АННОТАЦИЯ

Данная пояснительная записка содержит в себе характеристику приложения, ведущего учет авиарейсов.  Поля базы данных содержат номер рейса, название рейса, марку самолета, общие расходы на рейс и количество пассажиров. С помощью приложения пользователь может осуществить создание списка, добавление элемента, вывод в виде таблицы, удаление записи, поиск, сортировка и редактирование по ключевому полю, а также подсчет итоговых данных. Данное приложение может применяться в аэропортах для учета расходов на авиарейсы.

СОДЕРЖАНИЕ

АННОТАЦИЯ 4

ВВЕДЕНИЕ 6

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ ПРОГРАММЫ 7
2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ 8
   1. Постановка задачи на разработку программы 8
   2. Метод, применяемый для индивидуального задания 8
   3. Описание и обоснование выбора метода организации входных,

выходных и промежуточных данных 8

* 1. Описание алгоритмов функционирования программы 9

1. ВЫПОЛНЕНИЕ ПРОГРАММЫ 16
   1. Условия выполнения программы 16
   2. Загрузка и запуск программы 16
   3. Проверка работоспособности программы 16

ВЫВОДЫ 23

ПЕРЕЧЕНЬ ССЫЛОК 24

ПРИЛОЖЕНИЕ А 25

ВВЕДЕНИЕ

Целью данного курсового проекта является систематизация, закрепление и углубление знаний в области основ программирования и совершенствование практических навыков разработки программ на языке Си и С++.

Для достижения цели на разных этапах курсового проектирования должны быть решены следующие задачи:

- выбор варианта задания и детализация поставки задачи;

- определение требований к функциям, выполняемых разрабатываемой программой;

- выбор типов и проектирование структур данных, определяющих способы представления, хранения и преобразования входных, выходных и промежуточных данных;

- разработка модульной структуры программы, определение функций модулей и способов их взаимодействия;

- написание текста (кодирование) программных модулей на алгоритмическом языке;

- разработка тестовых примеров;

- тестирование и отладка программы;

- разработка программных документов в соответствии с действующими стандартами.

На сегодняшний день данное приложение не имеет особой актуальности, так как есть более современные и удобные аналоги с большим функционалом, однако для базового подсчета использоваться может.

Курсовой проект состоит из пояснительной записки, включающей в себя аннотацию, введение, основные разделы, выводы, список литературы, приложение и программы на электронном носителе.

1 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ ПРОГРАММЫ

Программа осуществляет учет планируемых авиарейсов, для создания отчета по расходам на каждый рейс, а также на все рейс в совокупности. Все данные можно вывести в форме таблицы с такими полями: номер рейса, название рейса (имена городов, разделенные дефисом), марка самолета, расходы на рейс, количество пассажиров на самолете. Таблица содержит уникальное поле с расчетом средних расходов на одного пассажира отдельного рейса. С помощью данной программы можно осуществить базовые операции с со списком данных, который создается вручную или загружается из файла. Также данные можно сохранить в файл по умолчанию или создать свой.

Область применения ограничена, так как программа является узконаправленной и подходит для применения сотрудниками аэропортов для ведения отчетности по расходам на рейсы.

2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

* 1. Постановка задачи на разработку программы

Задача: Даны сведения о перевозках авиапассажиров на рейсах одного аэропорта. Структура записи: номер рейса, название маршрута (например, «Севастополь − Лондон», 20 символов), марка самолета (6 символов), общие стоимостные затраты на рейс, количество пассажиров. Подсчитать среднюю стоимость перевозки од­ного пассажира на рейсе, а также итоговые данные по затратам и количеству пассажиров и среднюю стоимость провозки одного пассажира по аэропорту. Реализовать создание, просмотр, удаление списка, добавление в него записи, сортировку, поиск по списку, ввод и вывод в файл.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер рейса | Название рейса | Марка самолета | Затраты на рейс | Количество пассажиров |
|  |  |  |  |  |

Таблица 1 – Структура записи

* 1. Применяемые математические методы.

Чтобы получить итоговую информацию по расходам, была применена простая формула нахождения среднего арифметического. Получаем общие затраты и количество пассажиров и делим первое на второе. Также было добавлено уникальное поле в таблицу вывода согласно индивидуальному заданию.

* 1. Описание и обоснование выбора метода организации входных, выходных и промежуточных данных.

Исходные данные хранятся в текстовом файле, содержащем сведения о рейсах данного аэропорта. Каждая запись содержит:

- номер рейса;

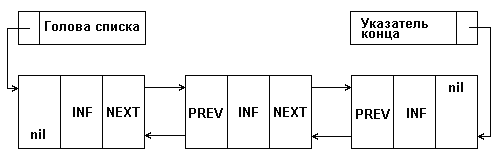
- название маршрута (20 символов);

- марка самолета (6 символов);

- общие стоимостные затраты на рейс;

- количество пассажиров;

Для данных используется динамический двунаправленный список (рис. 2.1), что упрощает доступ и работу с каждым элементом.



**NULL**

**NULL**

Рисунок 2.1 – Схема двусвязного списка

Выходные данные могут выводиться в файл или на дисплей с возможностью перелистывания. Один экран вывода содержит в себе максимум 10 записей.

2.4 Описание алгоритмов функционирования программы

В данном разделе не будут приведены структурные схемы простейших функций, таких как создание списка, добавление элемента, вывод на экран и. т. д, так как они не нуждаются в пояснении.

Схема алгоритма основной программы на представлена на рисунке 2.2.

Поблочное описание:

1. Получение из функции пункта меню
2. Проверка введенного значения пункта меню
3. Вызов функции добавления записей
4. Вызов функции просмотра записей
5. Вызов функции удаления записей
6. Вызов функции поиска
7. Вызов функции сортировки.
8. Вызов функции редактирования.
9. Вызов функции подсчета итоговых значений.
10. Вызов функции записи в файл.
11. Вызов функции чтения из файла.
12. Выход из программы.

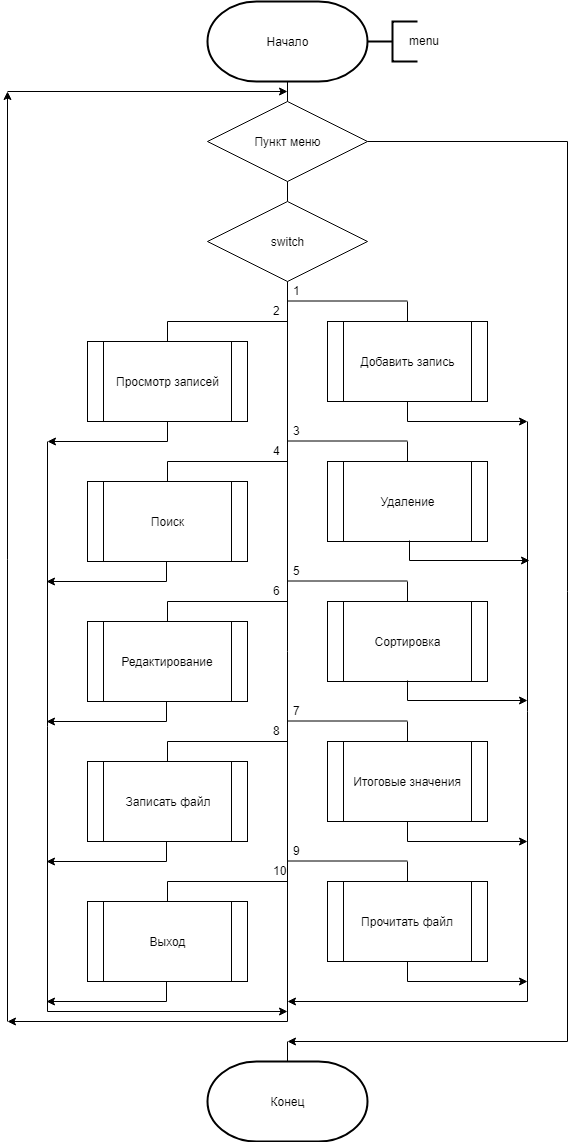


Рисунок 2.2 – Структурная схема основной программы

Далее приведена схема главного меню программы (рис. 2.3), все последующие меню работают по тому же принципу, что и главное: отображение главного меню, считывание кода клавиши и последующая передача его в основную программу.

