Оглавление

[**1. ВВЕДЕНИЕ** 2](#_Toc9166290)

[**2.ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ** 3](#_Toc9166291)

[**3.РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ПРОГРАММЫ** 4](#_Toc9166292)

[**3.1 Поэтапная разработка программы** 4](#_Toc9166293)

[**4. ТРЕБОВАНИЯ К СИСТЕМЕ** 5](#_Toc9166294)

[4.1 Требования к функциональным характеристикам; 5](#_Toc9166295)

[ Осуществление выбора для клиента; 5](#_Toc9166296)

[ Создание некоторого количества путевок для заполнения «базы данных»; 5](#_Toc9166297)

[ Реализация функции поиска по заданным критериям ; 5](#_Toc9166298)

[4.2 Требования к информационному обеспечению; 5](#_Toc9166299)

[4.3 Требования к лингвистическому обеспечению; 5](#_Toc9166300)

[4.4 Требования к программному обеспечению; 5](#_Toc9166301)

[4.5 Требования к техническому обеспечению. 5](#_Toc9166302)

[**5. ПРОЕКТНАЯ ЧАСТЬ** 6](#_Toc9166303)

[**6. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ** 9](#_Toc9166304)

[**7. ЗАКЛЮЧЕНИЕ** 13](#_Toc9166305)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 14](#_Toc9166306)

[**Приложение А** 15](#_Toc9166307)

[**Приложение Б** 17](#_Toc9166308)

**Описание проекта**

# **1. ВВЕДЕНИЕ**

Представленная курсовая работа выполнена с помощью методов объектно-ориентированного программирования языка С++.

Объектно-ориентированное программирование – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

C++ – компилируемый, статически типизированный язык программирования общего назначения. C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений.

В ходе выполнения данной курсовой работы была реализована программа для продажи путёвок клиентам.Пользователь методом поиска находит тур, который он впоследствии приобретет.

# **2.ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Индивидуальное задание: Программа для продажи путевок клиентам

Функциональные требования: Необходимо разработать программу для заказа товаров в магазине путевок с использованием функции поиска нужного товара по заданным критериям поиска

Уточнение требований: Для продажи должны быть доступны различные виды туров, включая горящие и предусмотрены скидки для постоянных клиентов.

# 

# **3.РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ПРОГРАММЫ**

# **3.1 Поэтапная разработка программы**

Разработка программы состоит из следующих этапов:

1. создание заголовочного файла Headerrrr.h и прописывания в нём дизайна программы путем создания класса Decoration, создание структуры Vacation
2. создание main и прописывания в нем переменных структуры Vacation с критериями путёвок и функции для повторного поиска
3. создание Source.cpp и прописывание в нем функции Search\_Criteria() для проверки на совпадение в путевках введённых критериев, функции poisk() для процедуры ввода критериев и сортировки индексов наиболее совпавших путевок, функции Skidka() для расчета скидки постоянным клиентам и функции returnnn() для вывода путевки;
4. вызов программы

# **4. ТРЕБОВАНИЯ К СИСТЕМЕ**

### Требования к функциональным характеристикам;

### Осуществление выбора для клиента;

### Создание некоторого количества путевок для заполнения «базы данных»;

### Реализация функции поиска по заданным критериям ;

### Требования к информационному обеспечению;

Для работы программы используется руководство пользователя, находящееся в папке с приложением (Инструкция.txt).

### Требования к лингвистическому обеспечению;

Интерфейс программы представлен на русском языке. Клиент-пользователь вводит информацию, используя буквы кириллицы. Писать необходимо грамотно. Система не распознает грамматические ошибки.

### Требования к программному обеспечению;

Программное приложение функционирует на ОС Windows следующих версий: Windows 7, Windows 8, Windows 8.1, Windows 10.

### Требования к техническому обеспечению.

Для успешной работы программы необходимо 512 МБ свободной оперативной памяти.

# **5. ПРОЕКТНАЯ ЧАСТЬ**

Предоставленная курсовая работа основана на структуре «Vacation». В ней присутствуют такие члены структуры как: continent, country, type, number\_of\_stars, payment, days. Так же методами структуры являются Skidka(), returnnn(), Search\_Criteria(), poisk().

#### 5.1.Описание класса «Client»;

Структура «Vacation» отвечает за информацию о путёвке. Обладает следующими методами и свойствами:

continent – переменная типа string отвечает за континент

country - переменная типа string отвечает за страну назначения

type - переменная типа string отвечает за вид отдыха

number\_of\_stars - переменная типа string отвечает за количество звезд в отеле

payment - переменная типа string отвечает за цену в отеле за сутки

days - переменная типа string отвечает за количество дней отдыха

Skidka() – функция типа int, рассчитывает и возвращает стоимость путевки со скидкой постоянного клиента, либо сообщает об отсутствии такой скидки и предлагает завести карту постоянного клиента

returnnn() – функция типа void, выводит предлагаемые путевки

Search\_Criteria() – функция типа double, сравнивает введенные пользователем критерии с элементами членов структур, подситывает совпадения

poisk() – функция, осуществляющая ввод критериев, заданных пользователем, прогоняет их через Search\_Criteria(), сортирует путевки из базы данных по мере их соотвествия пожеланиям пользователя и выводит на экран наиболее подходящие, включая горящие туры.

5.2.Описание класса «Decoration»;

Класс «Decoration» позволяет создать дизайн программы. Он обладает следующими методами и свойствами:

public:

int NotUsed – переменная типа int, отвечающая за изменение цвета на цвет из цветовой консоли.

enum ConsoleColor – переменная перечисляющего типа (enum), включающая в себя список цветов, которые могут использоваться для изменения интерфейса системы.

void SetColor() – функция, отвечающая за изменение цвета текста и фона приложения.

