7**](#_Toc70793482)

[**2.3. Результаты работы программы 9**](#_Toc70793483)

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 13](#_Toc70793484)

[ЗАКЛЮЧЕНИЕ 15](#_Toc70793485)

[ПРИЛОЖЕНИЕ 17](#_Toc70793486)

# ВВЕДЕНИЕ

Была поставлена задача разработать программный продукт «Бизнес процессы ресторана», предназначенный для использования работниками ресторана. В программном продукте необходимо реализовать учёт блюд, создание меню, оформление заказов. В программе должна содержать информацию о блюдах, возможность взаимодействовать с блюдом, так же создавать меню и так далее.

Программа создана для учёта блюд и меню, автоматизации заказов блюд клиентами, в частности, и автоматизации заведения в общем.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять запросы ресторана, а также должен быть возможность пополнять базу данных, для учёта блюд, меню и заказов. Основная же задача стоит в высокой оптимизация, чтобы пользователь могли запустить даже на самом слабом устройстве.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны владельцев ресторана. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было пользователем приятно пользоваться программой. Так же требуется предусмотреть возможность добавления новых функций в программе.

Для разработки программы был использован текстовый файл и язык программирования C++ разработанный в среде Visual Studio компаний Microsoft.

Visual Studio-линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой Windows Forms, а также веб-сайты, веб-приложения, веб-службы, как в родном, так и в управляемом кодах для всех платформ, поддерживающих Windows, Windows Mobile, Windows Ce, .Net Framework, Xbox, Windows Phone .Net Compact Framework и Silverling.

# 1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Общие положения

Оптимизация процессов любого бизнеса ведёт к серьёзной экономии не только денежных ресурсов, но и времени. Для обеспечение подобного рода операций необходимо использовать технологии, которые будут обеспечивать возможность подобной работы.

Была поставлена задача разработать программный продукт «Бизнес процессы ресторана», предназначенный для использования работниками ресторана. В программном продукте необходимо реализовать учёт блюд, создание меню, оформление заказов. В программе должна содержать информацию о блюдах, возможность взаимодействовать с блюдом, так же создавать меню и так далее.

Ресторан имеет помещения, в которых можно питаться (за столиком), танцевать, приготавливать пищу. Клиент может либо заказать столик заранее, либо (если будут свободные места) придти без предварительной заявки. Ресторан расписывает заранее столики по заявкам и рассаживает по свободным местам. Клиенты по заявкам могут заранее заказать меню. Можно выбрать заказ блюд из меню, представленного официантом. Затем заказ на блюда относится на кухню, где эти блюда изготавливают и подносятся клиенту. По окончанию либо до окончания своего присутствия в ресторане клиент рассчитывается за питание и уходит.

Данный программный продукт выгодно отличается от других подобных тем что пользуясь им, пользователь может не только структурированно хранить данные блюд, меню и клиентов, но и производить различные взаимодействия с ними (добавление/изменение удаление).

## 1.2. Постановка задачи

Исходя из задач, описанных выше главными требованиями к языку, на котором будет написана программы — это тонкая настройка и возможность использованием минимум вычислительных ресурсов добиваться максимум результата.

## 1.3. Спецификация входных и выходных данных

В данном программном продукте используются следующие типы данных типа данных: sting, double. Так же существуют пользовательские типы данных: . dish, menu, date, table, order

**boll** – используется только внутри программы, для проверки, подсчёта, иннерваций и так далее. Пользователь на данный этого типа влиять и пользоваться не может.

**String** – используется в двух случаях. Либо для наименования данных (введение информации о договоре). Во втором же случае данный тип данных используется для вывода всех данных на экран.

**Int** – используется для реализации различных алгоритм (пользователь не работает с этим типом данных) и так же для наименования данных.

