**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 34-36**

**«Программирование задач с использованием структур в функциях, работа с файлами и структурами»**

Специальность 09.02.03 «Программирование в компьютерных системах»

Дисциплина «Основы программирования»

|  |  |
| --- | --- |
| Преподаватель:  Данилина Татьяна Викторовна  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_201\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Санкт-Петербург  2017/2018 | Выполнил:  студент группы Y2235  Матвеенко Дмитрий Владимирович |

СОДЕРЖАНИЕ

Введение……………………………………………………………………….3

1. Постановка задачи…………………………………………………………….4

1.1. Формулировка задания.……………………………………………..…4

1.2. Исходные данные. Ограничения на исходные данные…………...…4

1.3. Выходные данные и формы…………………………………..……….5

2. Функциональные требования к программе…………………………………6

3. Схема алгоритма и её описание…………………………………..………….7

4. Выбор программного обеспечения…………………………………..………9

5. Разработка программного кода…………..…………………………..……..10

6. Отладка и тестирование программы………………………………………..11

6.1. Контрольный пример.…………………………………………….......11

7. Анализ результатов…………..………………………...……………..……..16

8. Приложение 1…………..……………………………………………..……..17

ВВЕДЕНИЕ

Целью работой является овладение навыками алгоритмизации и программирования задач с использованием структур и возможностью их передачи как параметров в функцию, приобретение практических навыков в проектировании структуры файла, а также закреплении навыков по вводу данных в файл и их обработке с помощью подпрограмм пользователя.

На сегодняшний день ИТ широко распространены в мире. Широко используются базы данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами моделирования данных.

Базы данных по своему назначению схожа с информационной системой — системой, предназначенной для хранения, поиска и обработки информации, и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию.  
 Информационная система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации для достижения цели управления. В современных условиях основным техническим средством обработки информации является персональный компьютер. Большинство современных информационных систем преобразуют не информацию, а данные. Поэтому часто их называют системами обработки данных.

Важнейшими принципами построения эффективных информационных систем являются следующие:

1. Принцип интеграции, заключающийся в том, что обрабатываемые данные, однажды введенные в систему, многократно используются для решения большого числа задач.
2. Принцип системности, заключающийся в обработке данных в различных аспектах, чтобы получить информацию, необходимую для принятия решений на всех уровнях управления.
3. Принцип комплексности, заключающийся в механизации и автоматизации процедур преобразования данных на всех этапах функционирования информационной системы.

Информационные системы также классифицируются:

1. По функциональному назначению: производственные, коммерческие, финансовые, маркетинговые и др.;
2. По объектам управления: информационные системы автоматизированного проектирования, управления технологическими процессами, управления предприятием (офисом, фирмой, корпорацией, организацией) и т. п.;
3. По характеру использования результатной информации: информационно-поисковые, предназначенные для сбора, хранения и выдачи информации по запросу пользователя; информационно-советующие, предлагающие пользователю определенные рекомендации для принятия решений (системы поддержки принятия решений); информационно-управляющие, результатная информация которых непосредственно участвует в формировании управляющих воздействий.
4. ПОСТАНОВКА ЗАДАЧИ
   1. Формулировка задания

Целью работы является овладение навыками алгоритмизации и программирования задач с использованием структур и возможностью их передачи как параметров в функцию. Приобрести практические навыки в проектировании структуры файла, а также закрепить навыки по вводу данных в файл и их обработке с помощью подпрограмм пользователя.

Задание: составить ваучер туров по странам Европы из Санкт-Петербурга, включающий следующую информацию о каждом туре: название страны, вид перевозки, стоимость, вид отеля для проживания, длительность тура в днях, цель поездки.

Составить программу, выдающую следующую справочную информацию:

Список туров по названию введенной страны;

Список туров, имеющих отель для проживания категории три.

* 1. Исходные данные. Ограничения на исходные данные

Согласно формулировке задания, были сформированы входные данные с необходимыми типами, которые указаны ниже (Таблица 1).

Ограничения были выставлены по аналогии с сайтом [tutu.ru](https://www.tutu.ru/):

1. Самое большое название страны ограничено 59 символами.
2. Вид перевозки бывает наземным и воздушным.
3. Стоимость ограничена двумя знаками после запятой.
4. Оценка отелей идет от 1 до 5 звезд.
5. Длительность тура ограничена 1 годом.
6. Цель поездки ограничена 40 символами, как в бланке заполнения визы.

*Таблица**1*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Информация о туре | Тип | Ограничения на исходные данные |
| 1 | Название страны | Строковый | От 2 до 59 |
| 2 | Вид перевозки | Строковый | Выбор между наземным и воздушным транспортом |
| 3 | Стоимость | Рациональный | От 0 до +∞, каждая стоимость ограничена двумя знаками после запятой |
| 4 | Вид отеля для проживания | Целочисленный | От 1 до 5 |
| 5 | Длительность тура | Целочисленный | От 1 до 365 |
| 6 | Цель поездки | Строковый | Не более 40 символов |

* 1. Выходные данные и формы

По запросу пользователя программа может выдать результат, который показан ниже (Таблица 2). Данная форма является единственной и достаточной для выдачи информации по всем запросам.