#### 5.3.Используемые библиотеки;

Список библиотек, использованных при выполнении данной курсовой работы:

* #include <iostream> - библиотека для поддержки файлового вывода (ввода) данных встроенных типов. Данная библиотека имеет расширение для чтения и записи новых типов данных.
* #include <windows.h> - позволяет использовать в программе функционал, предоставляемый операционной системой (Windows 95, 98, NT, 2000, XP).
* #include <cstring> - библиотека определяет несколько функций для обработки Cи-строк и массивов.
* #include <conio.h> - заголовочный файл используемый для создания текстового интерфейса пользователя
* #include <string> - позволяет работать со строками
* #include <vector> - позволяет создавать двумерные массивы
* #include <algorithm> - библиотека, позволяющая подсчитывать, сортировать элементы

#### 5.4.Диаграмма классов.

Диаграмма классов – это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их атрибутов, интерфейсов, методов и различных взаимосвязей между ними. UML является языком графического описания для объектного моделирования, системного проектирования и отображения организационных структур. Это своеобразный открытый стандарт, который использует графические обозначения для создания абстрактной модели системы.

В данной курсовой работе представлены классы «Vacation» и «Decoration», их UML диаграммы располагаются ниже:

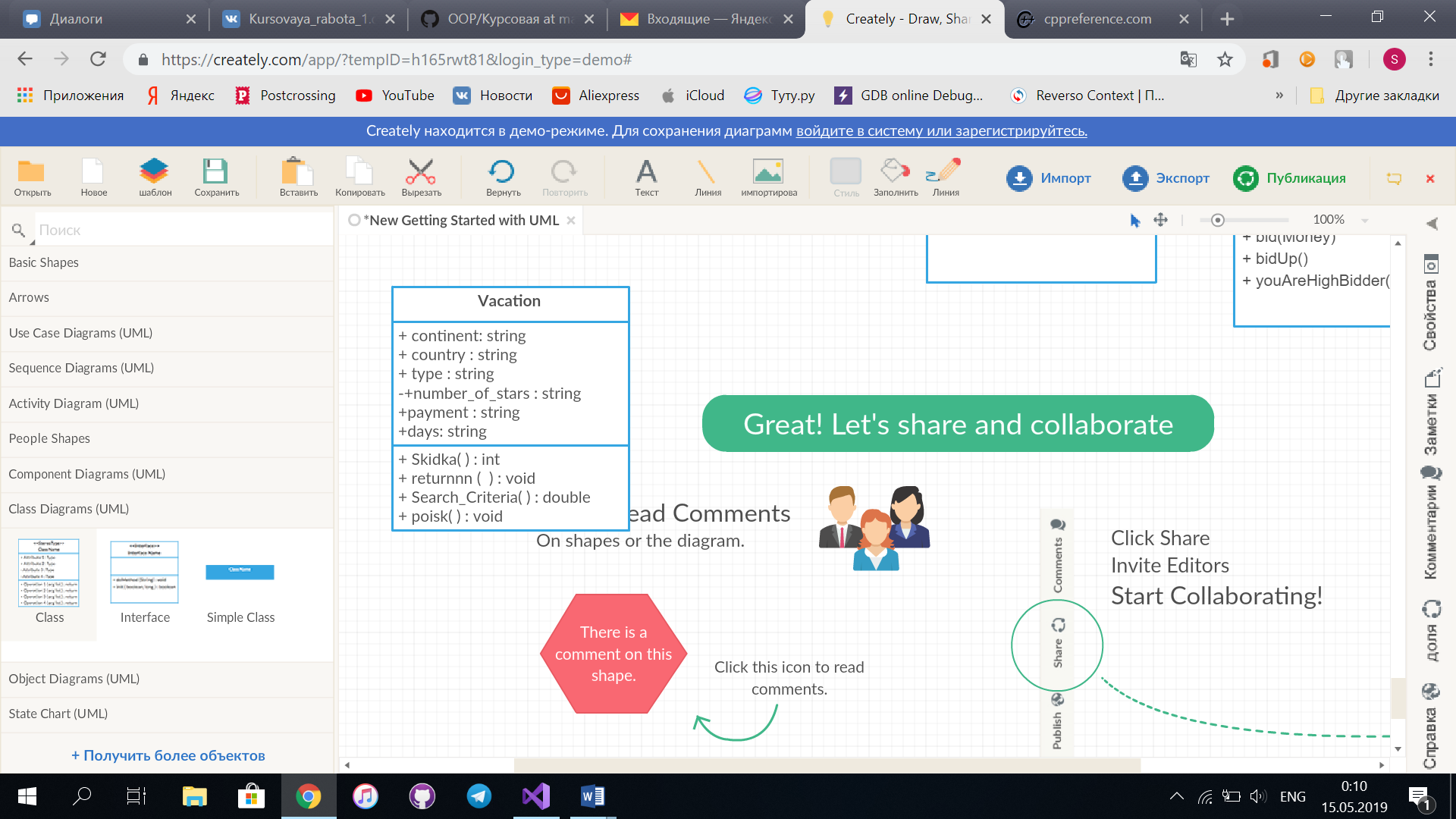


Рисунок 1 UML – диаграмма «Vacation»