**Dish** – хранит данные блюда.

**Menu** – хранит данные мню.

**Date** – предназначен для работы с датой.

**Table** – предназначен для вывода данных в таблицы.

**Order** – хранит данные заказа.

## 1.4. Выбор средства

Для выполнения данных задач очень подходит такой язык программирования как с++. Так как его из-за того, что он является весьма низкоуровневым языком программирования, а значит он больше других приближен к машинному коду, из-за этого с++ дает возможность оптимизировать проект так что даже на очень старых устройствах он будет работать.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных поля классов. Способ хранения данных – динамические массивы. Каждая логическая завершенная данных подзадача программы реализована в виде методов. Построение программного кода соответствует с <<C++ Code Convention>>. К защите курсовой предоставляется: консольное приложение и пояснительная записка

Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 – Windows Server 2016. Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений – голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 – последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

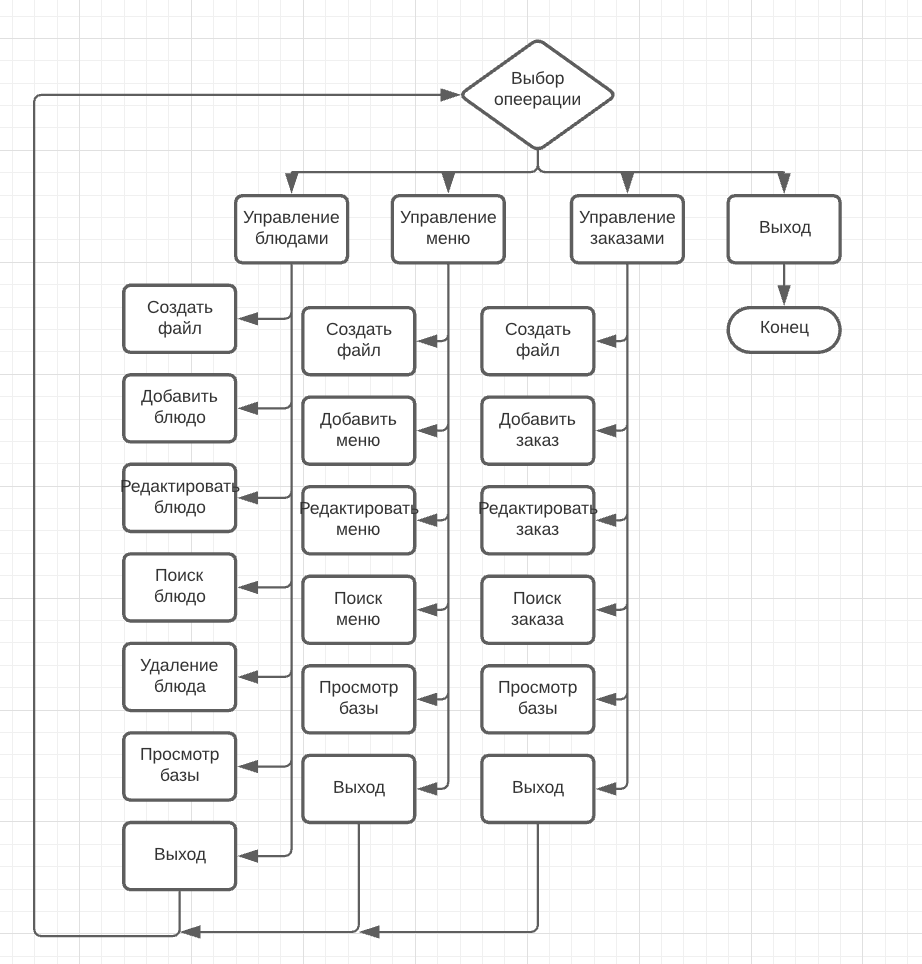
## 1.5.Описывается требуемое оборудование и программное обеспечение

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор Intel Core i3;
* оперативная память DDR3 512 МБ или больше;
* минимально свободного места на жёстком диске 512 МБ;
* мышь;
* операционная система Windows 7/8/10.

# 2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

## 2.1. Алгоритм решения



## 2.2. Описание программы

В данной программы используются следующие библиотеки:

**#include <math.h>** - заголовочный файл стандартной библиотеки языка программирования С, разработанный для выполнения простых математических операций. Большинство функций привлекают использование чисел с плавающей точкой. C++ также реализует данные функции для обеспечения совместимости, все они содержатся в заголовочном файле cmath.

Все эти функции принимают double, если не определено иначе. Для работы с типами float и long double используются функции с постфиксами f и l соответственно. Все функции, принимающие или возвращающие угол, работают с радианами.

**#include <stdio.h>** — стандартный заголовочный файл ввода-вывода) заголовочный файл стандартной библиотеки языка Си, содержащий определения макросов, константы и объявления функций и типов, используемых для различных операций стандартного ввода и вывода. Функциональность унаследована от «портативного пакета ввода-вывода».