Поблочное описание:

1. Вызов функции меню
2. Проверка нажатия клавиши Вверх
3. Проверка значения point после декрементации
4. Переход выделения вниз
5. Проверка нажатия клавиши Вниз
6. Проверка значения point после инкрементации
7. Переход выделения Верх
8. Условия цикла Do-While

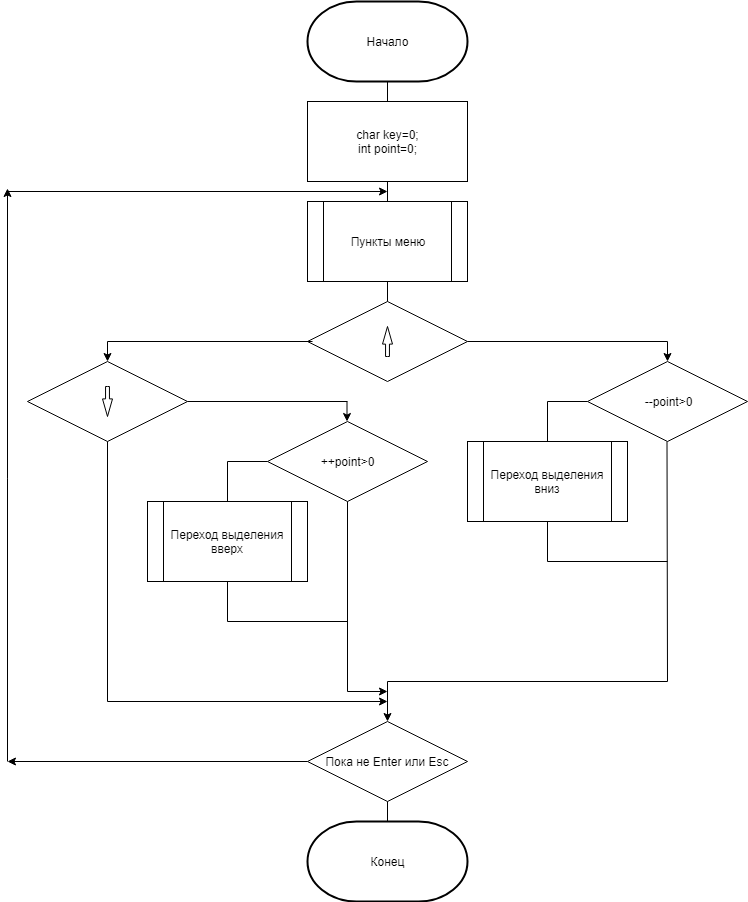


Рисунок 2.3 – Структурная схема функции Menu()

Функция, реализующая индивидуальное задание (рис. 2.4), в качестве параметров получает указатель на начало списка и используя формулу поиска среднего арифметического находит общие средние затраты.

Поблочное описание:

1. Присвоение временной структуре значений начала списка
2. Цикл до конца списка
3. Нахождение общих значений расходов и пассажиров
4. Нахождение среднего значения
5. Вывод в виде таблицы

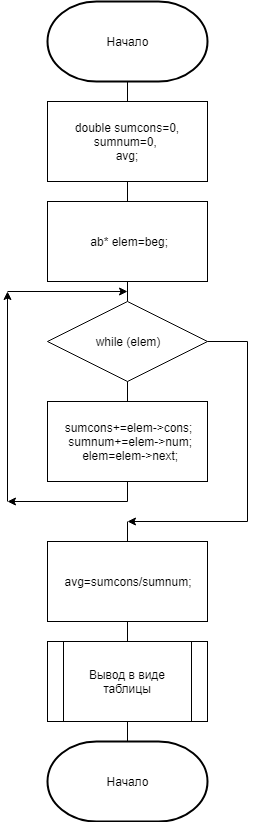


Рисунок 2.4 – Структурная схема функции Total()

Функция DeleteOne(ab \*beg, int n) (рис.2.5) выполняет удаление одного элемента списка по выбранному номеру рейса. В качестве параметров получает указатель на начало списка и номер необходимой записи.

Поблочное описание:

1. Цикл до конца списка
2. Проверка записи на совпадение запросу
3. Проверка наличия записи
4. Замещение удаляемой записи
5. Освобождение временной записи

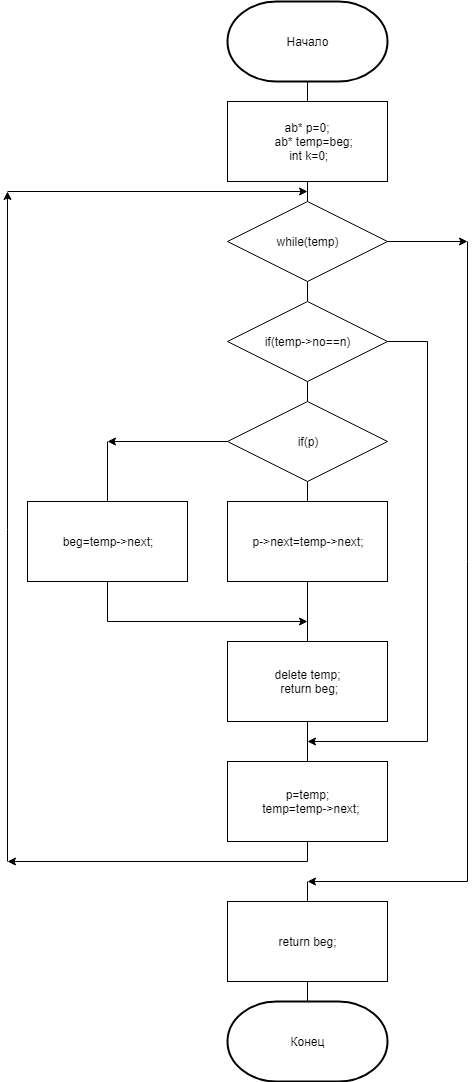


Рисунок 2.5 – Структурная схема функции DeleteOne(ab \*beg, int n)

С помощью функций Sort() осуществляется сортировка данных списка по одному из полей (рис. 2.6). В программе имеется несколько функций сортировки, работающих по одному принципу и отличающиеся лишь ключевым полем.

Поблочное описание:

1. Цикл до предпоследней записи
2. Сравнение двух записей
3. Перемещение записей местами
4. Следующая итерация

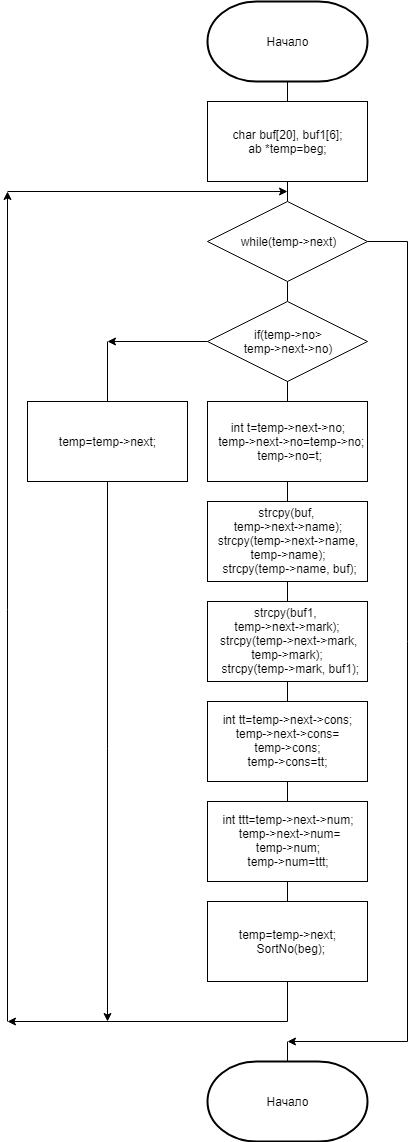


Рисунок 2.6 – Структурная схема функции Sort()

В результате компиляции программы получается файл с расширением .exe и размером 966 КБ (989 290 байт). Также в ходе выполнения программы могут быть созданы текстовые файлы неопределенного размера.