*Таблица**2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название страны | Вид перевозки | Стоимость (рублей) | Вид отеля для проживания (звезды) | Длительность тура (дни) | Цель поездки |
| США | Воздушный | 39990.00 | 4 | 14 | Посещение музеев |
| Бразилия | Наземный | 25990.00 | 3 | 7 | Визит родственников |
| Япония | Воздушный | 32990.00 | 5 | 9 | Посещение храмов |
| Китай | Наземный | 19990.00 | 2 | 5 | Осмотр поселений |

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММЕ

Согласно заданию, необходимо создать набор туров по странам Европы из Санкт-Петербурга, включающий следующую информацию о каждом туре: название страны, вид перевозки, стоимость, вид отеля для проживания, длительность тура в днях, цель поездки.

Любой информационный блок, описанный выше, может быть изменен администратором. Опишем основные возможности, опираясь на характерные функции ИС, которые должна поддерживать программа в процессе создания и ведения файла.

Основные:

1. Создание файла.
2. Просмотр файла.
3. Добавление новых записей в файл.
4. Редактирование записей в файл.
5. Удаление записей из файла.

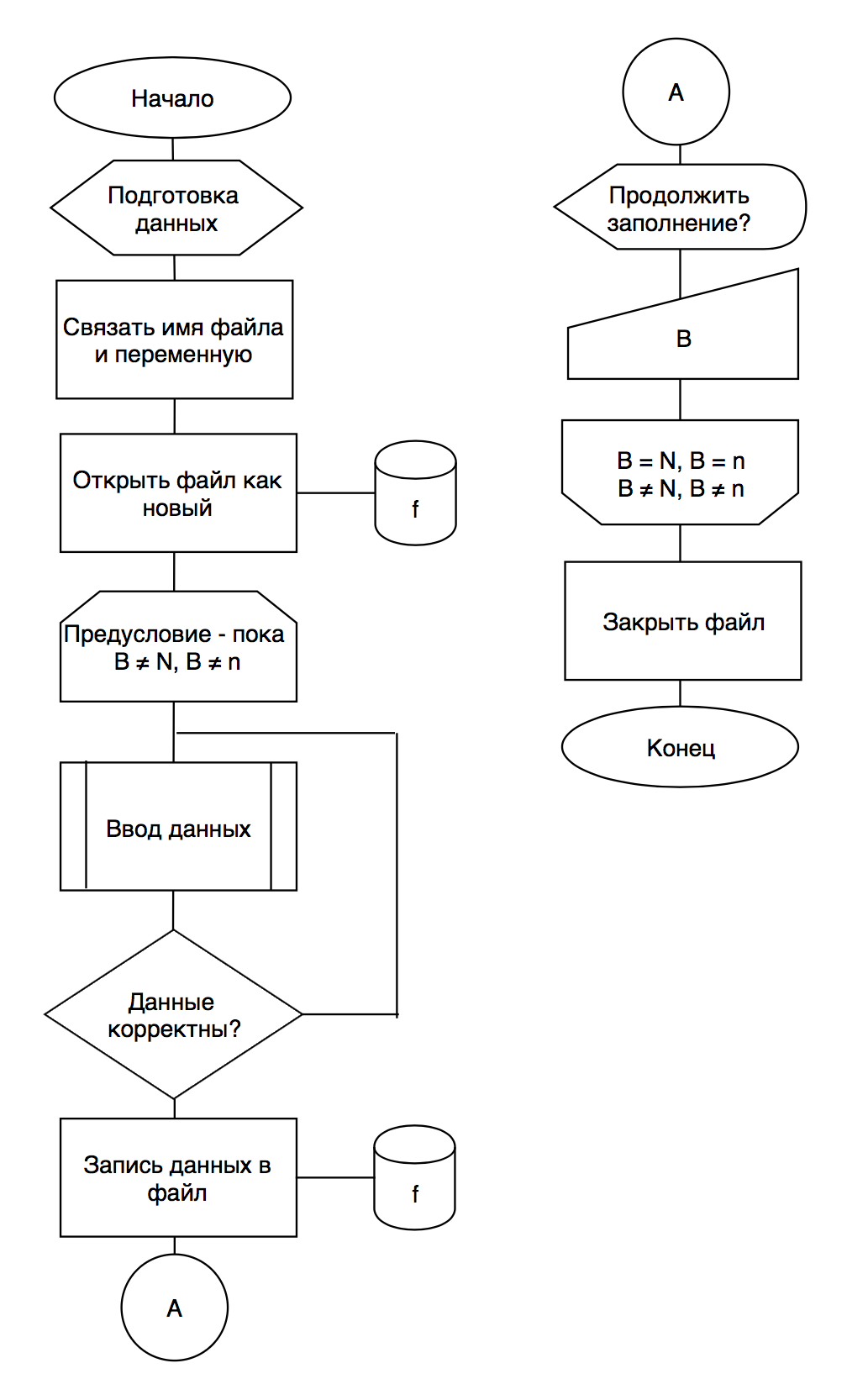
Дополнительные:

1. Выдача информации в диалоговом окне, согласно запросу пользователя: список туров по названию введенной страны; список туров, имеющих отель для проживания категории три.