**#include <conio.h>** - консольный ввод-вывод) — заголовочный файл, используемый в старых компиляторах, работающих в операционных системах MS-DOS, для создания текстового интерфейса пользователя. Тем не менее, он не является частью языка программирования Си, стандартной библиотеки языка Си, ISO C или требуемой стандартом POSIX.

**#include <locale.h>** — заголовочный файл стандартной библиотеки языка программирования С, который используется для задач, связанных с локализацией.

**#include <cstdlib>** - заголовочный файл стандартной библиотеки языка Си, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие. Заголовок вполне совместим с C++ и известен в нём как cstdlib. Название «stdlib» расшифровывается как «standard library» (стандартная библиотека).

**#include <string>** - класс с методами и переменными для организации работы со строками в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от имени строчного типа данных (англ. string; с англ. — «строка»). В языке C++ и его предшественнике, языке программирования Си, нет встроенной поддержки строкового типа данных, вместо этого используется массив символов. string управляет строками, как и string.h в Си. string использует единственный объект string для организации работы со строками. Являясь частью стандартной библиотеки C++, эти объекты также являются частью стандартного пространства имён — std.

**#include <fstream>** - аголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл. Для манипуляции с данными файлов используются объекты, называемые потоками («stream»).

**#include <iostream>** - заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от Input/Output Stream («поток ввода-вывода»).

**#include <algorithm>** - заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций над контейнерами и над другими последовательностями. Все функции библиотеки расположены в пространстве имён std.

**Dish** – хранит данные блюда (рис 2.2.1).

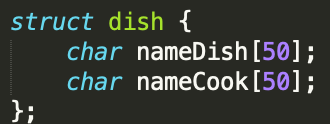


рис 2.2.1

Состоит из двух переменных:

*nameDish* – название блюда.

*nameCook* – название повара.

**Menu** – хранит данные мню (рис 2.2.2).

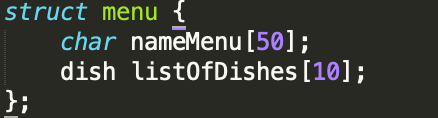
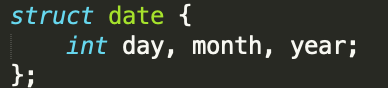


рис 2.2.2

*nameMenu* – название меню.

*listOfDishes* – список блюд в меню.

**Date** – предназначен для работы с датой (рис 2.2.3).



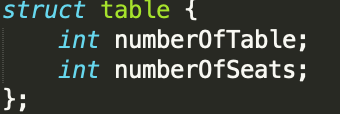
*Day* – день.

*Month* – месяц.

Year – год.

рис 2.2.3

**Table** – предназначен для вывода данных в таблицы (рис 2.2.4).



*numberOfTable* – количество строк.

*numberOfSeats* – количество столбцов.

рис 2.2.4

**Order** – хранит данные заказа (рис 2.2.5).

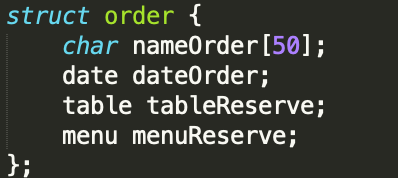


рис 2.2.5

*nameOrder* – название заказа.

*dataOrder* – дата принятия заказа.

*tableReserver* – таблица бля вывода заказа.

*menuReserve* – меню заказа.

Функции:

**Void newFileMenus()** – данная функция создаёт файл который хранит данные меню (рис 2.2.6).



**Void displayAllDishes()** – выводит из фала все данные о блюдах (рис 2.2.7)

рис 2.2.7

**Void editNoteOrders() –** выводи данные заказа из текстового файла(рис 2.2.8)



рис 2.2.7

## 2.3. Результаты работы программы

Результат работы программы представляет собой вывод данных из структур, которые были описаны в предыдущий главе.

# 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Так как была поставлена задача разработать консольное приложение, а значит при разработке интерфейса возникают все проблемы для консольных приложений. Этот факт необходимо учитывать при разработке интерфейса.

При запуске программы на консоль выводиться выбор операции. Для осуществления выбора необходимо ввести соответствующую цифру в консоль.