Программа потребляет в среднем 2,1 Мб оперативной памяти. Поэтому она будет нормально функционировать на любом устройстве под управлением OC Windows.

3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

* 1. Условия выполнения программы

Минимальные системные требования:

- Операционная система: Windows 10

- Оперативная память: 32 Мб

- Свободное место на жестком диске: 10 Мб

* 1. Загрузка и запуск программы

Запуск программы производится путем открытия файла Avia.exe. Сразу после запуска открывается окно с выбором читаемого файла (рис. 3.1).

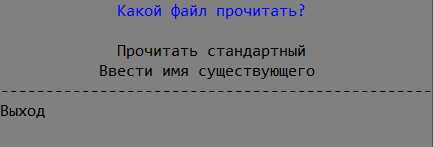


Рисунок 3.1 – Открытие файла

* 1. Проверка работоспособности программы

После выбора читаемого файла, открывается главное меню. Управление осуществляется с помощью стрелок Вверх и Вниз. При выборе пункта меню, его надпись подсвечивается зеленым цветом и увеличивается в размерах. При переключении с последнего элемента вниз будет выбран первый пункт меню, аналогично при переключении с первого элемента вверх будет выбран последний пункт (рис. 3.1 и 3.2). Выход из программы осуществляется с помощью пункта меню “Выход” или клавиши Esc. Если нажать любую другую клавишу, откроется пояснение к пользованию данным меню (рис. 3.3).

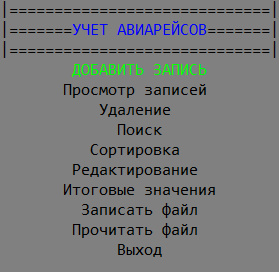


Рисунок 3.1 – Верхняя позиция меню

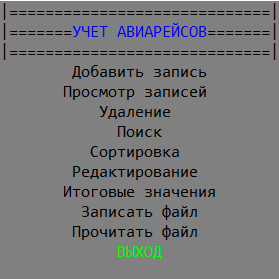


Рисунок 3.2 – Нижняя позиция меню

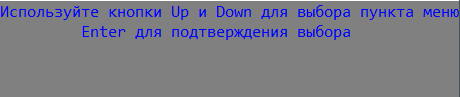


Рисунок 3.3 – Пояснение к главному меню

С помощью пункта меню “Добавить запись” можно добавить новый элемент базы данных (рис. 3.4). Были предусмотрены защиты от введения некорректных данных.

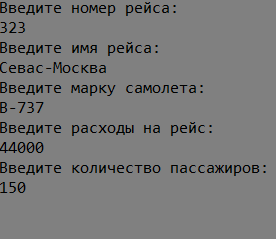


Рисунок 3.4 – Меню добавления

После ввода элемента возможно сразу добавить еще (Рисунок 3.5), а также реализована возможность выхода в главное меню.

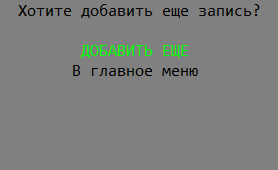


Рисунок 3.5 – Выбор дальнейших действий после ввода записи

Просмотреть все записи можно с помощью пункта меню “Просмотр записей”. Вывод реализован в форме таблицы с необходимыми полями, а также добавлено уникальное поле “Расходы на 1 пассажира”, где высчитывается расходы на одного пассажира данного рейса. Реализована возможность пролистывания экранов вывода с помощью стрелок Влево и Вправо (Рисунки 3.6 и 3.7).

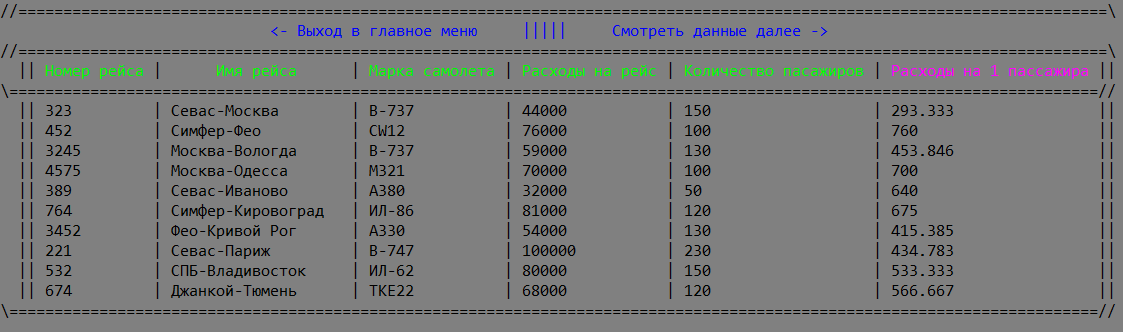


Рисунок 3.6 – Начало списка

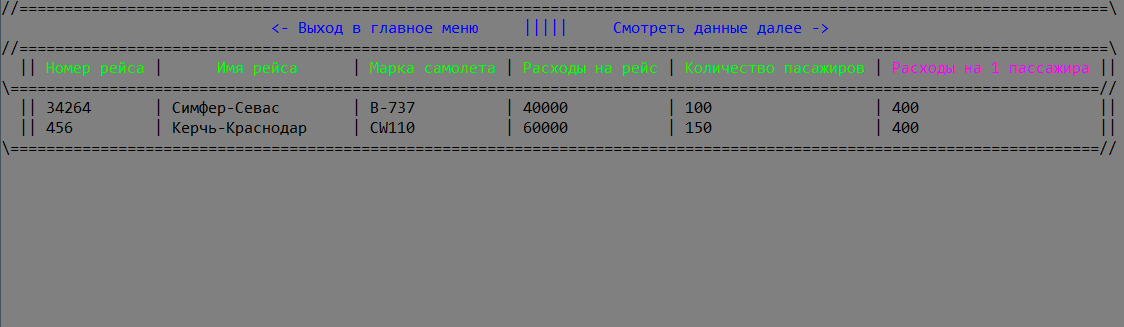


Рисунок 3.7 – Конец списка

При выборе пункта “Удаление” открывается меню выбора (рис. 3.8)..

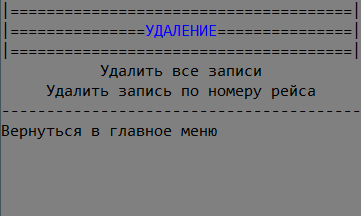


Рисунок 3.8 – Меню удаления

На Рисунках 3.9 и 3.10 показаны результаты удаления по номеру.

C:\Users\feo66\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Удал по ном.png

Рисунок 3.9 – Успешное удаление по номеру

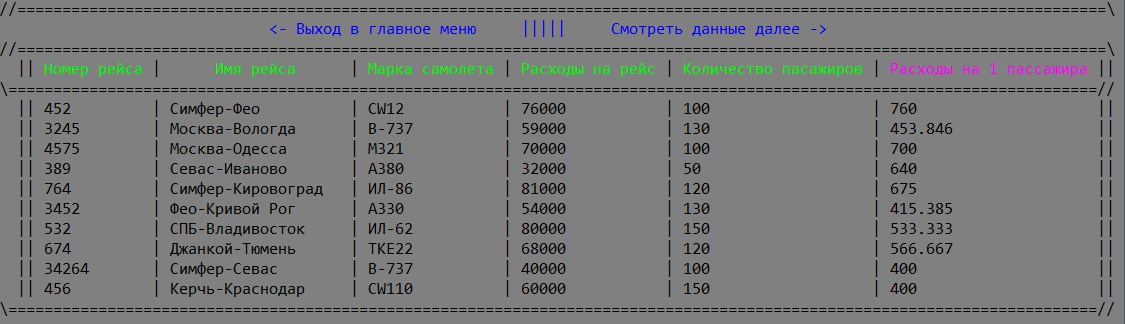


Рисунок 3.10 – Результаты удаления по номеру

Также была реализована возможность поиска данных по ключевому полю, для которой было написано отдельное меню (рис.3.11). Найденные записи выводятся в отдельном окне (рис. 3.12).