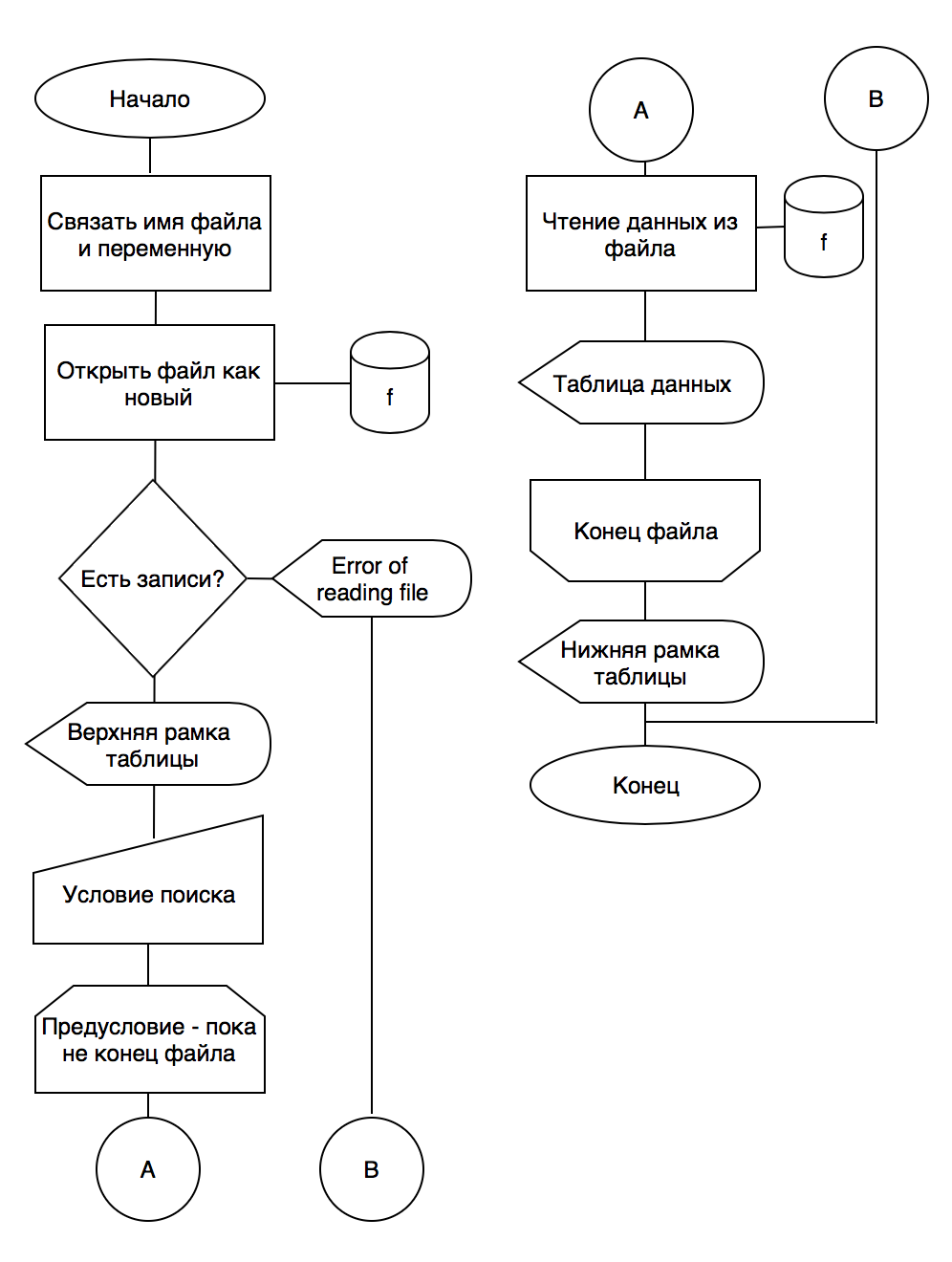
Сервисные:

1. Настройка сортировки для выдачи информации;
2. Защита от неверного ввода данных.
3. СХЕМА АЛГОРИТМА И ЕЁ ОПИСАНИЕ

Ниже покажем два алгоритма к программе (Рисунок 1 и 2):



*Рисунок 1 Алгоритм проверки корректности ввода данных*



*Рисунок 2 Алгоритм выдачи информации по условию*

1. ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Согласно поставке задачи можно сделать вывод, что программа должна хранить, обрабатывать и выводить большой объем информации. Все вышеперечисленное можно реализовать с помощью базы данных (БД), определение:

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

В нашемслучае взаимодействие с базой данных осуществляется с помощью системы управления базами данных (СУБД), определение:

Система управления базами данных (СУБД) — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации.

В среде СУБД можно выделить основных компонент ­— программное обеспечение.