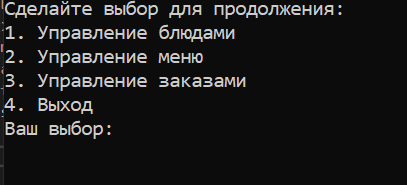


Рис 3.1

При выборе пункта № 1 на консоль выводиться все возможные операции с блюдами (рис 3.2)

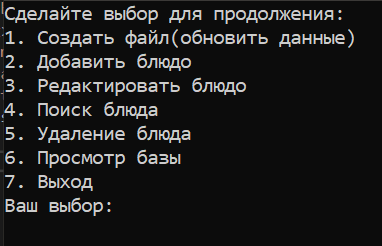


Рис 3.2

1. Создается файл, который служит в качестве базы данных, в которой храниться данные о блюдах. В том случае если файл уже создан, то файл очищается.
2. Происходит добавление блюда в файл.
3. Пользователь может редактировать уже созданное блюдо.
4. Производиться поиск блюда.
5. Удаление определённого блюда.
6. Просмотреть все блюда, находящиеся в файле.
7. Возвращение в главное меню.

При выборе пункта № 2 на консоль выводиться все возможные операции с меню (рис 3.2).

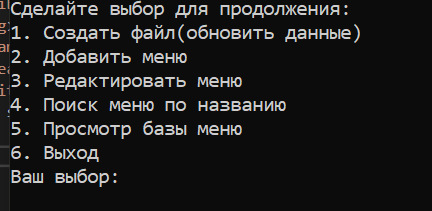


рис 3.2

1. Создается файл, который служит в качестве базы данных, в которой храниться данные о меню. В том случае если файл уже создан, то файл очищается.
2. Происходит добавление меню в файл, путём выбора составляющих из базы с блюдами.
3. Пользователь может редактировать уже созданное меню.
4. Производиться поиск меню по названию.
5. Просмотреть все меню, находящиеся в файле.
6. Возвращение в главное меню.

При выборе пункта № 3 на консоль выводиться все возможные операции с заказами (Рис 3.3).

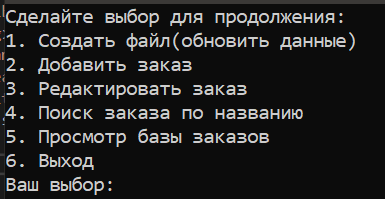


Рис 3.3

1. Создается файл, который служит в качестве базы данных, в которой храниться данные о заказах. В том случае если файл уже создан, то файл очищается.
2. Происходит добавление заказ в файл, путём выбора составляющих из базы с блюдами.
3. Пользователь может редактировать уже созданный заказ.
4. Производиться поиск заказа по названию.
5. Просмотреть все заказы, находящиеся в файле.
6. Возвращение в главное меню.

При выборе пункта № 4 происходит выход из программы.

# ЗАКЛЮЧЕНИЕ

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль «Учёт клиентов санатория», предназначенный для использования работниками санатория. Были реализованы задачи:

1. Хранение данных блюд.
2. Создание меню
3. Алгоритм заказа
4. Проверка данных

Цель была выполнены, программный продукт имеет следующие преимущества:

1. Работа программного продукта происходит быстро и максимально эффективно. Программа удовлетворяет запросам пользователя.
2. Имеет возможность пополнять базу данных, для расширения ассортимента блюд.
3. Так же программы имеет высокую оптимизацию. Из-за этого программа работает весьма быстра и запускается даже на самом слабом устройстве.

Так же данный программный продукт выгодно отличается от других представленных на рынке, возможностью не только хранить данные блюд и меню, но и возможность клиентами заказывать блюда, ещё она снабжена интуитивным интерфейс.

Таким образом программа работает без сбоев и все задачи были выполнены.

# СПИСОК ЛИТЕРАТУРЫ

1. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
2. Обзор обновлений и новых функций Windows 10 [Электронный ресурс]. – Microsoft, 2020. – Режим доступа: https://www.microsoft.com/ru-ru/windows/features. – Дата доступа: 22.05.2020.
3. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
4. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
5. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
6. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
7. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.

