Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Дисциплина: Программирование

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Программный модуль управления туристическими программами.

Студент: гр 053551 Шеверев А.О.

Руководитель: Летохо А.С.

Минск 2022

Содержание

Введение…………………………………………………………………………3

1 Разработка программного модуля управления туристическими ………….4

2.Методика работы с полученной программой………………………………7

Заключение……………………………………………………………………...9

Литература………………………………………………………………………10

Приложение 1. Исходный код………………………………………………….11

Введение

Santaren. Программный модуль управления туристическими программами.

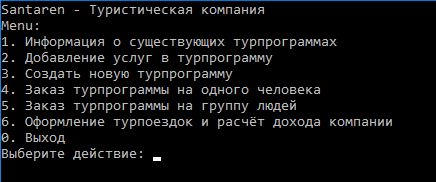
В каждую программу входит следующая информация: маршрут, дата, количество мест, общая стоимость тура. Программа состоит из набора обязательных и необязательных туристических услуг.

Пользователь может самостоятельно сформировать свою программу либо выбрать полную программу. Возможно создание новых туристических программ и добавление услуг в существующие. Запись туристов на каждую программу, оформление туристических поездок по мере комплектования групп. Расчет доходов компании.

1. Разработка программного модуля управления туристическими

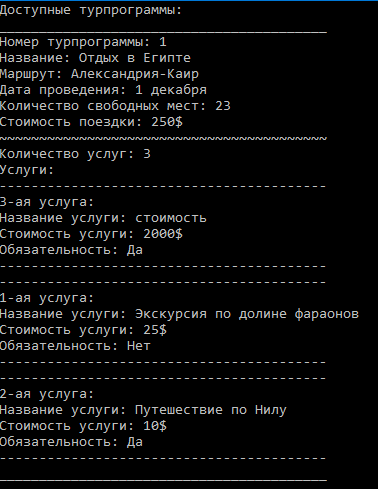
В программе планируется реализовать простой интерфейс для работы управления туристических программ. Интерфейс будет включать в себя возможность указания маршрута, дату маршрута, количество ограниченных мест и общую стоимость тура.

Вид разработанного интерфейса:



Вкладка 1:

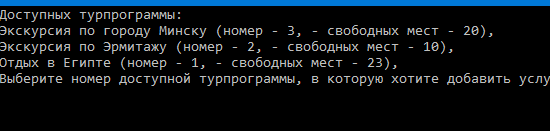
Данная вкладка будет включать в себя возможность просмотра информации о уже добавленных с помощью данного приложения турпрограмм.



Вкладка 2:

Позволяет добавить услуги в уже существующую турпрограмму.

Услугу можно объявить как обязательной так и не обязательной.



Вкладка 3:

Отвечает за создание совершенно новой турпрограммы, включает в себя возможность указания номера, названия, стоимости , информации о самом туре:

\*Маршрут

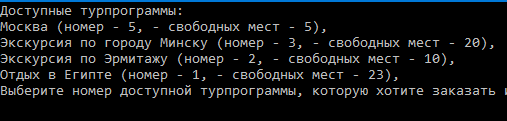
\*Количество свободных мест

\*Дата поездки

\* Добавление услуг

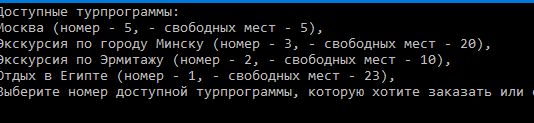
Вкладка 4:

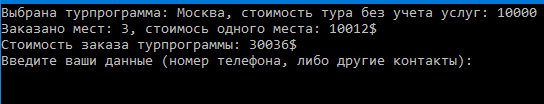
Заказ турпрогаммы на одного человека.

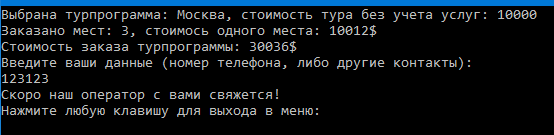


Вкладка 5:

Заказ турпрогаммы на компанию людей.