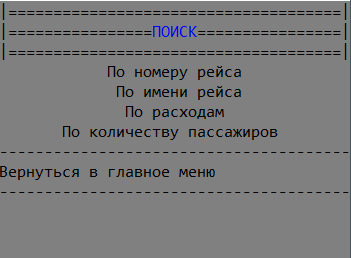


Рисунок 3.11 – Меню поиска

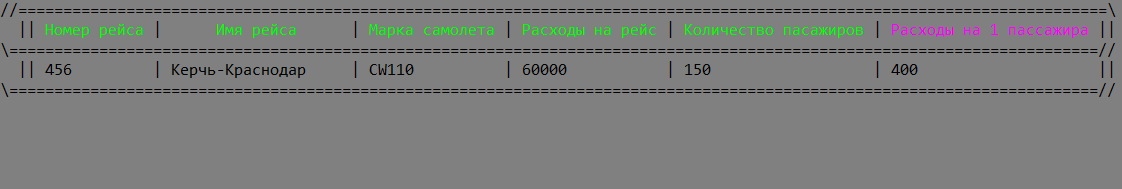


Рисунок 3.12 – Результат поиска по номеру

Имеется возможность отсортировать список по ключевому полю (рис. 3.13). Меню данной функции аналогично меню поиска (см. рис. 3.11).

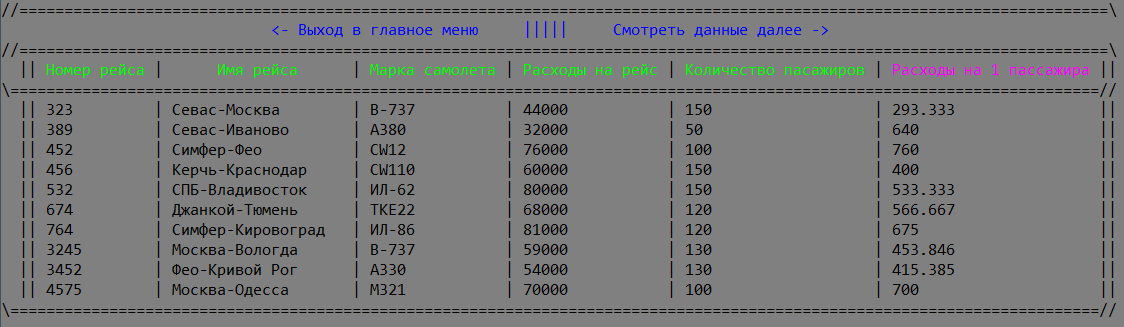


Рисунок 3.13 – Результаты сортировки по номеру

Также в программе реализована возможность редактирования записи по номеру. Причем возможно редактирование как всей записи, так и отдельного поля выбранной записи. Меню данной функции аналогично меню поиска и сортировки (см. рис. 3.11), однако добавлен пункт “Редактирование марки самолета”. Процесс и результат редактирования представлены на Рисунках 3.14 и 3.15.

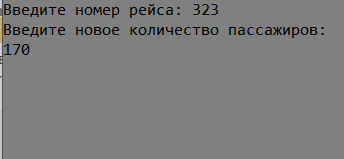


Рисунок 3.14 – Редактирование количества пассажиров

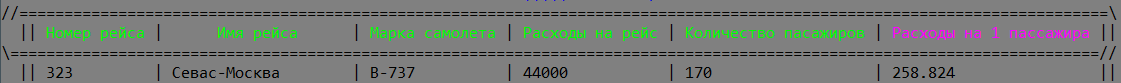


Рисунок 3.15 – Результат редактирования

Индивидуальное задание было реализовано в виде функции, подсчитывающей итоговые значения расходов (рис. 3.16).

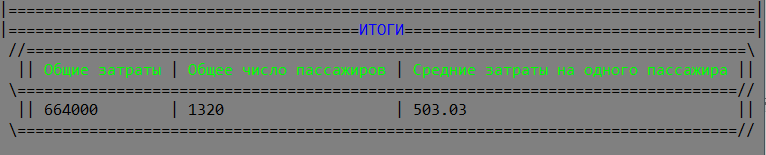


Рисунок 3.16 – Итоговые значения

Все данные можно сохранить в уже открытый в начале программы файл, либо создать новый файл и сохранить в него (рис. 3.17). Результат сохранения представлен на Рисунке 3.18.

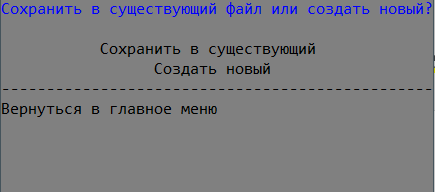


Рисунок 3.17 – Меню сохранения в файл

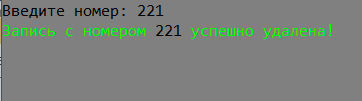


Рисунок 3.18 – Результат сохранения в файл

ВЫВОДЫ

Целью курсового проекта являлась систематизация, закрепление и углубление знаний в области программирования и совершенствование практических навыков разработки программ на языке С/С++. Этим задачам было уделено огромное внимание во время написания программы.

В ходе выполнения курсового проекта была написана программа, ведущая учет авиарейсов и подсчет итоговых значений по расходам. В качестве алгоритма хранения данных была выбрана структура в виде двунаправленного списка. Этот подход к организации данных помог упростить доступ к элементам списка. В программе были реализованы все необходимые функции для работы со списком, такие как создание, редактирование, удаление, добавление записи, поиск по списку, сортировка, запись и чтения файла. Были предусмотрены защиты от ввода некорректных данных.

Таким образом, цель курсового проекта была достигнута. Если улучшить программу, написанную в ходе курсового проекта, то ее использование возможно для учета расходов по авиарейсам. Знания, полученные при разработке курсового проекта, оказались полезными и будут применены для последующего обучения.

ПЕРЕЧЕНЬ ССЫЛОК

1. Операторы в C и C++ [Электронный ресурс]: Материал из Википедии — свободной энциклопедии. — Электрон. дан. — Сан-Франциско: Фонд Викимедиа, 2018. — URL: https://ru.wikipedia.org/?oldid=96144719.
2. Павловская Т.А. C/C++. Программирование на языке высокого уровня. / Т.А. Павловская. – СПб.: Питер, 2003. – 461 с.
3. Страуструп, Б. Язык программирования C++: Специальное издание / Б. Страуструп; Пер. с англ. Н.Н. Мартынов. — М.: БИНОМ, 2012. — 1136 c.
4. Головин, И.Г. Языки и методы программирования: Учебник для студентов учреждений высшего профессионального образования / И.Г. Головин, И.А. Волкова. — М.: ИЦ Академия, 2012. — 304 c.
5. Боровский, А.Н. Qt4.7+. Практическое программирование на C++. / А.Н. Боровский. – СПб.: BHV, 2012. – 496 c.