Этот компонент включает операционную систему, программное обеспечение самой СУБД, прикладные программы, включая и сетевое программное обеспечение, если СУБД используется в сети. Обычно приложения создаются на языках третьего поколения, таких как С, COBOL, Fortran, Ada или Pascal.

Из всех вышеперечисленных языков программирование был хорошо изучен и, как следствие, выбран для работы над заданием — С. Выбранная среда программирования — Xcode. Выбранный язык и среда программирования поддерживает все необходимые действия для работы с СУБД и БД, что позволяет нам выполнить задачу целиком и полностью.

Достоинства языка С:

* Си обеспечивает полный набор операторов структурного программирования. Си предлагает необычно большой набор операций.
* Си поддерживает указатели на переменные и функции.
* В своем составе Си содержит препроцессор, который обрабатывает текстовые файлы перед компиляцией.
* Си-гибкий язык, позволяющий принимать в конкретных ситуациях самые разные решения.  
    
  Недостатки языка С:
* Язык Си предъявляет достаточно высокие требования к квалификации использующего его программиста. При изучении Си желательно иметь представление о структуре и работе компьютера. Большую помощь и более глубокое понимание идей Си, как языка системного программирования, обеспечат хотя бы минимальное знание языка ассемблер. Уровень старшинства некоторых операторов не является общепринятым, некоторые синтаксические конструкции могли бы быть лучше. Тем не менее, как оказалось Си – чрезвычайно эффективный и выразительный язык, пригодный для широкого класса задач.

1. РАЗРАБОТКА ПРОГРАММНОГО КОДА

Разработка любой программы состоит из нескольких этапов, грамотная реализация которых является обязательным условием для получения хорошего результата. Четкое следование выверенным временем этапам разработки программного обеспечения становится основополагающим критерием. Рассмотрим каждую стадию общепризнанной методологии разработки ПО:

1. Анализ требований.
2. Проектирование.
3. Кодирование.
4. Тестирование и отладка.
5. Внедрение.

Программный код — это компьютерная программа, написанная на определенном языке программирования по алгоритму, заданному педагогическим и технологическим сценарием.

1. Подпрограмма оболочки.
2. Подпрограмма создания записей.
3. Подпрограмма просмотра записей.
4. Подпрограмма запроса паролей.
5. Подпрограмма меню администратора.
6. Подпрограмма изменения записей.
7. Подпрограмма удаления записей.
8. Подпрограмма сортировки.
9. Подпрограмма сортировки по звездам.
10. Подпрограмма сортировки по цене.

Код данных подпрограмм, приведенных выше, показан в Приложении 1.

1. ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ

Под отладкойпрограммыпонимается процесс, позволяющий получить программное обеспечение, полностью функционирующее с требующимися характеристиками в заданной области входных данных.  Отладка не является разновидностью тестирования, хотя слова «отладка» и «тестирование» часто используют как сино­нимы. Эти два вида деятельности очень тесно связаны и поэтому они обычно рассматриваются совместно. Под ними подразумеваются разные виды деятельности:

Отладка – деятельность, направленная на установление точной природы известной ошибки, для дальнейшего исправления этой ошибки.

Программные ошибки, как правило, делятся на три вида:

1. Синтаксическая ошибка – неправильное употребление синтаксических конструкций, например употребление оператора цикла do без while.
2. Семантическая ошибка – нарушение семантики той или иной конструкции, например передача функции параметров, не соответствующих ее аргументам.
3. Логическая ошибка – нарушение логики программы, приводящее к неверному результату. Это наиболее трудный для "отлова" тип ошибки, ибо подобного рода ошибки, как правило, кроются в алгоритмах и требуют тщательного анализа.

При отладке программы было протестировано множество вариантов входных значений, которые полностью бы удовлетворяли всевозможные пользовательские требования при использовании программы.

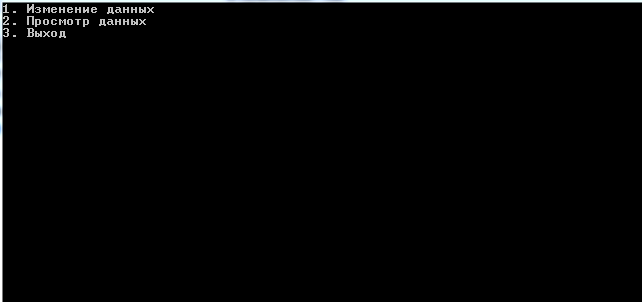
В результате отладки было выявлено и устранено множество синтаксический и логических ошибок, которые приводили к ошибкам в ходе использования программы, либо к полной остановки программы.

Тестирование – деятельность, направленная на обна­ружение ошибок. Результаты тестирования являются исходными данными для отладки.

Ниже покажем контрольный пример, который предусмотрит все логические функции программы.