# ПРИЛОЖЕНИЕ

#include <math.h>

#include <stdio.h> //printf

#include <conio.h> //fstream(для коректной работы)

#include <locale.h> //Rus lang

#include <cstdlib> //Using system

#include <string> //Work with string(строки)

#include <fstream> //file

#include <iostream> //cout-cin

#include <algorithm> //enable sort algorithm

using namespace std;

struct dish;

struct menu;

struct date;

struct table;

struct order;

void manageMenuOrders();

void newFileOrders();

void addNoteOrders();

void editNoteOrders();

void searchNote\_NameOrders();

void displayAllOrders();

void manageMenuDishes();

void newFileDish();

void addNoteDish();

void editNoteDish();

void searchNoteDish();

void searchNote\_NameDish();

void searchNote\_NameCook();

void deleteNoteDish();

void displayAllDishes();

void manageMenuMenus();

void newFileMenus();

void addNoteMenus();

void editNoteMenus();

void searchNote\_NameMenus();

void displayAllMenus();

struct dish {

char nameDish[50];

char nameCook[50];

};

dish dishesArray[15];

int d = 0; // number of dishes array

int df = 0; // file of dishes array

struct menu {

char nameMenu[50];

d ish listOfDishes[10];

};

menu menusArray[15];

int m = 0; // number of dishes array

int mf = 0; // file of dishes array

struct date {

int day, month, year;

};

struct table {

int numberOfTable;

int numberOfSeats;

};

struct order {

char nameOrder[50];

date dateOrder;

table tableReserve;

menu menuReserve;

};

order orderArray[15];

int ord = 0; // number of dishes array

int ordf = 0; // file of dishes array

// Заказы

void manageMenuOrders()

{

bool OOAM = true;

do

{

printf("Сделайте выбор для продолжения:\n");

int choice;

do

{

printf("1. Создать файл(обновить данные)\n");

printf("2. Добавить заказ\n");

printf("3. Редактировать заказ\n");

printf("4. Поиск заказа по названию\n");

printf("5. Просмотр базы заказов\n");

printf("6. Выход\n");

printf("Ваш выбор: ");

cin >> choice;

system("cls");

} while (choice > 6);

switch (choice)

{

case 1: newFileOrders(); break;

case 2: addNoteOrders(); break;

case 3: editNoteOrders(); break;

case 4: searchNote\_NameOrders(); break;

case 5: displayAllOrders(); break;

case 6: OOAM = false; break;

default: printf("Ваш выбор неверен.\n");

}

} while (OOAM);

}

void newFileOrders()

{

fstream fileIO;

string pathFile = "OrdersFile.txt";

fileIO.open(pathFile, fstream::out);

fileIO.close();

fileIO.open(pathFile, fstream::out | fstream::in | fstream::binary);

ordf++; //Счетчик проверяющий создан ли файл

fileIO.close();

cout << "Файл успешно создан\n\n";

system("pause");

system("cls");

}

void addNoteOrders()

{

system("cls");

if (ordf != 0)

{

fstream fileIO;

string pathFile = "OrdersFile.txt";

fileIO.open(pathFile, fstream::in | fstream::out | fstream::binary);

cin.ignore();

printf("Введите название заказа: ");

cin.getline(orderArray[ord].nameOrder, 50);

printf("Введите день заказа: ");

cin >> orderArray[ord].dateOrder.day;

printf("Введите месяц заказа: ");

cin >> orderArray[ord].dateOrder.month;

printf("Введите год заказа: ");

cin >> orderArray[ord].dateOrder.year;

printf("Введите номер стола: ");

cin >> orderArray[ord].tableReserve.numberOfTable;

printf("Введите количество мест: ");

cin >> orderArray[ord].tableReserve.numberOfSeats;

cout << "Список меню:\n";

for (int i = 0; i < m; i++)

{

cout << i << "-" << menusArray[i].nameMenu << "\n";

}

//cin.ignore();

int idMenuAdd;

cout << "Введите номер меню, которое вы хотите добавить: ";

do

{

cin >> idMenuAdd;

if (idMenuAdd >= 0 && idMenuAdd < m)

{

orderArray[ord].menuReserve = menusArray[idMenuAdd];

}

else

{

cout << "Введите данные снова: ";

}

} while (idMenuAdd < 0 || idMenuAdd >= m);

fileIO.write((char\*)&orderArray, sizeof orderArray);

fileIO.close();

//Для проверки, состояния массивов

/\*fileIO.read((char\*)&orderArray, sizeof orderArray);

fileIO.close();\*/

printf("\nЗапись успешно добавлена\n\n");

ord++; //Увеличивается если добавляется еще один заказ.

system("pause");

system("cls");

}

else

{

system("cls");

printf("Файл не создан\n\n");

system("pause");

}

system("cls");

}

void editNoteOrders()