ПРИЛОЖЕНИЕ А

Текст программы

#include <fstream>

#include <cstdlib>

#include <stdlib.h>

#include <cstring>

#include <iomanip>

#include <conio.h>

#include <iostream>

#include <windows.h>

using namespace std;

struct ab

{

int no, num;

char name[20], mark[6];

float cons;

ab \*next, \*prev;

} ;

ab\* Create(const ab& z);

ab\* Add(ab \*beg, ab \*end, const ab& z);

ab\* Delete(ab\* beg);

ab\* DeleteOne(ab \*beg, int n);

ab\* Search(ab \*beg);

ab\* Edit(ab \*beg);

ab Input();

int Menu();

int DeleteMenu();

int SortMenu();

int SearchMenu();

int InputMenu();

int AddMenu();

int EditMenu();

int WritefileMenu();

int ReadfileMenu();

void Print(const ab& z);

void View(ab\* beg);

void SortNo(ab\* beg);

void SortName(ab\* beg);

void SortCons(ab\* beg);

void SortNum(ab\* beg);

void Total(ab \*beg);

int ReadFile(char\* rfile, ab\*\* beg, ab\*\* end);

int WriteFile(char\* file, ab\* elem);

void Green(char \*text);

void Red(char \*text);

void Blue(char \*text);

void NEBlue(char \*text);

void EndlGreen(char \*text);

void Violet(char \*text);

//----------Основная программа----------

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(0,"Russian");

ab \*beg=0, \*end=0;

int kk, n;

bool q=true, tob;

char \*file="123.txt";

char rfile[15]="123.txt", wfile[15];

while (q)

{

system("mode con cols=48 lines=9");

switch (ReadfileMenu())

{

case 1: ReadFile(file, &beg, &end);

q=false;

break;

case 2: cout<<"Введите название файла:"<<endl;

while (!(cin.getline(rfile, 15)))

{

Red("Введите корректное имя файла");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.clear();

ReadFile(rfile, &beg, &end);

q=false;

break;

case 3: return 0;

default: system("mode con cols=51 lines=13");

Blue("Используйте кнопки Up и Down для выбора пункта меню");

cout<<setw(39);

Blue("Enter для подтверждения выбора");

getch();

break;

}

system("color 80");

}

while (1)

{

system("mode con cols=31 lines=14");

switch (Menu())

{

case 1: system("mode con cols=31 lines=13");

system("cls");

if (beg)

{

end=Add(beg, end, Input());

int k=0;

while(k==0)

switch(AddMenu())

{

case 1: end=Add(beg, end, Input());

break;

case 2: k++;

break;

}

}

else

{

beg=Create(Input());

end=beg;

int k=0;

while(k==0)

switch(AddMenu())

{

case 1: end=Add(beg, end, Input());

break;

case 2: k++;

break;

}

}

break;

case 2: system("mode con cols=125 lines=17");

system("cls");

View(beg);

break;

case 3: if(beg)

{

system("mode con cols=40 lines=13");

switch (DeleteMenu())

{

case 1: beg=Delete(beg);

cout<<setw(34);

Green("Все данные успешно удалены!");

getch();

break;

case 2: {

ab\* temp=beg;

bool g=false;

int k=0;

cout<<"Введите номер: ";

while (!(cin>>n && n>0))

{

Red("Введите корректное значение");

break;

}

while(temp)

{

if(temp->no==n) g=true;

temp=temp->next;

}

if (g)

{

beg=DeleteOne(beg, n);

Green("Запись с номером ");

cout<<n;

Green(" успешно удалена!");

}

else

{

Red("Ошибка: такой записи нет!");

getch();

break;

}

}

getch();

break;

case 3: break;

}

}

else

{

cout<<setw(25)<<endl;

Red("Ошибка: данных нет!");

getch();

}

break;

case 4: if(beg)

{

system("mode con cols=39 lines=13");

Search(beg);

break;

}

else

{

cout<<setw(24)<<endl;

Red("Ошибка: данных нет!");

getch();

}

break;

case 5: if(beg)

{

system("mode con cols=40 lines=13");

switch (SortMenu())

{

case 1: SortNo(beg);

cout<<setw(34);

Green("Данные успешно отсортированы!");

getch();

break;

case 2: SortName(beg);

cout<<setw(34);

Green("Данные успешно отсортированы!");

getch();

break;

case 3: SortCons(beg);

cout<<setw(34);

Green("Данные успешно отсортированы!");

getch();

break;

case 4: SortNum(beg);

cout<<setw(34);

Green("Данные успешно отсортированы!");

getch();

break;

case 5: break;

}

}

else

{

cout<<setw(25)<<endl;

Red("Ошибка: данных нет!");

getch();

}

break;

case 6: if(beg)

{

system("mode con cols=38 lines=14");

Edit(beg);

}

else

{

cout<<setw(25)<<endl;

Red("Ошибка: данных нет!");

getch();

}

break;

case 7: system("mode con cols=85 lines=9");

if(beg)

Total(beg);

else

{

cout<<setw(50)<<endl;

Red("Ошибка: данных нет!");

getch();

}

break;

case 8: system("mode con cols=48 lines=13");

switch (WritefileMenu())

{

case 1:

{

WriteFile(rfile, beg);

break;

}

case 2:

{

cout<<"Введите название нового файла:"<<endl;

while (!(cin.getline(wfile, 15)))

{

Red("Введите корректное имя файла");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.clear();

WriteFile(wfile, beg);

break;

}

case 3: break;

}

break;

case 9: system("mode con cols=48 lines=9");

switch (ReadfileMenu())

{

case 1: ReadFile(file, &beg, &end);

break;

case 2: cout<<"Введите название файла:"<<endl;

while (!(cin.getline(rfile, 15)))

{

Red("Введите корректное имя файла");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.clear();

ReadFile(rfile, &beg, &end);

break;

case 3: return 0;

}

break;

case 10: system("mode con cols=31 lines=13");

return 0;

default: system("mode con cols=51 lines=13");

Blue("Используйте кнопки Up и Down для выбора пункта меню");

cout<<setw(39);

Blue("Enter для подтверждения выбора");

getch();

break;

}

}

}

//----------------Функция создания списка-------------------

ab\* Create(const ab& z)

{

ab \*beg=new ab;

\*beg=z;

beg->next=0;

beg->prev=0;

return beg;

}

//----------------Функция добавления элемента-------------------

ab\* Add(ab \*beg, ab \*end, const ab& z)

{

ab \*elem=new ab;

ab \*temp=beg;

\*elem=z;

while(temp)

{

if(temp->no==z.no)

{

system("cls");

cout<<setw(29)<<endl;

Red("Данный рейс уже существует!");

getch();

return temp;

}

temp=temp->next;

}

elem->next=0;

elem->prev=beg;

end->next=elem;

end=elem;

return end;

}

//----------------Функция удаления всего списка-------------------

ab\* Delete(ab \*beg)

{

ab \*elem=beg;

if (!beg)

{

Red("Ошибка: данных нет!");

return 0;

}

while(elem->next!=0)

{

ab\* temp;

temp=elem;

elem=elem->next;

delete temp;

}

elem=elem->next;

delete elem;

beg=NULL;

return beg;

}

//----------------Функция удаления одной записи-------------------

ab\* DeleteOne(ab \*beg, int n)

{

ab\* p=0;

ab\* temp=beg;

int k=0;

while(temp)

{

if(temp->no==n)

{

k++;

if(p) p->next=temp->next;

else beg=temp->next;

delete temp;

return beg;

}

p=temp;

temp=temp->next;

}

return beg;

}

//----------------Функция ввода записи-------------------

ab Input()

{

ab z;

cout<<"Введите номер рейса:"<<endl;

while (!(cin>>z.no && z.no>0))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

cout<<"Введите имя рейса:"<<endl;

while (!(cin.getline(z.name, 20)))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите марку самолета:"<<endl;

while (!(cin.getline(z.mark, 6)))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите расходы на рейс:"<<endl;

while (!(cin>>z.cons && z.cons>0) || z.cons>1000000 || z.cons<0)

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите количество пассажиров:"<<endl;

while (!(cin>>z.num && z.num>0))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

return z;

}

//----------------Функция основного меню-------------------

int Menu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

cout<<"|=============================|"<<endl;

cout<<"|=======";