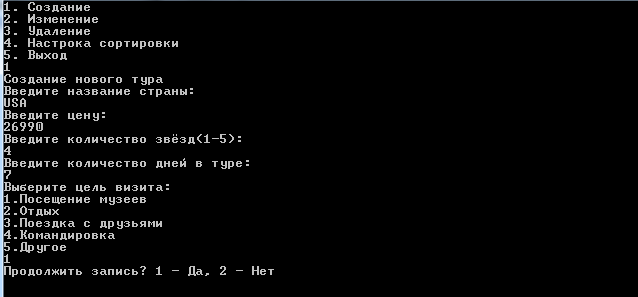
* 1. Контрольный пример

При запуске отображается экранная форма, которая отражает основные возможности программного продукта. Покажем ниже (Рисунок 3).



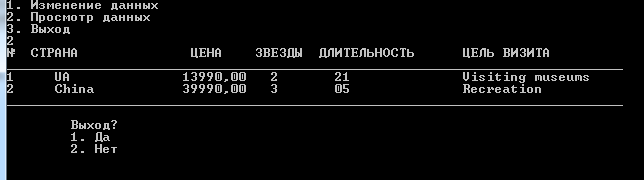
*Рисунок 3* Подпрограмма оболочки

Основной функцией при работе с данными из файла является создание новых записей (Рисунок 4). Данная подпрограмма защищена от неправильного ввода типа данных.



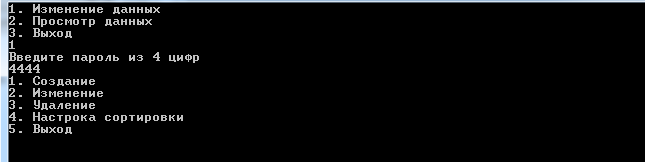
*Рисунок 4* Подпрограмма создания записей

Экранная форма, показанная ниже (Рисунок 5), отражает целостные данные, хранящиеся в файле.



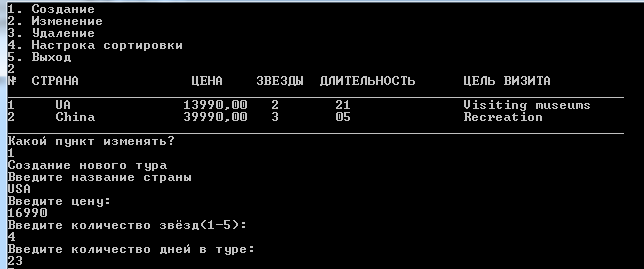
*Рисунок 5* Подпрограмма просмотра записей

Перед тем, как предоставить возможности администрирования необходимо ввести пароль, как показано ниже (Рисунок 6). После будут показаны возможности для изменения данных в файле.



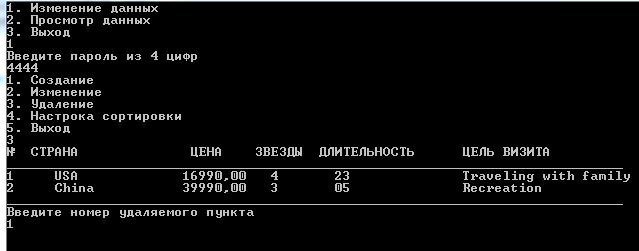
*Рисунок 6* Подпрограмма запроса паролей и меню администратора

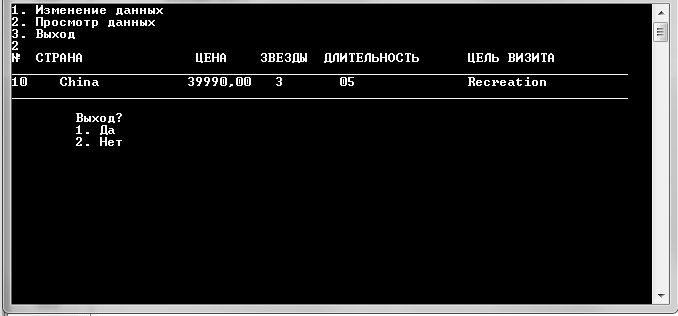
Вторая возможность в меню администрирования является изменение данных, которые записаны в файле. Выбираем номер строки в таблице и изменяем выбранные пункты, как показано ниже (Рисунок 7).



*Рисунок 7* Подпрограмма изменения записей

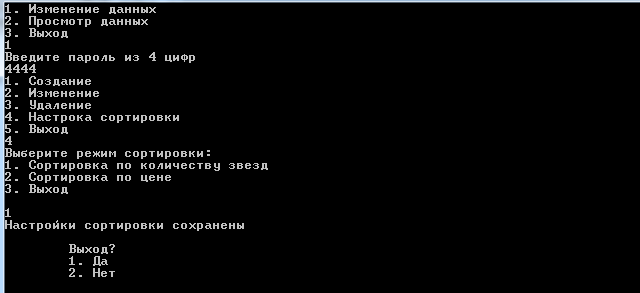
Следующая подпрограмма предоставляет удаление по номеру строки из таблицы (Рисунок 8). Все остальные строки, если они находились ниже, поднимаются выше и заменяют удаленную строку (Рисунок 9).