{

system("cls");

if (ordf != 0) //Проверка, создан ли файл

{

fstream fileIO;

fileIO.open("OrdersFile.txt", fstream::in | fstream::out | fstream::binary);

char findOrderName[50];

cin.ignore();

printf("Введите название заказа для изменения меню: ");

cin >> findOrderName;

fileIO.write((char\*)&orderArray, sizeof orderArray);

bool temp = false;

for (int j = 0; j < ord; j++)

{

if (string(findOrderName) == string(orderArray[j].nameOrder))

{

temp = true;

cout << "Название заказа: " << orderArray[j].nameOrder << "\n";

cout << "Дата заказа: " << orderArray[j].dateOrder.day << "." << orderArray[j].dateOrder.month << "." << orderArray[j].dateOrder.year << "\n";

cout << "Номер стола: " << orderArray[j].tableReserve.numberOfTable << "\n";

cout << "Количество мест: " << orderArray[j].tableReserve.numberOfSeats << "\n";

cout << "Название меню: " << orderArray[j].menuReserve.nameMenu << "\n";

cout << "Список блюд(номер блда, название блюда, имя повара): \n";

for (int i = 0; string(orderArray[j].menuReserve.listOfDishes[i].nameDish) != string("\0"); i++)

{

cout << i << " - " << orderArray[j].menuReserve.listOfDishes[i].nameDish << " - " << orderArray[j].menuReserve.listOfDishes[i].nameCook << "\n";

}

// Начинать от сюда. Ввыод списком названий меню, выбор меню по id.

cout << "Список меню:\n";

for (int i = 0; string(menusArray[i].nameMenu) != string("\0"); i++)

{

cout << i << " - " << menusArray[i].nameMenu << "\n";

}

cout << "Введите номер меню, на которое вы хотите изменить существующее меню: ";

int switchChoice;

do

{

cin >> switchChoice;

if (switchChoice >= 0 && switchChoice < m)

{

orderArray[j].menuReserve = menusArray[switchChoice];

}

else

{

cout << "Введите данные снова: ";

}

} while (switchChoice < 0 || switchChoice >= m);

}

}

if (!temp)

cout << "\nМеню с таким названием не обнаружено. Попробуйте ввести имя еще раз";

fileIO.write((char\*)&orderArray, sizeof orderArray);

fileIO.close();

//Для проверки, состояния массивов

/\*fileIO.read((char\*)&orderArray, sizeof orderArray);

fileIO.close();\*/

system("pause");

}

else

{

system("cls");

printf("Файл не создан\n\n");

system("pause");

}

system("cls");

}

void searchNote\_NameOrders()

{

system("cls");

if (ordf != 0) //Проверка, создан ли файл

{

fstream fileIO;

fileIO.open("OrdersFile.txt", fstream::in | fstream::out | fstream::binary);

char findOrderName[50];

cin.ignore();

printf("Введите название меню просмотра состава блюд: ");

cin >> findOrderName;

fileIO.read((char\*)&orderArray, sizeof orderArray);

bool temp = false;

for (int j = 0; j < ord; j++)

{

if (string(findOrderName) == string(orderArray[j].nameOrder))

{

temp = true;

cout << "Название заказа: " << orderArray[j].nameOrder << "\n";

cout << "Дата заказа: " << orderArray[j].dateOrder.day << "." << orderArray[j].dateOrder.month << "." << orderArray[j].dateOrder.year << "\n";

cout << "Номер стола: " << orderArray[j].tableReserve.numberOfTable << "\n";

cout << "Количество мест: " << orderArray[j].tableReserve.numberOfSeats << "\n";

cout << "Название меню: " << orderArray[j].menuReserve.nameMenu << "\n";

cout << "Список блюд(номер блда, название блюда, имя повара): \n";

for (int i = 0; string(orderArray[j].menuReserve.listOfDishes[i].nameDish) != string("\0"); i++)

{

cout << i << " - " << orderArray[j].menuReserve.listOfDishes[i].nameDish << " - " << orderArray[j].menuReserve.listOfDishes[i].nameCook << "\n";

}

}

}

if (!temp)

cout << "\nЗаказа с таким названием не обнаружено. Попробуйте ввести название еще раз";

fileIO.close();

// Для проверки, состояния массивов

//fileIO.read((char\*)&orderArray, sizeof orderArray);

//fileIO.close();

system("pause");

}

else

{

system("cls");

printf("Файл не создан\n\n");

system("pause");

}

system("cls");

}