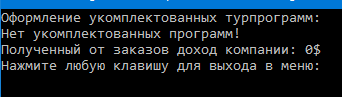




Вкладка 6: Оформление турпоездок и расчёт дохода компании

Если оформленных турпрограмм нет, то на экран будет выведено

Следующие-

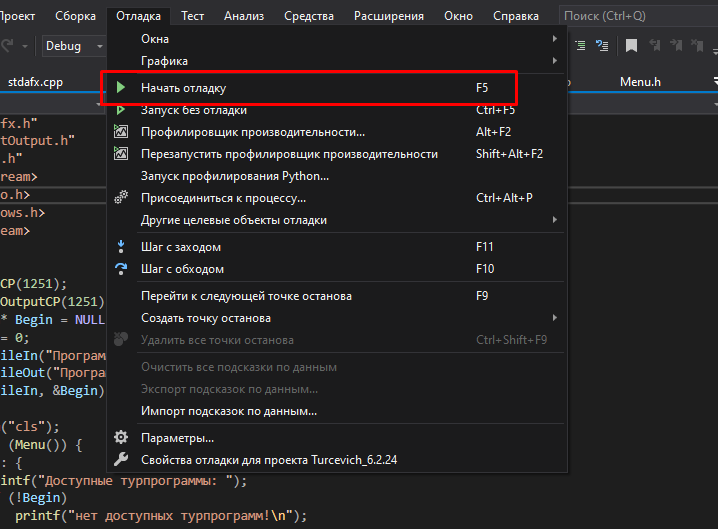


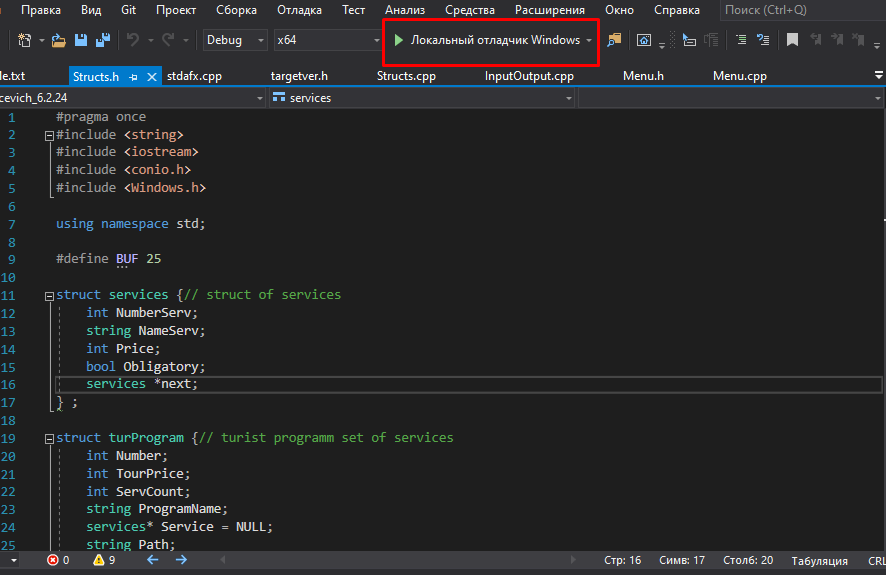
Методика работы с полученной программой

Для инициализации и запуска программы на персональном компьютере должен быть установлен C++ компилятор, в нашем случае это необходимо установить Visual Studio.

Вся курсовая работа проводилась на Visual Studio 2019 и работала должным образом.

После запуска компилятора и открытия проекта нашей программы все что необходимо сделать, эта нажать это cntrl + F5 либо же во вкладе Отладка скриншот 1 нажать начать отладку, и еще одна возможность скриншот 2.

Скриншот 1

Скриншот 2

Заключение

В результате работы над курсовым проектом было разработано программное средство для управления туристическими программами.

Общая последовательность работы с программой следующая: вначале открывается файл базы данных и информация считывается в динамический двунаправленный список.

После этого пользователю предлагается выбор функций, утвержденных в ТЗ. Пользовательский интерфейс может быть реализован в виде текстового или графического меню. После завершения работы с программой информация из динамического списка записывается обратно в файл, вся динамическая память освобождается, а файл закрывается.