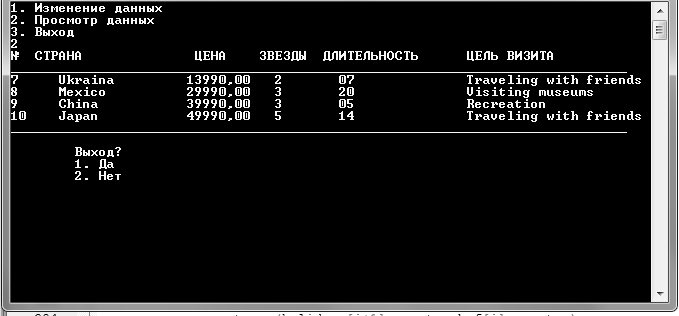
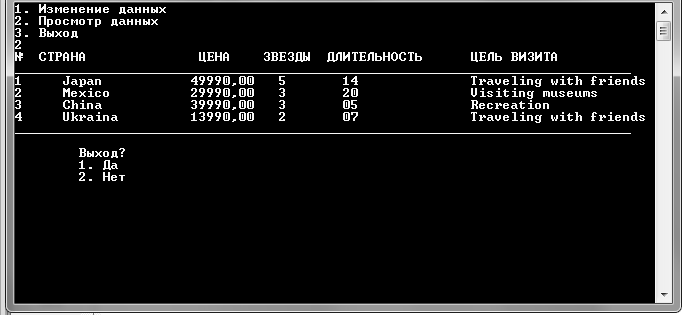




*Рисунок 8 и 9* Подпрограмма удаления записей

Последняя возможность в меню администрирования – настройка сортировки (Рисунок 9). Между экранными формами были добавлены записи в файл, для более полноценного отражения сортировок. В данной подпрограмме есть возможность выбора сортировки по цене или по количеству звезд (Рисунок 10 и 11).



*********Рисунок 9, 10, 11* Подпрограмма настройки сортировки

1. АНАЛИЗ РЕЗУЛЬТАТОВ

После выполнения работы проанализируем ход действий. В ходе чтения формулировки задания были созданы входные и выходные формы программы (Раздел 1.2 и 1.3). После были определены функциональные требования, полностью удовлетворяющие все возможности, которые должна поддерживать программа в процессе создания и ведения файла (Раздел 2). Далее были разработаны алгоритмы, соответствующие всем требованиям, описанных в предыдущих пунктах (Раздел 3). После разработки алгоритмов было сформулировано и обосновано решение о выборе программного обеспечения для написания ИС (Раздел 4). Далее с помощью алгоритмов и выбранного программного обеспечения был разработан программный код (Раздел 5). При выполнении программы были выявлены ошибки, которые с помощью отладки и тестирования были исправлены и показаны в контрольном примере (Раздел 6).

Таким образом, после анализа проделанной работы, можно судить о том, что программа полностью удовлетворяет всевозможные пользовательские требования при использовании. Поставленная задача выполнена в полном объеме.

ПРИЛОЖЕНИЕ 1

**Программный код лабораторной работы**

//

// main.c

// labaFinal

//

// Created by Дмитрий Матвеенко on 31.05.2018.

// Copyright © 2018 Дмитрий Матвеенко. All rights reserved.

//

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <locale.h>

#define n 10

void make(void); //Создание записей

void view(void); //Просмотр записей

int passw(void); //Запрос пароля

void edit(void); //Меню администратора

void rem(void); //Изменение записей

void del(void); //Удаление записей

void sortStar(void); //Сортировка по звездам

void sortset(void); //Меню сортировки

void sortPrice(void); //Меню сортировки цены

**//****Подпрограмма «оболочки»**

struct str {

char country[20];

int star;

int day;

float price;

char target[50];

}holidays[n];

struct b {

char country[20];

int star;

int day;

float price;

char target[50];

}buf[n];

FILE \*fl;

int main() {

setlocale(LC\_ALL, "Rus");

int a=0,b;

do{

a=0;

printf("1. Изменение данных\n2. Просмотр данных\n3. Выход\n");

fflush(stdin);

scanf("%d",&b);

switch(b) {

case 1: edit();break;

case 2: view();break;

case 3:break;

default:printf("Error\n");break;

}

printf("\n\tВыход?\n\t1. Да\n\t2. Нет\n");

fflush(stdin);

scanf("%d",&b);

switch(b) {

case 1: break;

case 2: a=1;system("cls"); break;

default: printf("Ошибка ввода\n"); break;

}

}while(a==1);

return 0;

}

**//Подпрограмма «просмотр записей»**

void view (void) {

setlocale(LC\_ALL, "Rus");

int i;

FILE \*fl;

fl = fopen("d.txt", "r");

if (fl != NULL) {

fread(holidays, sizeof(holidays), 1, fl);

fclose(fl);

printf("№ %-18s %-7s %-7s %-17s %-20s\n", "СТРАНА", "ЦЕНА", "ЗВЕЗДЫ", "ДЛИТЕЛЬНОСТЬ", "ЦЕЛЬ ВИЗИТА");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(i=0;i<n;i++){

if((holidays[i].price!=0)&&(holidays[i].star!=0)&&(holidays[i].day!=0)) {

printf("%-5d %-15s %-10.2f %-7.1d %-15.2d %-20s\n",i+1,holidays[i].country, holidays[i].price, holidays[i].star, holidays[i].day, holidays[i].target);