NEBlue("УЧЕТ АВИАРЕЙСОВ");

cout<<"=======|"<<endl;

cout<<"|=============================|"<<endl;

if (key==72)

{

if (--point<0) point=10;

}

else if (key==80)

{

if (++point>10) point=0;

}

if(point==1)

{

cout<<setw(23);

EndlGreen("ДОБАВИТЬ ЗАПИСЬ");

}

else cout<<setw(23)<<"Добавить запись"<<endl;

if(point==2)

{

cout<<setw(23);

EndlGreen("ПРОСМОТР ЗАПИСЕЙ");

}

else cout<<setw(23)<<"Просмотр записей"<<endl;

if(point==3)

{

cout<<setw(19);

EndlGreen("УДАЛЕНИЕ");

}

else cout<<setw(19)<<"Удаление"<<endl;

if(point==4)

{

cout<<setw(18);

EndlGreen("ПОИСК");

}

else cout<<setw(18)<<"Поиск"<<endl;

if(point==5)

{

cout<<setw(20);

EndlGreen("СОРТИРОВКА");

}

else cout<<setw(20)<<"Сортировка"<<endl;

if(point==6)

{

cout<<setw(22);

EndlGreen("РЕДАКТИРОВАНИЕ");

}

else cout<<setw(22)<<"Редактирование"<<endl;

if(point==7)

{

cout<<setw(24);

EndlGreen("ИТОГОВЫЕ ЗНАЧЕНИЯ");

}

else cout<<setw(24)<<"Итоговые значения"<<endl;

if(point==8)

{

cout<<setw(22);

EndlGreen("ЗАПИСАТЬ ФАЙЛ");

}

else cout<<setw(22)<<"Записать файл"<<endl;

if(point==9)

{

cout<<setw(22);

EndlGreen("ПРОЧИТАТЬ ФАЙЛ");

}

else cout<<setw(22)<<"Прочитать файл"<<endl;

if(point==10)

{

cout<<setw(18);

EndlGreen("ВЫХОД");

}

else cout<<setw(18)<<"Выход"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 10;

}

//----------------Функция меню удаления-------------------

int DeleteMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

if (key==72)

{

if (--point<0) point=3;

}

else if (key==80)

{

if (++point>3) point=0;

}

cout<<"|======================================|"<<endl;

cout<<"|===============";

NEBlue("УДАЛЕНИЕ");

cout<<"===============|"<<endl;

cout<<"|======================================|"<<endl;

if(point==1)

{

cout<<setw(29);

EndlGreen("УДАЛИТЬ ВСЕ ЗАПИСИ");

}

else cout<<setw(29)<<"Удалить все записи"<<endl;

if(point==2)

{

cout<<setw(35);

EndlGreen("УДАЛИТЬ ЗАПИСЬ ПО НОМЕРУ РЕЙСА");

}

else cout<<setw(35)<<"Удалить запись по номеру рейса"<<endl;

cout<<"----------------------------------------"<<endl;

if(point==3) EndlGreen("ВЕРНУТЬСЯ В ГЛАВНОЕ МЕНЮ");

else cout<<"Вернуться в главное меню"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 3;

}

//----------------Функция меню сортировки-------------------

int SortMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

if (key==72)

{

if (--point<0) point=5;

}

else if (key==80)

{

if (++point>5) point=0;

}

cout<<"|======================================|"<<endl;

cout<<"|==============";

NEBlue("СОРТИРОВКА");

cout<<"==============|"<<endl;

cout<<"|======================================|"<<endl;

if(point==1)

{

cout<<setw(28);

EndlGreen("ПО НОМЕРУ РЕЙСА");

}

else cout<<setw(28)<<"По номеру рейса"<<endl;

if(point==2)

{

cout<<setw(28);

EndlGreen("ПО ИМЕНИ РЕЙСА");

}

else cout<<setw(28)<<"По имени рейса"<<endl;

if(point==3)

{

cout<<setw(26);

EndlGreen("ПО РАСХОДАМ");

}

else cout<<setw(26)<<"По расходам"<<endl;

if(point==4)

{

cout<<setw(32);

EndlGreen("ПО КОЛИЧЕСТВУ ПАССАЖИРОВ");

}

else cout<<setw(32)<<"По количеству пассажиров"<<endl;

cout<<"----------------------------------------"<<endl;

if(point==5) EndlGreen("ВЕРНУТЬСЯ В ГЛАВНОЕ МЕНЮ");

else cout<<"Вернуться в главное меню"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 5;

}

//----------------Функция меню поиска-------------------

int SearchMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

if (key==72)

{

if (--point<0) point=5;

}

else if (key==80)

{

if (++point>5) point=0;

}

cout<<"|=====================================|"<<endl;

cout<<"|================";

NEBlue("ПОИСК");

cout<<"================|"<<endl;

cout<<"|=====================================|"<<endl;

if(point==1)

{

cout<<setw(27);

EndlGreen("ПО НОМЕРУ РЕЙСА");

}

else cout<<setw(27)<<"По номеру рейса"<<endl;

if(point==2)

{

cout<<setw(27);

EndlGreen("ПО ИМЕНИ РЕЙСА");

}

else cout<<setw(27)<<"По имени рейса"<<endl;

if(point==3)

{

cout<<setw(25);

EndlGreen("ПО РАСХОДАМ");

}

else cout<<setw(25)<<"По расходам"<<endl;

if(point==4)

{

cout<<setw(31);

EndlGreen("ПО КОЛИЧЕСТВУ ПАССАЖИРОВ");

}

else cout<<setw(31)<<"По количеству пассажиров"<<endl;

cout<<"---------------------------------------"<<endl;

if(point==5) EndlGreen("ВЕРНУТЬСЯ В ГЛАВНОЕ МЕНЮ");

else cout<<"Вернуться в главное меню"<<endl;

cout<<"---------------------------------------"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 5;

}

//----------------Функция меню добавления-------------------

int AddMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

cout<<setw(29)<<"Хотите добавить еще запись?"<<endl<<endl;

if (key==72)

{

if (--point<0) point=2;

}

else if (key==80)

{

if (++point>2) point=0;

}

if(point==1)

{

cout<<setw(21);

EndlGreen("ДОБАВИТЬ ЕЩЕ");

}

else cout<<setw(21)<<"Добавить еще"<<endl;

if(point==2)

{

cout<<setw(22);

EndlGreen("В ГЛАВНОЕ МЕНЮ");

}

else cout<<setw(22)<<"В главное меню"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 2;

}

//----------------Функция меню записи в файл-------------------

int WritefileMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

Blue("Сохранить в существующий файл или создать новый?");

cout<<endl;

if (key==72)

{

if (--point<0) point=3;

}

else if (key==80)

{

if (++point>3) point=0;

}

if(point==1)

{

cout<<setw(35);

EndlGreen("СОХРАНИТЬ В СУЩЕСТВУЮЩИЙ");

}

else cout<<setw(35)<<"Сохранить в существующий"<<endl;

if(point==2)

{

cout<<setw(30);

EndlGreen("СОЗДАТЬ НОВЫЙ");

}

else cout<<setw(30)<<"Создать новый"<<endl;

cout<<"------------------------------------------------"<<endl;

if(point==3) EndlGreen("ВЕРНУТЬСЯ В ГЛАВНОЕ МЕНЮ");

else cout<<"Вернуться в главное меню"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 3;

}

//----------------Функция меню чтения файла-------------------

int ReadfileMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

cout<<setw(34);

Blue("Какой файл прочитать?");

cout<<endl;

if (key==72)

{

if (--point<0) point=3;

}

else if (key==80)

{

if (++point>3) point=0;

}

if(point==1)

{

cout<<setw(34);

EndlGreen("ПРОЧИТАТЬ СТАНДАРТНЫЙ");

}

else cout<<setw(34)<<"Прочитать стандартный"<<endl;

if(point==2)

{

cout<<setw(35);