Список использованных источников

1. Морозов А.Д. Введение в теорию фракталов. Москва-Ижевск: Институт компьютерных исследований, 2002, 160с.
2. Пайтген Х.-О. Рихтер П.Х. Красота фракталов. Образы комплексных динамических систем. Издательство: Мир Год: 1993 Страниц: 206
3. Фрактальная геометрия природы Б.Мандельброт Издательство: Институт компьютерных исследований, 2002, Страниц: 656
4. Сайт http://fractalworld.xaoc.ru – теория и принципы алгоритмов
5. Cайт http://Wikipedia.org – теория

Приложение 1. Исходный код

//---------------------------------------------------------------------------

#include "stdafx.h"

#include "InputOutput.h"

#include "Menu.h"

#include <iostream>

#include <conio.h>

#include <Windows.h>

#include <fstream>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

turProgram\* Begin = NULL;

int DOXOD = 0;

ifstream FileIn("Программы.txt");

ofstream FileOut("Программы.txt", ios\_base::app);

FileInit(FileIn, &Begin);

for (;;) {

system("cls");

switch (Menu()) {

case 1: {

printf("Доступные турпрограммы: ");

if (!Begin)

printf("нет доступных турпрограмм!\n");

else

ShowProgram(Begin);

ExMenu();

}break;

case 2:{

printf("Доступных турпрограммы: ");

int T = AvaibleProgram(Begin);

if (!T) {

printf("нет доступных турпрограмм.\nЧтобы добавить услугу необходимо существование турпрограммы!\n");

Sleep(3000);

break;

}

AddServiceInProgram(&Begin);

ClearFile();

FileUpd(FileOut, Begin);

ExMenu();

}break;

case 3:{

puts("Создание новой тур программы:");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

AddProgram(&Begin);

FileOutput(FileOut, Begin);

}break;

case 4:{

printf("Доступные турпрограммы: ");

int T = AvaibleProgram(Begin);

if (!T) {

printf("нет доступных турпрограмм.\nДля выбора программы необходимо ее существование!\n");

Sleep(3000);

break;

}

ChooseProgram(&Begin, true, &DOXOD);

ClearFile();

FileUpd(FileOut, Begin);

ExMenu();

}break;

case 5:{

printf("Доступные турпрограммы: ");

int T = AvaibleProgram(Begin);

if (!T) {

printf("нет доступных турпрограмм.\nДля выбора программы необходимо ее существование!\n");

Sleep(3000);

break;

}

ChooseProgram(&Begin, false, &DOXOD);

ClearFile();

FileUpd(FileOut, Begin);

ExMenu();

}break;

case 6:{

int K;

int N = 0;

printf("Оформление укомплектованных турпрограмм:\n");

for (;;) {

K = OformProgram(&Begin);

N++;

if (!K) {

break;

}

}

if (N == 1)

printf("Нет укомплектованных программ!\n");

else {

ClearFile();

FileUpd(FileOut, Begin);

}

printf("Полученный от заказов доход компании: %d$\n", DOXOD);

ExMenu();

}break;

case 0:{

FileIn.close();

FileOut.close();

exit(0);

}

}

}

system("pause");

return 0;

}

//---------------------------------------------------------------------------

#pragma once

#include <string>

#include <iostream>

#include <conio.h>

#include <Windows.h>

using namespace std;

#define BUF 25

struct services {// struct of services

int NumberServ;

string NameServ;

int Price;

bool Obligatory;

services \*next;

} ;

struct turProgram {// turist programm set of services

int Number;

int TourPrice;

int ServCount;

string ProgramName;

services\* Service = NULL;

string Path;

string Date;

int EmptySpace;

turProgram \*next;

};

void ExMenu();

int Input();// func of input validation

int AddService(services\*\* Head, int i);

int OformProgram(turProgram\*\* Begin);

int AvaibleProgram(turProgram\* Begin);

int ChooseService(services\* Head, int \*Money, int Pass, bool X);

void ChooseProgram(turProgram\*\* Begin, bool X, int \*Money);

void AddServiceInProgram(turProgram\*\* Begin);

int CheckNum(turProgram\* Begin);

void AddProgram(turProgram\*\* Begin);

//---------------------------------------------------------------------------

#pragma once

#include "stdafx.h"

#include "Structs.h"

static bool Obl()

{

char Ch;

Ch = \_getch();

cout << endl;

if (Ch == 'y' || Ch == 'Y')

return true;

else

return false;

}

void ExMenu()

{

printf("Нажмите любую клавишу для выхода в меню: ");

\_getch();