}

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

else {

puts("Error of reading file");

fclose(fl);

}

}

**//Подпрограмма «создание записей»**

void make(void) {

setlocale(LC\_ALL, "Rus");

int a = 1,b=0;

int i = 0;

while (a==1) {

if(i==n){

puts("Места для записи больше нет");

break;

}

puts("Создание нового тура\nВведите название страны:");

fflush(stdin);

gets(holidays[i].country);

puts("Введите цену:");

fflush(stdin);

scanf("%f", &holidays[i].price);

puts("Введите количество звёзд(1-5):");

fflush(stdin);

scanf("%d", &holidays[i].star);

if((holidays[i].star<1)||(holidays[i].star>5)) {

printf("ERROR");

break;

}

puts("Введите количество дней в туре:");

fflush(stdin);

scanf("%d", &holidays[i].day);

puts("Выберите цель визита:\n1.Посещение музеев\n2.Отдых\n3.Поездка с друзьями\n4.Командировка\n5.Другое");

fflush(stdin);

scanf("%d", &b);

switch(b)

{ case 1:strcpy(holidays[i].target,"Visiting museums");break;

case 2:strcpy(holidays[i].target,"Recreation");break;

case 3:strcpy(holidays[i].target,"Traveling with friends");break;

case 4:strcpy(holidays[i].target,"Business trip");break;

case 5:strcpy(holidays[i].target,"Other");break;

default:printf("ERROR");break;

}

i++;

puts("Продолжить запись? 1 - Да, 2 - Нет\n");

scanf("%d", &a);

if(a==1)system("cls");

}

FILE \*fl;

fl = fopen("d.txt", "w");

fwrite(&holidays, sizeof(holidays), 1, fl);

fclose(fl);

i = 0;

}

**//Подпрограмма «запрос пароля»**

int passw(void) {

setlocale(LC\_ALL, "Rus");

int s=0;

int b=4444;

puts("Введите пароль из 4 цифр");

scanf("%d",&s);

if(s==b)return 0;

else {

puts("Ошибка ввода пароля");

return 1;

}

}

**//Подпрограмма «меню администратора»**

void edit(void) {

int a;

setlocale(LC\_ALL, "Rus");

if(passw()==0) {

printf("1. Создание\n2. Изменение\n3. Удаление\n4. Настрока сортировки\n5. Выход\n");

fflush(stdin);

scanf("%d",&a);

switch(a) {

case 1:make();break;

case 2:rem();break;

case 3:del();break;

case 4:sortset();break;

case 5:break;

default:puts("ERROR");

}

}

}

**//Подпрограмма «изменение записей»**

void rem(void) {

setlocale(LC\_ALL, "Rus");

int i,g=1,b=0;

int a=1;

FILE \*fl;

fl = fopen("d.txt", "r");

if (fl != NULL) {

fread(holidays, sizeof(holidays), 1, fl);

fclose(fl);

printf("№ %-18s %-7s %-7s %-17s %-20s\n", "СТРАНА", "ЦЕНА", "ЗВЕЗДЫ", "ДЛИТЕЛЬНОСТЬ", "ЦЕЛЬ ВИЗИТА");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(i=0;i<n;i++){

if((holidays[i].price!=0)&&(holidays[i].star!=0)&&(holidays[i].day!=0)) {

printf("%-5d %-15s %-10.2f %-7.1d %-15.2d %-20s\n",i+1,holidays[i].country, holidays[i].price, holidays[i].star, holidays[i].day, holidays[i].target);

}

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

else {

puts("ERROR READING");

fclose(fl);

}

while(g==1) {

puts("Какой пункт изменять?");

fflush(stdin);

scanf("%d",&a);

puts("Создание нового тура\nВведите название страны");

fflush(stdin);

gets(holidays[a-1].country);

//scanf("%s", &holidays[a-1].country);

puts("Введите цену:");

fflush(stdin);

scanf("%f", &holidays[a-1].price);

puts("Введите количество звёзд(1-5):");

fflush(stdin);

scanf("%d", &holidays[a-1].star);

if((holidays[a-1].star<1)||(holidays[a-1].star>5)) {

printf("ERROR");

break;

}

puts("Введите количество дней в туре:");

fflush(stdin);

scanf("%d", &holidays[a-1].day);

puts("Выберите цель визита:\n1.Посещение музеев\n2.Отдых\n3.Поездка с друзьями\n4.Командировка\n5.Поездка с семьей\n6.Другое\n");

fflush(stdin);

scanf("%d",&b);

switch(b)

{ case 1:strcpy(holidays[a-1].target,"Visiting museums");break;

case 2:strcpy(holidays[a-1].target,"Recreation");break;

case 3:strcpy(holidays[a-1].target,"Traveling with friends");break;

case 4:strcpy(holidays[a-1].target,"Business trip");break;

case 5:strcpy(holidays[a-1].target,"Traveling with family");break;

case 6:strcpy(holidays[a-1].target,"Other");break;

default:printf("ERROR");break;

}

puts("Продолжить запись? 1 - Да, 2 - Нет\n");

scanf("%d",&g);