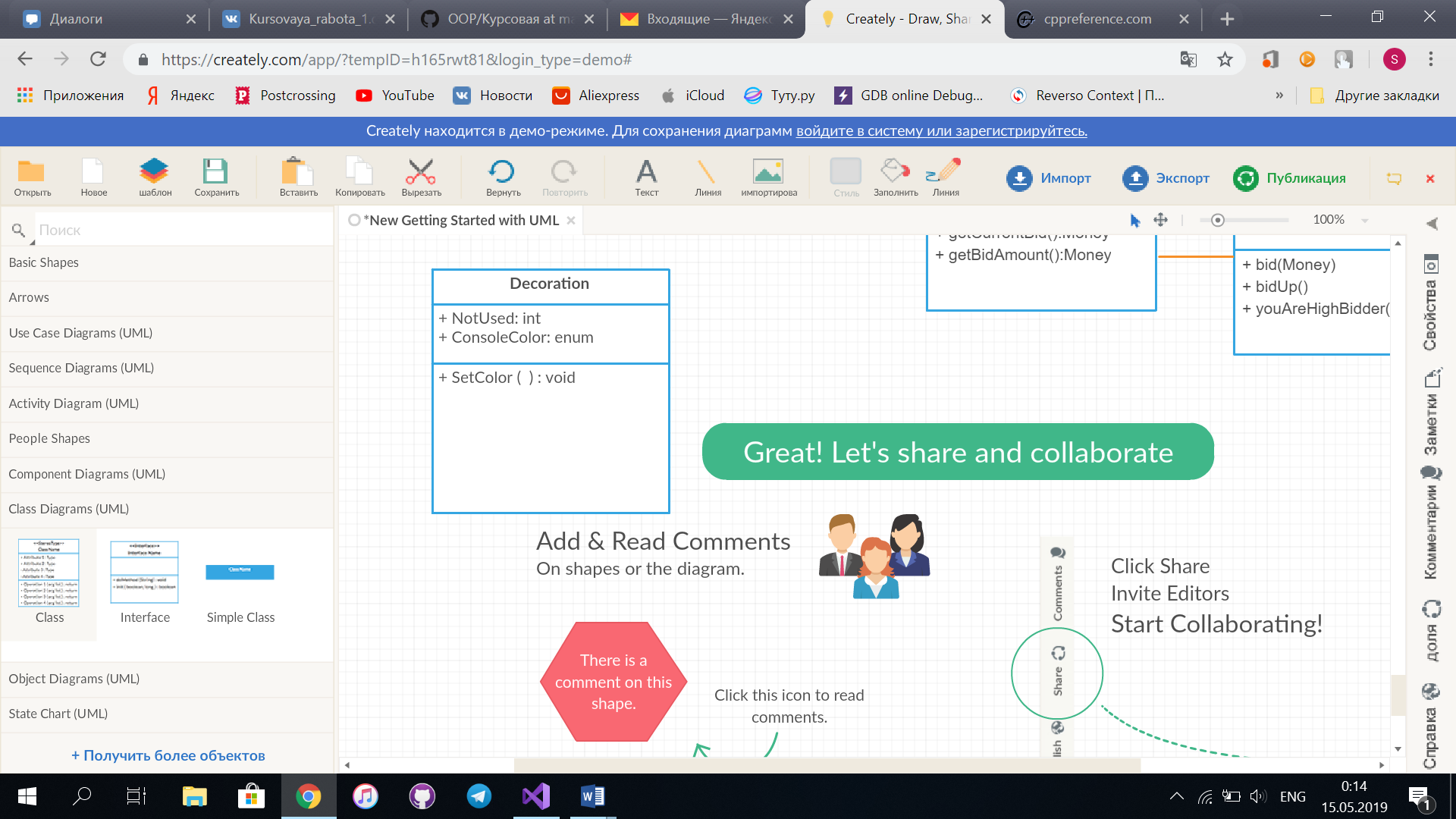


Рисунок 2 UML – диаграмма «Decoration»

# **6. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ**

При запуске программы открывается консольное приложение. На консоли выводится фраза «ТУР АГЕНСТВО». Дальше пользователю дается выбор, ввод F1, F2 или любой другой клавиши. F1 открывает инструкцию пользователя, F2 открывает информацию об авторе, любая другая клавиша начинает поиск путевки.