}

int Input() {

char a[BUF];

for (;;) {

gets\_s(a, BUF);

if (a[0] == 'e' && a[1] == 'x' && a[2] == 'i' && a[3] == 't')

return -1;

int b = strlen(a);

if (b >= 10) {

printf("Ошибка недопустимо большое значение. Введите значение меньше 10 знаков: ");

}

else {

for (int i = 0; i < BUF; i++) {

if (a[i] <= 57 && a[i] >= 48) {

b--;

}

}

if (a[0] == '0')

return 0;

else if (!atoi(a))

printf("Error. Enter natural number or control word exit: ");

else if (!b)

return atoi(a);

else

printf("Error. Enter natural number or control word exit: ");

}

}

}

int AddService(services\*\* Head, int i)

{

services \*TempServ = new services;

printf("------------------------------------------\n");

printf("Добавление %d услуги: \n", i);

TempServ->NumberServ = i;

printf("Введите название услуги: ");

getline(cin, TempServ->NameServ);

printf("Введите стоимость услуги ($): ");

TempServ->Price = Input();

printf("Обязательная услуга?(Введите Y, если да, при нажатии другой клавиши - нет): ");

TempServ->Obligatory = Obl();

TempServ->next = \*Head;

\*Head = TempServ;

return 1;

}

int OformProgram(turProgram\*\* Begin)

{

if (\*Begin == NULL) {

return 0;

}

turProgram \*avaible = \*Begin;

if (avaible->EmptySpace == NULL) {

printf("Программа %d оформлена;\n", avaible->Number);

\*Begin = avaible->next;

delete avaible;

return 1;

}

turProgram \*avaible1 = avaible->next;

while (avaible1) {

if (avaible1->EmptySpace == NULL) {

printf("Программа %d оформлена;\n", avaible1->Number);

avaible->next = avaible1->next;

delete avaible1;

return 1;

}

avaible = avaible1;

avaible1 = avaible1->next;

}

return 0;

}

int AvaibleProgram(turProgram \* Begin)

{

int i = 0;

turProgram \*avaible = Begin;

while (avaible) {

if (avaible->EmptySpace > NULL) {

cout << endl<< avaible->ProgramName;

printf(" (номер - %d, - cвободных мест - %d), ", avaible->Number, avaible->EmptySpace);

i++;

}

avaible = avaible->next;

}

if (i!=0)

printf("\n");

return i;

}

int ChooseService(services \*Head, int \*Money, int Pass, bool X)

{

services \*show = Head;

while (show) {

if(X){

\*Money += Pass\*show->Price;

show = show->next;

}

else {

if (show->Obligatory) {

\*Money += Pass\*show->Price;

show = show->next;

}

else {

printf("-----------------------------------------\n");

cout << "Номер услуги: " << show->NumberServ << endl;

cout << "Название услуги: " << show->NameServ << endl;

cout << "Стоимость услуги: " << show->Price << "$" << endl;

printf("Добавить в список услуг?(Введите Y, если да, при нажатии другой клавиши - нет): ");

if (Obl()) {

printf("Добавлено!\n");

\*Money += Pass\*show->Price;

}

else

printf("Не добавлено!\n");

show = show->next;

}

}

}

return 1;

}

void ChooseProgram(turProgram\*\* Begin, bool X, int \*Money)

{

printf("Выберите номер доступной турпрограммы, которую хотите заказать или exit для отмены заказа: ");

int Passenger;

int YourMoney = 0;

int LoL = 0;

for (;;) {

turProgram \*avaible = \*Begin;

int Choice = Input();

if (Choice == -1)

break;

while (avaible) {

if (Choice == avaible->Number) {

system("cls");

if (X) {

Passenger = 1;

avaible->EmptySpace--;

}

else{

printf("Введите количество мест которые хотите заказать: \n");

for (;;) {

Passenger = Input();

if (Passenger > avaible->EmptySpace)

printf("Количества свободных мест недостаточно для данного заказа. Введите меньшее число!\n");

else

break;

}

avaible->EmptySpace -= Passenger;

}

YourMoney += Passenger\*avaible->TourPrice;

printf("Выбрать полную программу или выбрать самостоятельно необязательные услуги?(Введите Y для выбора полной программы, при нажатии другой клавиши - самостоятельно): ");

if (Obl()) {

printf("Выбрана полная программа!\n");