EndlGreen("ВВЕСТИ ИМЯ СУЩЕСТВУЮЩЕГО");

}

else cout<<setw(35)<<"Ввести имя существующего"<<endl;

cout<<"------------------------------------------------"<<endl;

if(point==3) EndlGreen("ВЫХОД");

else cout<<"Выход"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 3;

}

//----------------Функция меню редактирования-------------------

int EditMenu()

{

char key=0;

int point=0;

do

{

system("color 80");

system("cls");

if (key==72)

{

if (--point<0) point=7;

}

else if (key==80)

{

if (++point>7) point=0;

}

cout<<"|====================================|"<<endl;

cout<<"|===========";

NEBlue("РЕДАКТИРОВАНИЕ");

cout<<"===========|"<<endl;

cout<<"|====================================|"<<endl;

if(point==1)

{

cout<<setw(24);

EndlGreen("ВСЮ ЗАПИСЬ");

}

else cout<<setw(24)<<"Всю запись"<<endl;

if(point==2)

{

cout<<setw(24);

EndlGreen("НОМЕР РЕЙСА");

}

else cout<<setw(24)<<"Номер рейса"<<endl;

if(point==3)

{

cout<<setw(23);

EndlGreen("ИМЯ РЕЙСА");

}

else cout<<setw(23)<<"Имя рейса"<<endl;

if(point==4)

{

cout<<setw(26);

EndlGreen("МАРКУ САМОЛЕТА");

}

else cout<<setw(26)<<"Марку самолета"<<endl;

if(point==5)

{

cout<<setw(22);

EndlGreen("РАСХОДЫ");

}

else cout<<setw(22)<<"Расходы"<<endl;

if(point==6)

{

cout<<setw(29);

EndlGreen("КОЛИЧЕСТВО ПАССАЖИРОВ");

}

else cout<<setw(29)<<"Количество пассажиров"<<endl;

cout<<"--------------------------------------"<<endl;

if(point==7) EndlGreen("ВЕРНУТЬСЯ В ГЛАВНОЕ МЕНЮ");

else cout<<"Вернуться в главное меню"<<endl;

key=getch();

}

while (key!=13 && key!=27);

system ("cls");

return key==13 ? point : 7;

}

//----------------Функция печати списка-------------------

void Print(const ab& z)

{

cout<<" || "<<setw(12)<<left<<z.no;

cout<<"| "<<setw(20)<<left<<z.name;

cout<<"| "<<setw(15)<<left<<z.mark;

cout<<"| "<<setw(16)<<left<<z.cons;

cout<<"| "<<setw(21)<<left<<z.num;

cout<<"| "<<setw(23)<<left<<(z.cons/z.num)<<"|| ";

cout<<right;

}

//----------------Функция просмотра списка-------------------

void View(ab \*beg)

{

int i=1; char key = 77; int gg=0;

if (!beg)

{

cout<<setw(70)<<endl;

Red("Ошибка: данных нет!");

getch();

return;

}

ab \*elem=beg;

while(elem)

{

if (key==77)

{

cout<<"//=========================================================================================================================\\"<<endl;

cout<<setw(92);

Blue("<- Выход в главное меню ||||| Смотреть данные далее ->");

cout<<"//=========================================================================================================================\\"<<endl;

cout<<" || ";

Green("Номер рейса");

cout<<" | ";

Green("Имя рейса");

cout<<" | ";

Green("Марка самолета");

cout<<" | ";

Green("Расходы на рейс");

cout<<" | ";

Green("Количество пасажиров");

cout<<" | ";

Violet("Расходы на 1 пассажира");

cout<<" ||"<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

for(gg=0;gg<10;)

{

if(elem)

{

cout<<left;

Print(\*elem);

elem=elem->next;

i++; gg++;

}

else break;

}

cout<<"\\=========================================================================================================================//"<<endl;

}

if (key==75) break;

key=getch();

system("cls");

}

}

//----------------Функция чтения файла-------------------

int ReadFile(char\* rfile, ab\*\* beg, ab\*\* end)

{

int k;

system("color 80");

ifstream fin(rfile,ios::in);

if (!fin)

{

cout<<setw(34)<<endl;

Red("Файла не существует!");

cout<<endl<<endl;

cout<<setw(47);

Red("Нажмите любую клавишу для создания этого файла");

getch();

}

ab z;

\*beg = 0;

while (fin.getline(z.name, 20))

{

fin.getline(z.mark, 6);

fin>>z.no;

fin>>z.cons;

fin>>z.num;

fin.get();

if (\*beg)

\*end=Add(\*beg, \*end, z);

else

{

\*beg=Create(z);

\*end=\*beg;

}

}

return 0;

}

//----------------Функция записи в файл-------------------

int WriteFile(char\* file, ab\* elem)

{

ofstream fout(file);

if (!fout)

{

cout<<"Ошибка: файл не существует!"<<endl;

return 1;

}

while (elem)

{

fout<<elem->name<<endl;

fout<<elem->mark<<endl;

fout<<elem->no<<endl;

fout<<elem->cons<<endl;

fout<<elem->num<<endl;

elem=elem->next;

}

Green("Данные сохранены в файле ");

cout<<file<<endl;

cout<<"=============================="<<endl;

cout<<"Нажмите любую клавишу"<<endl;

getch();

return 0;

}

//----------------Функция сортровки по номеру-------------------

void SortNo(ab\* beg)

{

char buf[20], buf1[6];

ab \*temp=beg;

while(temp->next)

{

if(temp->no>temp->next->no)

{

int t=temp->next->no;

temp->next->no=temp->no;

temp->no=t;

strcpy(buf, temp->next->name);

strcpy(temp->next->name, temp->name);

strcpy(temp->name, buf);

strcpy(buf1, temp->next->mark);

strcpy(temp->next->mark, temp->mark);

strcpy(temp->mark, buf1);

int tt=temp->next->cons;

temp->next->cons=temp->cons;

temp->cons=tt;

int ttt=temp->next->num;

temp->next->num=temp->num;

temp->num=ttt;

temp=temp->next;

SortNo(beg);

}

else temp=temp->next;

}

}

//----------------Функция сортровки по имени-------------------

void SortName(ab\* beg)

{

char buf[20], buf1[6];

ab \*temp=beg;

while(temp->next)

{

if(strcmp(temp->name, temp->next->name)>0)

{

int t=temp->next->no;

temp->next->no=temp->no;

temp->no=t;

strcpy(buf, temp->next->name);

strcpy(temp->next->name, temp->name);

strcpy(temp->name, buf);

strcpy(buf1, temp->next->mark);

strcpy(temp->next->mark, temp->mark);

strcpy(temp->mark, buf1);

int tt=temp->next->cons;

temp->next->cons=temp->cons;

temp->cons=tt;

int ttt=temp->next->num;

temp->next->num=temp->num;

temp->num=ttt;

temp=temp->next;

SortName(beg);

}

else temp=temp->next;

}

}

//----------------Функция сортровки по расходам-------------------

void SortCons(ab\* beg)

{

char buf[20], buf1[6];

ab \*temp=beg;

while(temp->next)

{

if(temp->cons>temp->next->cons)

{

int t=temp->next->no;

temp->next->no=temp->no;

temp->no=t;

strcpy(buf, temp->next->name);

strcpy(temp->next->name, temp->name);

strcpy(temp->name, buf);

strcpy(buf1, temp->next->mark);

strcpy(temp->next->mark, temp->mark);

strcpy(temp->mark, buf1);

int tt=temp->next->cons;

temp->next->cons=temp->cons;

temp->cons=tt;

int ttt=temp->next->num;

temp->next->num=temp->num;

temp->num=ttt;

temp=temp->next;

SortCons(beg);

}

else temp=temp->next;

}

}

//----------------Функция сортровки количеству пасажиров-------------------

void SortNum(ab\* beg)

{

char buf[20], buf1[6];

ab \*temp=beg;

while(temp->next)