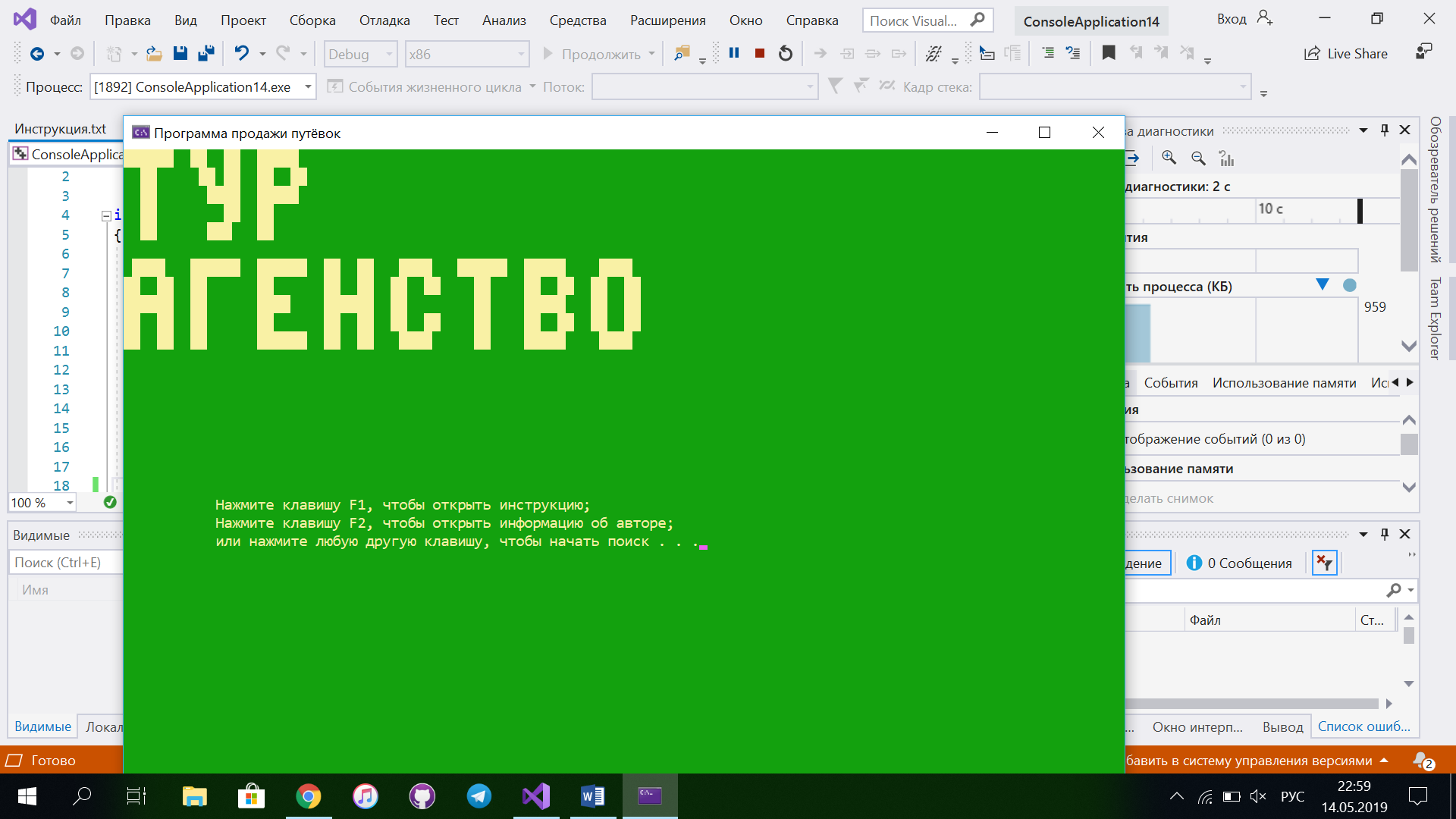


Рисунок 3 Главный экран программы

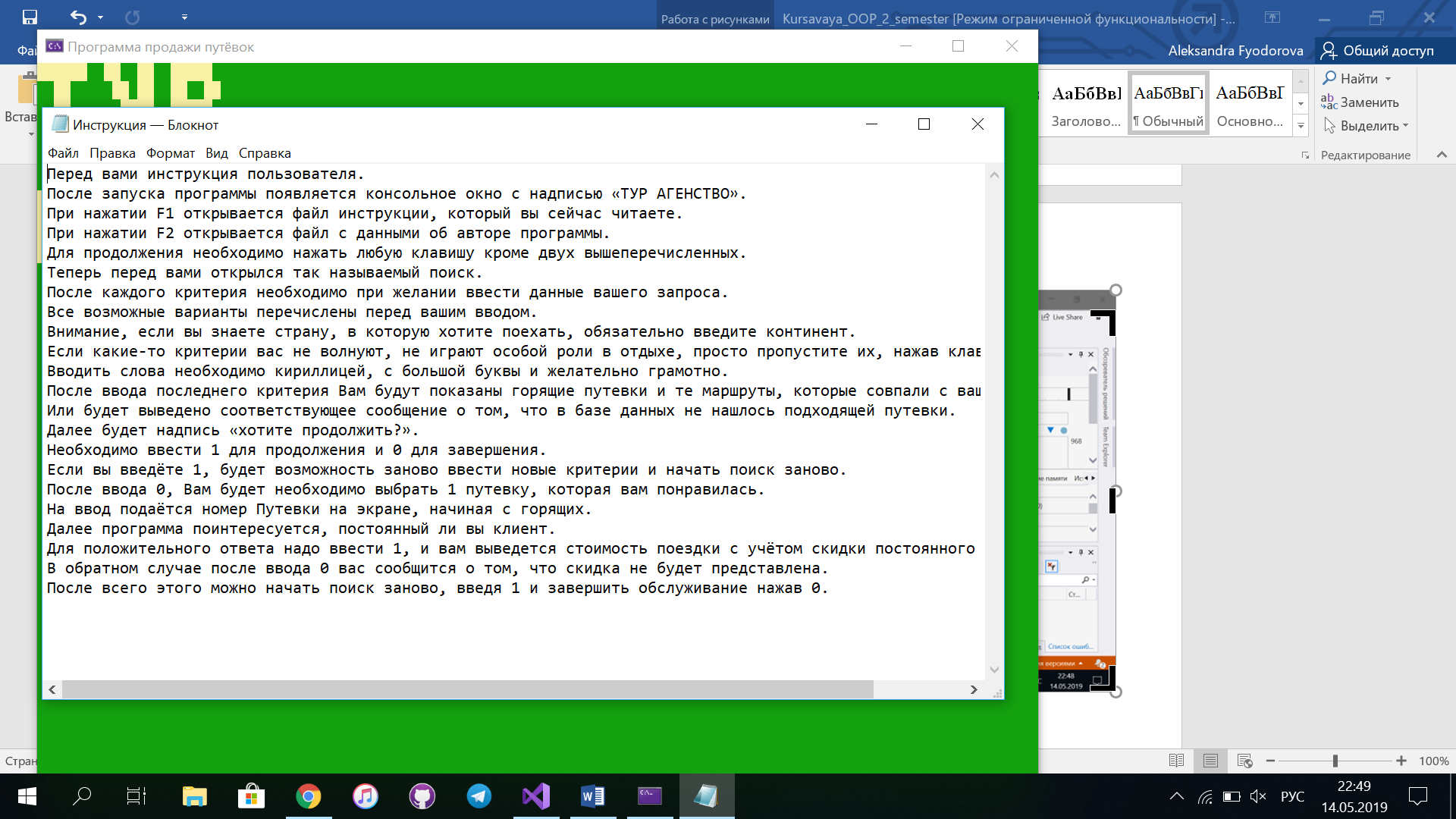


Рисунок 3 Открытие файла с инструкцией

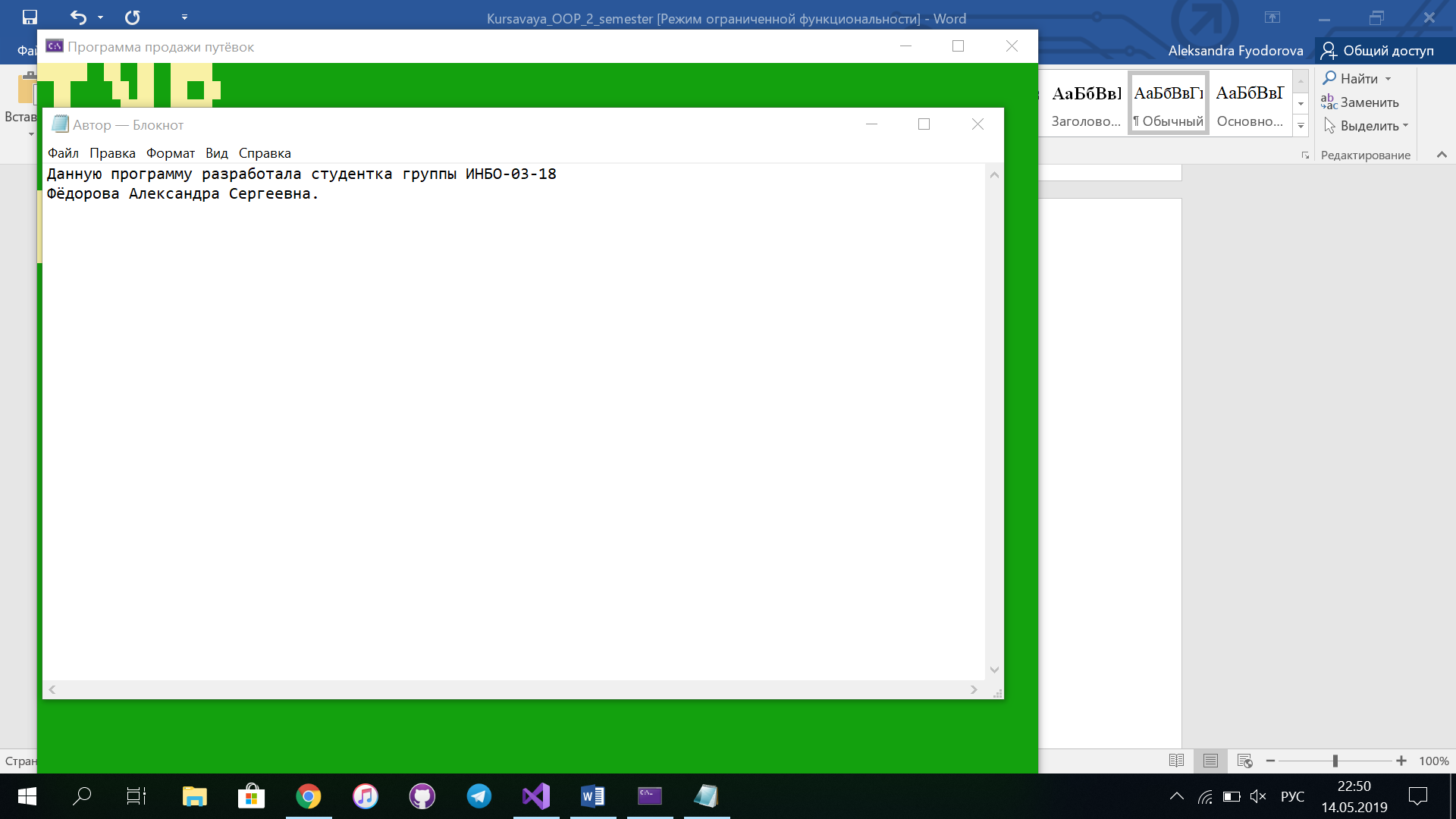


Рисунок 4 Открытие файла с информацией об авторе

После ознакомления со инструкцией и информацией об авторе пользователь может начать поиск путевки (неважные по его мнению критерии можно пропускать, нажав пробел):