LoL = ChooseService(avaible->Service, &YourMoney, Passenger, true);

}

else {

printf("Необязательные услуги, которые хотите внести в свою программу: \n");

LoL = ChooseService(avaible->Service, &YourMoney, Passenger, false);

}

Sleep(3000);

system("cls");

cout<<"Выбрана турпрограмма: " << avaible->ProgramName<<", стоимость тура без учета услуг: "<< avaible->TourPrice<<endl;

printf("Заказано мест: %d, стоимось одного места: %d$\n", Passenger, YourMoney/Passenger);

printf("Стоимость заказа турпрограммы: %d$\n", YourMoney);

string Contact;

printf("Введите ваши данные (номер телефона, либо другие контакты): \n");

getline(cin, Contact);

printf("Скоро наш оператор с вами свяжется!\n");

break;

}

avaible = avaible->next;

}

if (!LoL)

printf("Такой программы не существует. Выберите из списка доступных: ");

else {

\*Money += YourMoney;

break;

}

}

}

void AddServiceInProgram(turProgram\*\* Begin)

{

printf("Выберите номер доступной турпрограммы, в которую хотите добавить услугу или введите exit для отмены добавления: ");

int LoL = 0;

for (;;) {

turProgram \*avaible = \*Begin;

int Choice = Input();

if (Choice == -1)

break;

while (avaible) {

if (Choice == avaible->Number) {

LoL = AddService(&(avaible->Service), ++avaible->ServCount);

break;

}

avaible = avaible->next;

}

if (!LoL)

printf("Такой программы не существует. Выберите из списка доступных: ");

else

break;

}

}

int CheckNum(turProgram \* Begin)

{

turProgram \*check = Begin;

int A;

A = Input();

for (;;) {

while (check) {

if (A == check->Number) {

printf("Программа с таким номером уже существует. Введите другой номер: ");

A = Input();

check = Begin;

}

else

check = check->next;

}

return A;

}

}

void AddProgram(turProgram\*\* Begin) {

turProgram \*Temp = new turProgram;

printf("Введите номер турпрограммы: ");

Temp->Number = CheckNum(\*Begin);

printf("Введите название турпрограммы: ");

getline(cin, Temp->ProgramName);

printf("Введите стоимость турпрограммы ($): ");

Temp->TourPrice =Input();

printf("Информация о турпрограмме:\n");

printf("Введите маршрут: ");

getline(cin, Temp->Path);

printf("Введите количество свободных мест: ");

Temp->EmptySpace = Input();

printf("Введите дату поездки: ");

getline(cin, Temp->Date);

printf("Добавление услуг: \n");

for (int i = 1; ; i++) {

AddService(&(Temp->Service), i);

printf("Добавить еще услугу?(Введите Y, если да, при нажатии другой клавиши - нет): ");

if (!Obl()) {

Temp->ServCount = i;

break;

}

}

Temp->next = \*Begin;

\*Begin = Temp;

}

//---------------------------------------------------------------------------

#include "stdafx.h"

#include "Menu.h"

int Menu()

{

char Ch;

for (;;) {

puts("Santaren - Туристическая компания\nMenu:");

puts("1. Информация о существующих турпрограммах");

puts("2. Добавление услуг в турпрограмму");

puts("3. Создать новую турпрограмму");

puts("4. Заказ турпрограммы на одного человека");

puts("5. Заказ турпрограммы на группу людей");

puts("6. Оформление турпоездок и расчёт дохода компании");

puts("0. Выход");

printf("Выберите действие: ");

Ch = \_getch();

if (Ch >= 48 && Ch <= 54) {

system("cls");

break;

}

system("cls");

puts("Неправильный ввод. Введите цифру от 0 до 6 соответствующую пункту меню!");

Sleep(3000);

system("cls");

}

return atoi(&Ch);

}

//---------------------------------------------------------------------------

#include "stdafx.h"

#include "InputOutput.h"

void ClearFile()

{

fstream clear\_file("Программы.txt", ios::out);

clear\_file.close();

}

void FileInit(ifstream &FileIn, turProgram\*\* Begin)

{

turProgram\* Temp = new turProgram;

printf("loading file....\n");

if (!FileIn.is\_open()) {

puts("Файл не может быть открыт!");

\_getch();

exit(-1);