{

if(temp->num>temp->next->num)

{

int t=temp->next->no;

temp->next->no=temp->no;

temp->no=t;

strcpy(buf, temp->next->name);

strcpy(temp->next->name, temp->name);

strcpy(temp->name, buf);

strcpy(buf1, temp->next->mark);

strcpy(temp->next->mark, temp->mark);

strcpy(temp->mark, buf1);

int tt=temp->next->cons;

temp->next->cons=temp->cons;

temp->cons=tt;

int ttt=temp->next->num;

temp->next->num=temp->num;

temp->num=ttt;

temp=temp->next;

SortNum(beg);

}

else temp=temp->next;

}

}

//----------------Функция поиска-------------------

ab\* Search(ab \*beg)

{

int no, num, key, i=0;

char name[20], mark[6];

float cons;

ab \*elem=beg;

system("cls");

switch(SearchMenu())

{

case 1: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

system("mode con cols=125 lines=18");

while(elem)

{

if(elem->no==no)

{

cout<<"//=========================================================================================================================\\"<<endl;

cout<<" || ";

Green("Номер рейса");

cout<<" | ";

Green("Имя рейса");

cout<<" | ";

Green("Марка самолета");

cout<<" | ";

Green("Расходы на рейс");

cout<<" | ";

Green("Количество пасажиров");

cout<<" | ";

Violet("Расходы на 1 пассажира");

cout<<" ||"<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

Print(\*elem);

cout<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

i++;

break;

}

elem=elem->next;

}

if(i==0) Red("Запись не найдена!");

cout<<endl;

getch();

break;

case 2: cout<<"Введите имя рейса: ";

while (!(cin.getline(name, 20)))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.clear();

system("mode con cols=125 lines=18");

while(elem)

{

if(strcmp(elem->name, name)==0)

{

cout<<"//=========================================================================================================================\\"<<endl;

cout<<" || ";

Green("Номер рейса");

cout<<" | ";

Green("Имя рейса");

cout<<" | ";

Green("Марка самолета");

cout<<" | ";

Green("Расходы на рейс");

cout<<" | ";

Green("Количество пасажиров");

cout<<" | ";

Violet("Расходы на 1 пассажира");

cout<<" ||"<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

Print(\*elem);

cout<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

i++;

}

elem=elem->next;

}

if(i==0) Red("Запись не найдена!");

cout<<endl;

getch();

break;

case 3: cout<<"Введите расходы на рейс: ";

while (!(cin>>cons && cons>0))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

system("mode con cols=125 lines=18");

while(elem)

{

if(elem->cons==cons)

{

cout<<"//=========================================================================================================================\\"<<endl;

cout<<" || ";

Green("Номер рейса");

cout<<" | ";

Green("Имя рейса");

cout<<" | ";

Green("Марка самолета");

cout<<" | ";

Green("Расходы на рейс");

cout<<" | ";

Green("Количество пасажиров");

cout<<" | ";

Violet("Расходы на 1 пассажира");

cout<<" ||"<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

Print(\*elem);

cout<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

i++;

}

elem=elem->next;

}

if(i==0) Red("Запись не найдена!");

cout<<endl;

getch();

break;

case 4: cout<<"Введите количество пассажиров: ";

while (!(cin>>num && num>0))

{

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

system("mode con cols=125 lines=18");

while(elem)

{

if(elem->num==num)

{

cout<<"//=========================================================================================================================\\"<<endl;

cout<<" || ";

Green("Номер рейса");

cout<<" | ";

Green("Имя рейса");

cout<<" | ";

Green("Марка самолета");

cout<<" | ";

Green("Расходы на рейс");

cout<<" | ";

Green("Количество пасажиров");

cout<<" | ";

Violet("Расходы на 1 пассажира");

cout<<" ||"<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

Print(\*elem);

cout<<endl;

cout<<"\\=========================================================================================================================//"<<endl;

i++;

}

elem=elem->next;

}

if(i==0) Red("Запись не найдена!");

cout<<endl;

getch();

break;

case 5: break;

}

}

//----------------Функция редактирования-------------------

ab\* Edit(ab \*beg)

{

int no, num, key, i=0;

char name[20], mark[6];

float cons;

ab \*elem=beg;

system("cls");

switch(EditMenu())

{

case 1: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новый номер рейса:"<<endl;

while (!(cin>>elem->no && elem->no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

cout<<"Введите новое имя рейса:"<<endl;

while (!(cin.getline(elem->name, 20)))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите новую марку самолета:"<<endl;

while (!(cin.getline(elem->mark, 6)))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите новые расходы на рейс:"<<endl;

while (!(cin>>elem->cons) || elem->cons>1000000 || elem->cons<0)

{

cout<<endl;

Red("Введите число");

cout<<endl;

cin.clear();

fflush(stdin);

}

cout<<"Введите новое количество пассажиров:"<<endl;

while (!(cin>>elem->num && elem->num>0))

{

cout<<endl;

Red("Введите число");

cout<<endl;

cin.clear();

fflush(stdin);

}

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 2: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новый номер рейса:"<<endl;

while (!(cin>>elem->no && elem->no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 3: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новое имя рейса:"<<endl;

while (!(cin.getline(elem->name, 20)))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 4: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

cin.ignore();

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новую марку самолета:"<<endl;

while (!(cin.getline(elem->mark, 6)))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 5: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новые расходы на рейс:"<<endl;

while (!(cin>>elem->cons) || elem->cons>1000000 || elem->cons<0)

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 6: cout<<"Введите номер рейса: ";

while (!(cin>>no && no>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i=0;

while(elem)

{

if(elem->no==no)

{

cout<<"Введите новое количество пассажиров:"<<endl;

while (!(cin>>elem->num && elem->num>0))

{

cout<<endl;

Red("Введите корректное значение");

cout<<endl;

cin.clear();

fflush(stdin);

}

i++;

break;

}

elem=elem->next;

}

if(i==0)

{

Red("Запись не найдена!");

getch();

}

else

{

system("cls");

cout<<setw(34)<<endl;

Green("Запись успешно отредактирована");

getch();

}

break;

case 7: break;

}

}

//----------------Функция получения итоговых значений----------------

void Total(ab \*beg)

{

double sumcons=0, sumnum=0, avg;

ab\* elem=beg;

while (elem)

{

sumcons+=elem->cons;

sumnum+=elem->num;

elem=elem->next;

}

avg=sumcons/sumnum;

cout<<"|===================================================================================|"<<endl;

cout<<"|=======================================";

NEBlue("ИТОГИ");

cout<<"=======================================|"<<endl;

cout<<" //================================================================================\\ "<<endl;

cout<<" || ";

Green("Общие затраты");

cout<<" | ";

Green("Общее число пассажиров");

cout<<" | ";

Green("Средние затраты на одного пассажира");

cout<<" || ";

cout<<" \\================================================================================// "<<endl;

cout<<" || "<<setw(14)<<left<<sumcons;

cout<<"| "<<setw(23)<<left<<sumnum;

cout<<"| "<<setw(34)<<left<<avg<<" || "<<endl;;

cout<<right;

cout<<" \\================================================================================// "<<endl;

getch();

}

//----------------Функции определения цветв текста-------------------

void Green(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout<<text;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}

void Red(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_INTENSITY | FOREGROUND\_RED);

cout<<text;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}

void Blue(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_INTENSITY | FOREGROUND\_BLUE);

cout<<text<<endl;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}

void NEBlue(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_INTENSITY | FOREGROUND\_BLUE);

cout<<text;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}

void EndlGreen(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_INTENSITY | FOREGROUND\_GREEN);

cout<<text<<endl;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}

void Violet(char \*text)

{

HANDLE hConsole=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY | FOREGROUND\_INTENSITY | FOREGROUND\_BLUE | FOREGROUND\_RED);

cout<<text;

SetConsoleTextAttribute(hConsole, BACKGROUND\_INTENSITY);

}