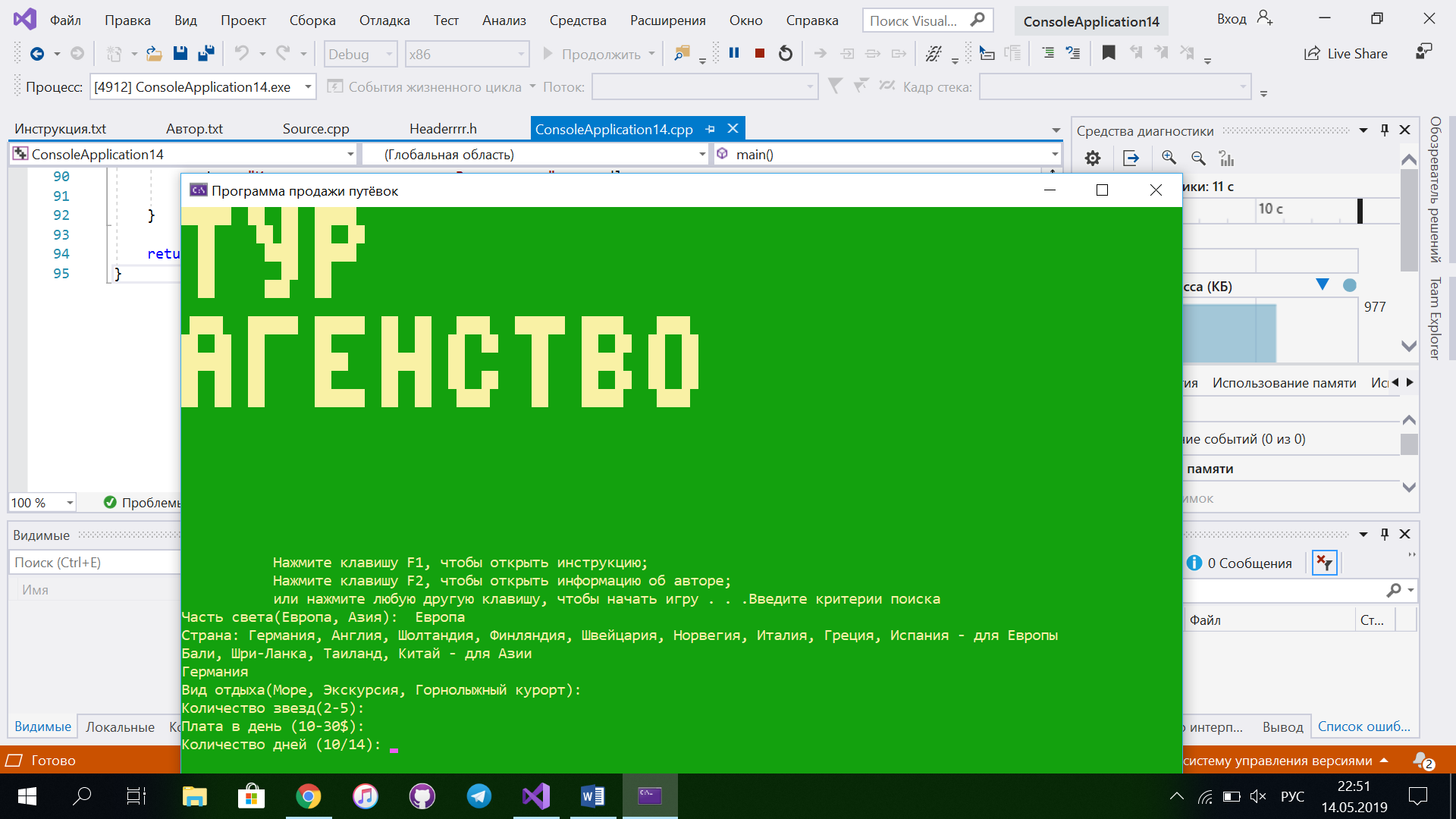


Рисунок 5 Ввод критериев поиска

После ввода всех критериев поиска пользователю предлагают 2 горящих тура и направления, найденные по его запросу. После чего спрашивают, желает ли он поискать ещё раз. При нажатии 1 начнется новый ввод критериев, а при нажатии 0 пользователя попросят выбрать номер путевки, которая ему больше всего приглянулась.

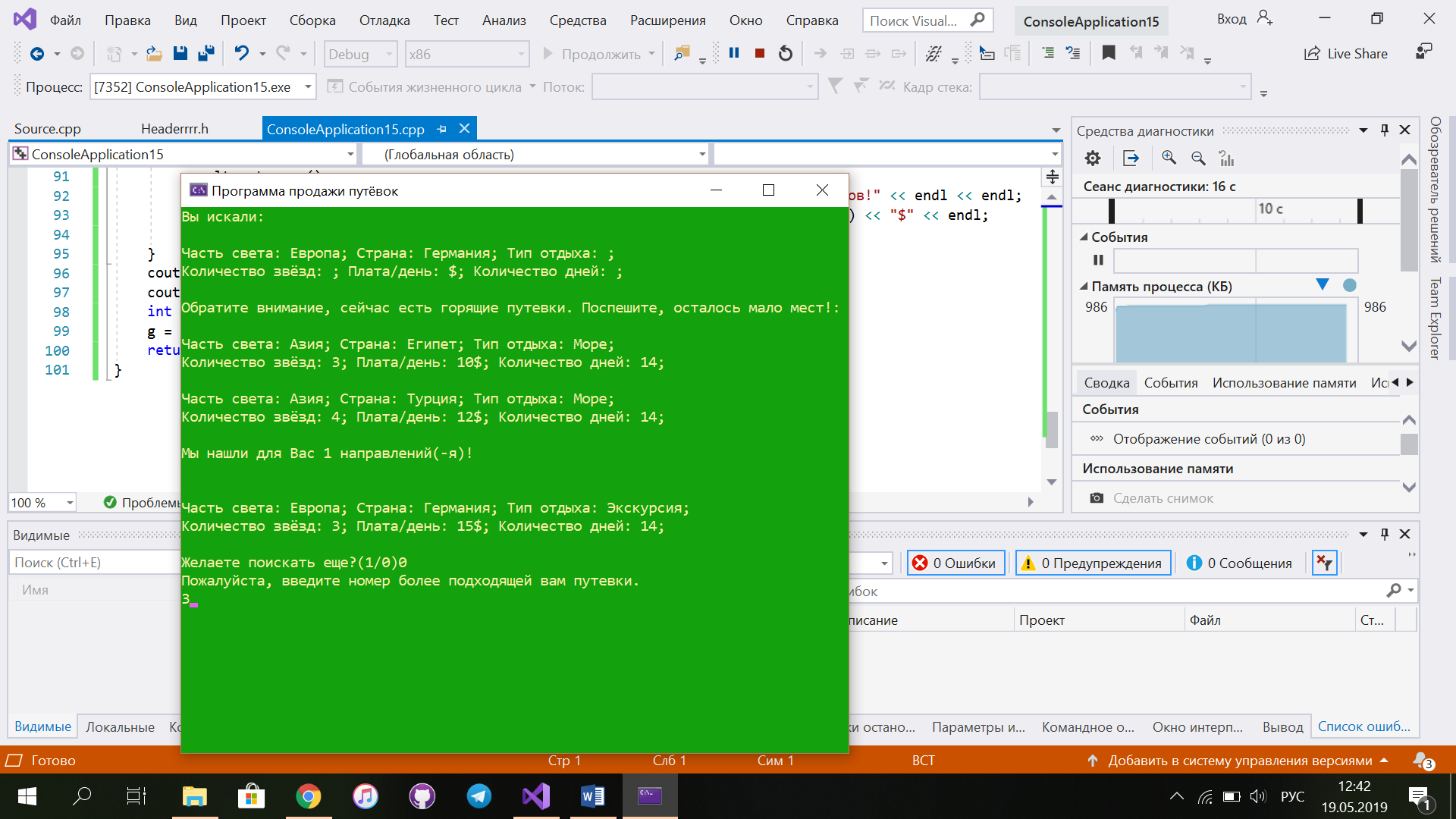


Рисунок 6 Вывод наиболее подходящих путевок и горящих туров

Далее последует вопрос «Вы постоянный клиент?» и при положительном ответе (т.е. вводе 1) ему посчитают стоимость путевки с 10% скидкой.