}

int i = 0;

const int N = 1000;

char str[N];

while (!FileIn.eof())

{

FileIn.getline(str, N, '\n');

i++;

}

FileIn.clear();

FileIn.seekg(0, ios\_base::beg);

char \*\*BUFER = new char\*[i - 1];

for (int A = 0; A < i - 1; A++)

BUFER[A] = new char[N];

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

FileIn.getline(BUFER[B], N);

InitProgram(&Temp, BUFER[B]);

Temp->next = \*Begin;

\*Begin = Temp;

}

for (int A = 0; A < i - 1; A++)

delete BUFER[A];

delete[] BUFER;

}

void FileOutput(ofstream &File, turProgram \*A) {

File << A->Number << '~';

File << A->ProgramName << '~';

File << A->Path << '~';

File << A->Date << '~';

File << A->EmptySpace << '~';

File << A->TourPrice << '~';

File << A->ServCount << '`';

services \*Temp = A->Service;

for (int i = 1; i <= A->ServCount; i++) {

File << Temp->NumberServ << '~';

File << Temp->NameServ << '~';

File << Temp->Price << '~';

File << Temp->Obligatory << '`';

Temp = Temp->next;

}

File << endl;

};

void InitService(services\*\* Head, char\* Str1, char\* Str2, char\* Str3, char\* Str4)

{

services \*TempServ = new services;

TempServ->NumberServ = atoi(Str1);

TempServ->NameServ = Str2;

TempServ->Price = atoi(Str3);

if (atoi(Str4))

TempServ->Obligatory = true;

else

TempServ->Obligatory = false;

TempServ->next = \*Head;

\*Head = TempServ;

}

void InitProgram(turProgram \*\* Begin, char\* Str)

{

int C = 8;

int Usl = -1;

for (int i = 0, j = strlen(Str); i <= j; i++)

if (Str[i] == '`')

Usl++;

C += Usl \* 4;

char \*\*Temp = new char\*[C];

for (int A = 0; A < C; A++)

Temp[A] = new char[1000];

for (int i = 0, j = strlen(Str), t = 0, k = 0; i <= j; i++, k++) {

if (Str[i] != '~' && Str[i] != '`')

Temp[t][k] = Str[i];

else

{

Temp[t][k] = '\0';

t++;

k = -1;

}

}

turProgram \*Elem = new turProgram;

Elem->Number = atoi(Temp[0]);

Elem->ProgramName = Temp[1];

Elem->Path = Temp[2];

Elem->Date = Temp[3];

Elem->EmptySpace = atoi(Temp[4]);

Elem->TourPrice = atoi(Temp[5]);

Elem->ServCount = atoi(Temp[6]);

for (int i = 0, j = 7; i < Usl; i++) {

InitService(&(Elem->Service), Temp[j], Temp[j + 1], Temp[j + 2], Temp[j + 3]);

j += 4;

}

Elem->next = \*Begin;

\*Begin = Elem;

}

void FileUpd(ofstream &FileOut, turProgram \* Begin)

{

turProgram\* Temp = Begin;

while (Temp) {

FileOutput(FileOut, Temp);

Temp = Temp->next;

}

}

void ShowService(services \* Head)

{

services \*show = Head;

while (show) {

printf("-----------------------------------------\n");

cout << show->NumberServ << "-ая услуга:\n";

cout << "Название услуги: " << show->NameServ << endl;

cout << "Стоимость услуги: " << show->Price << "$" << endl;

cout << "Обязательность: ";

if (show->Obligatory)

cout << "Да\n";

else

cout << "Нет\n";

printf("-----------------------------------------\n");

show = show->next;

}

}

void ShowProgram(turProgram \* Begin)

{

turProgram \*print = Begin;

while (print)

{

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

cout << "Номер турпрограммы: " << print->Number << endl;

cout << "Название: " << print->ProgramName << endl;

cout << "Маршрут: " << print->Path << endl;

cout << "Дата проведения: " << print->Date << endl;

cout << "Количество свободных мест: " << print->EmptySpace << endl;

cout << "Стоимость поездки: " << print->TourPrice << "$" << endl;

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

cout << "Количество услуг: " << print->ServCount << endl;

cout << "Услуги: " << endl;

ShowService(print->Service);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

print = print->next;

}

}

//---------------------------------------------------------------------------