}

fl = fopen("d.txt", "w");

fwrite(&holidays, sizeof(holidays), 1, fl);

fclose(fl);

}

**//Подпрограмма «удаление записей»**

void del(void) {

int b=0;

int i,a=0;

setlocale(LC\_ALL, "Rus");

FILE \*fl;

fl = fopen("d.txt", "r");

if (fl != NULL) {

fread(holidays, sizeof(holidays), 1, fl);

fclose(fl);

printf("№ %-18s %-7s %-7s %-17s %-20s\n", "СТРАНА", "ЦЕНА", "ЗВЕЗДЫ", "ДЛИТЕЛЬНОСТЬ", "ЦЕЛЬ ВИЗИТА");

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(i=0;i<n;i++){

if((holidays[i].price!=0)&&(holidays[i].star!=0)&&(holidays[i].day!=0)) {

printf("%-5d %-15s %-10.2f %-7.1d %-15.2d %-20s\n",i+1,holidays[i].country, holidays[i].price, holidays[i].star, holidays[i].day, holidays[i].target);

}

}

puts("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

else {

puts("Error of reading file");

fclose(fl);

}

puts("Введите номер удаляемого пункта");

scanf("%d",&b);

a=b-1;

if(b==n-1) //1 b=n-1

{ strcpy(holidays[a].country,"");

strcpy(holidays[a].target,"");

holidays[a].price=0;

holidays[a].day=0;

holidays[a].star=0;

}

for(a=0;a<n;a++){ //2 a=0

strcpy(holidays[a].country,holidays[a+1].country);

strcpy(holidays[a].target,holidays[a+1].target);

holidays[a].price=holidays[a+1].price;

holidays[a].day=holidays[a+1].day;

holidays[a].star=holidays[a+1].star;

}

fl = fopen("d.txt", "w");

fwrite(&holidays, sizeof(holidays), 1, fl);

fclose(fl);

puts("Элемент удален");

}

**//Подпрограмма «меню сортировки»**

void sortset(void)

{ int a;

puts("Выберите режим сортировки:\n1. Сортировка по количеству звезд\n2. Сортировка по цене\n3. Выход\n");

scanf("%d",&a);

switch(a)

{

case 1:sortStar();break;

case 2:sortPrice();break;

case 3:break;

default:puts("ERROR");

}

}

**//Подпрограмма «сортировка по звездам»**

void sortStar(void)

{int i,a;

FILE \*fl;

fl = fopen("d.txt", "r");

if (fl != NULL) {

fread(holidays, sizeof(holidays), 1, fl);

fclose(fl);

}

else {

puts("ERROR READING");

fclose(fl);

}

for(a=0;a<n;a++){

for(i=0; i<n-1;i++){

if(holidays[i].star<holidays[i+1].star){

strcpy(buf[i].country,holidays[i].country);

strcpy(buf[i].target,holidays[i].target);

buf[i].price=holidays[i].price;

buf[i].day=holidays[i].day;

buf[i].star=holidays[i].star;

strcpy(holidays[i].country,holidays[i+1].country);

strcpy(holidays[i].target,holidays[i+1].target);

holidays[i].price=holidays[i+1].price;

holidays[i].day=holidays[i+1].day;

holidays[i].star=holidays[i+1].star;

strcpy(holidays[i+1].country,buf[i].country);

strcpy(holidays[i+1].target,buf[i].target);

holidays[i+1].price=buf[i].price;

holidays[i+1].day=buf[i].day;

holidays[i+1].star=buf[i].star;

}

}

}

fl = fopen("d.txt", "w");

fwrite(&holidays, sizeof(holidays), 1, fl);

fclose(fl);

puts("Настройки сортировки сохранены");

}

**//Подпрограмма «сортировка цены»**

void sortPrice(void)

{int i,a;

FILE \*fl;

fl = fopen("d.txt", "r");

if (fl != NULL) {

fread(holidays, sizeof(holidays), 1, fl);

fclose(fl);

}

else {

puts("ERROR READING");

fclose(fl);

}

for(a=0;a<n;a++){

for(i=0; i<n-1;i++){

if(holidays[i].price>holidays[i+1].price){

strcpy(buf[

i].country,holidays[i].country);

strcpy(buf[i].target,holidays[i].target);

buf[i].price=holidays[i].price;

buf[i].day=holidays[i].day;

buf[i].star=holidays[i].star;

strcpy(holidays[i].country,holidays[i+1].country);

strcpy(holidays[i].target,holidays[i+1].target);

holidays[i].price=holidays[i+1].price;

holidays[i].day=holidays[i+1].day;

holidays[i].star=holidays[i+1].star;

strcpy(holidays[i+1].country,buf[i].country);

strcpy(holidays[i+1].target,buf[i].target);

holidays[i+1].price=buf[i].price;

holidays[i+1].day=buf[i].day;

holidays[i+1].star=buf[i].star;

}

}

}

fl = fopen("d.txt", "w");

fwrite(&holidays, sizeof(holidays), 1, fl);

fclose(fl);

puts("Настройки сортировки сохранены");

}