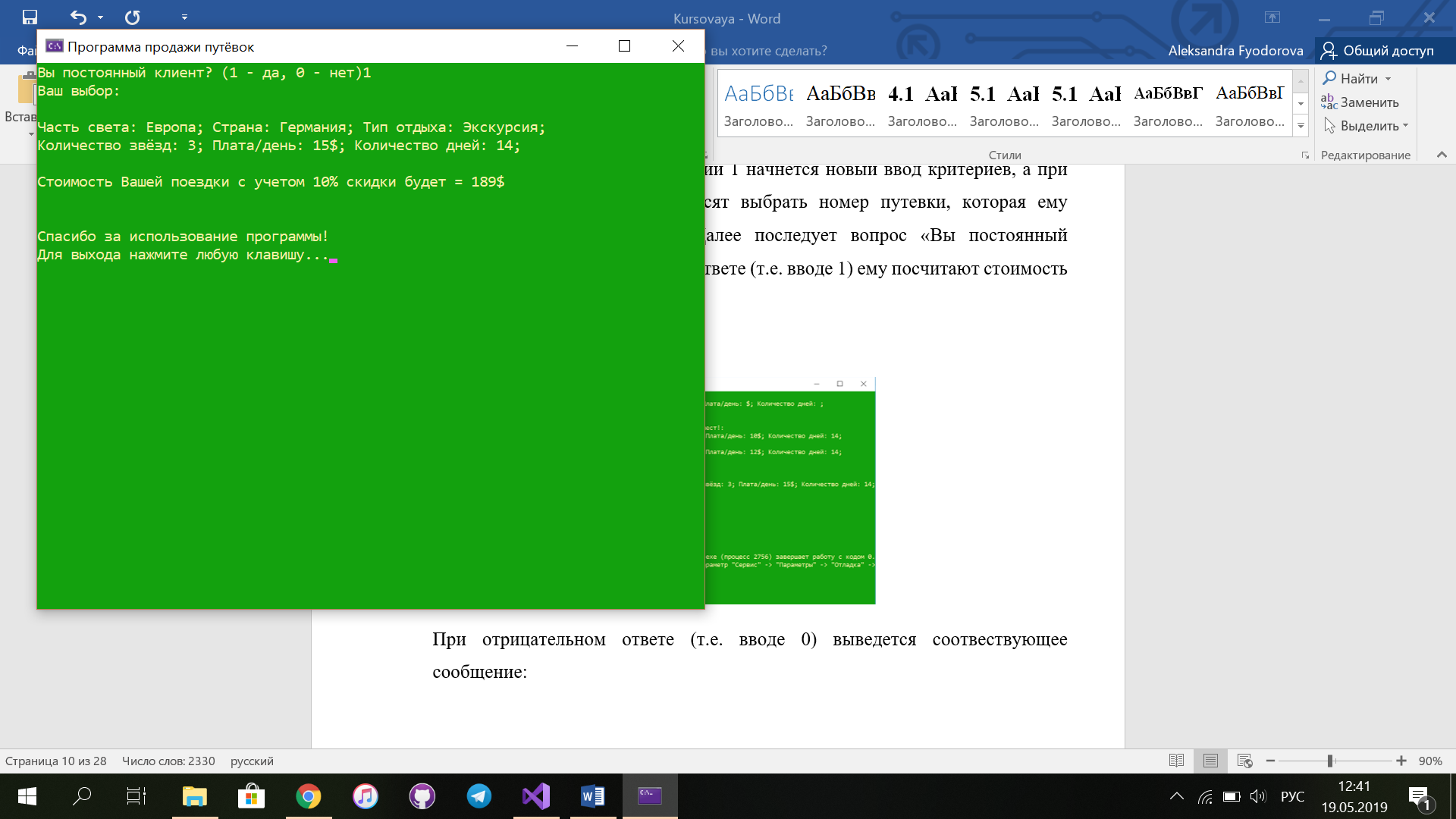


Рисунок 7 Случай расчета суммы для постоянного клиента

При отрицательном ответе (т.е. вводе 0) выведется соотвествующее сообщение:

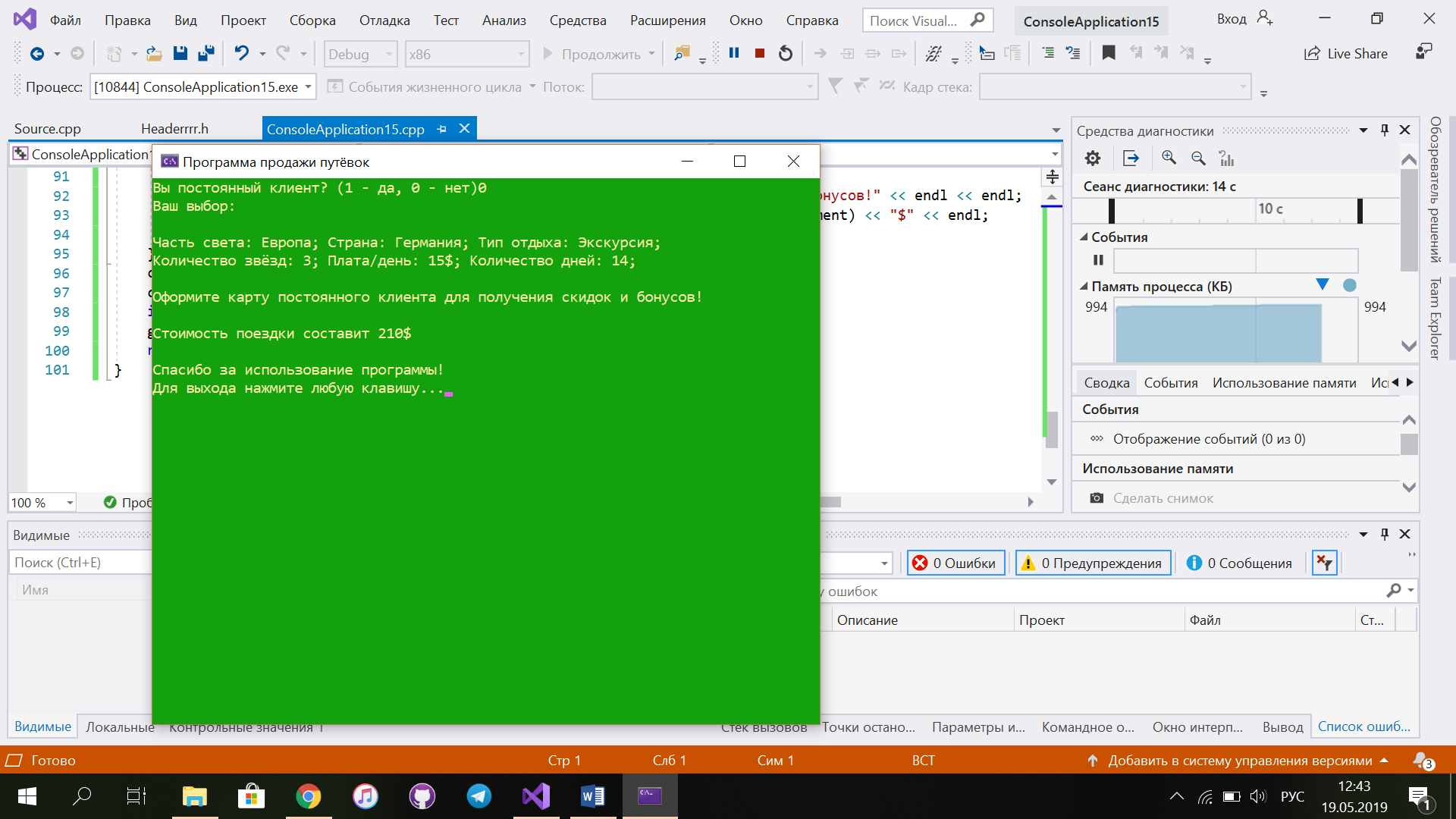


Рисунок 8 Случай расчета суммы для непостоянного клиента

# **7. ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы была разработана и реализована программа для продажи путевок клиентам. Предоставлена возможность поиска по заданным критериям. Каждый пользователь получает предложение в виде горящих путевок. Для постоянных клиентов предусмотрена скидка. Также есть возможность повторного поиска.

В результате создания данной программы были применены пройденные за этот семестр знания языка C++ и приобретены важнейшие навыки по разработке решений для различных задач.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Лекции Зориной Н.В. по ООП, 2019г.
2. https://www.sololearn.com/
3. <http://www.cyberforum.ru>
4. https://en.cppreference.com
5. <https://wikipedia.org>
6. Методические указания по выполнению практических работ для студентов, обучающихся по направлению подготовки 09.03.02/ Составитель Н. В. Зорина, Л. Б. Зорин, О.В. Соболев/ Редактор А. А. Хлебников
7. § Методические указания по выполнению лабораторных работ для студентов, обучающихся по направлению подготовки 09.03.02/ Составитель Н. В. Зорина, Л. Б. Зорин, О.В. Соболев/ Редактор А. А. Хлебников

# **Приложение А**

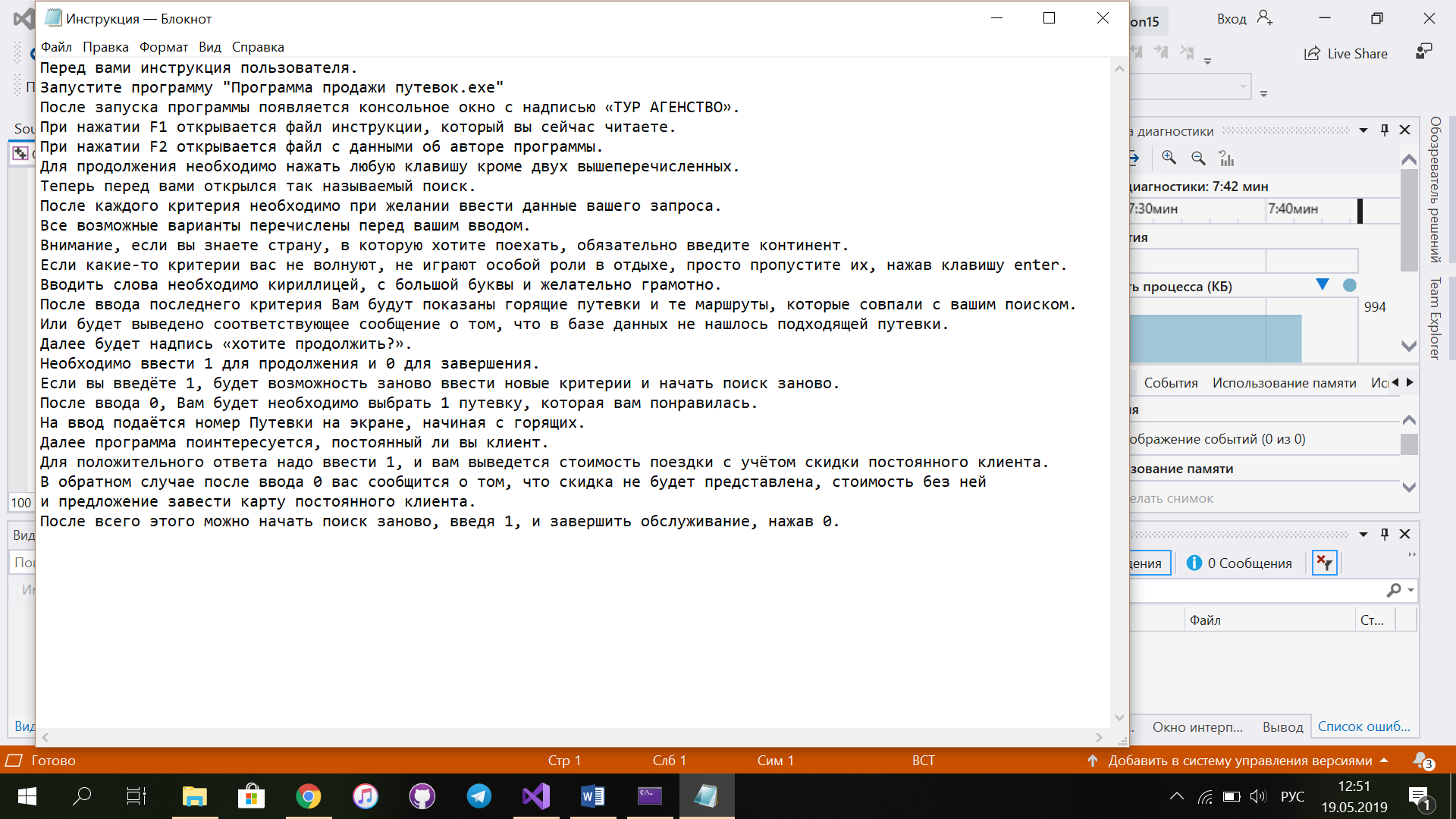


Рисунок 9 Файл с инструкцией пользователя

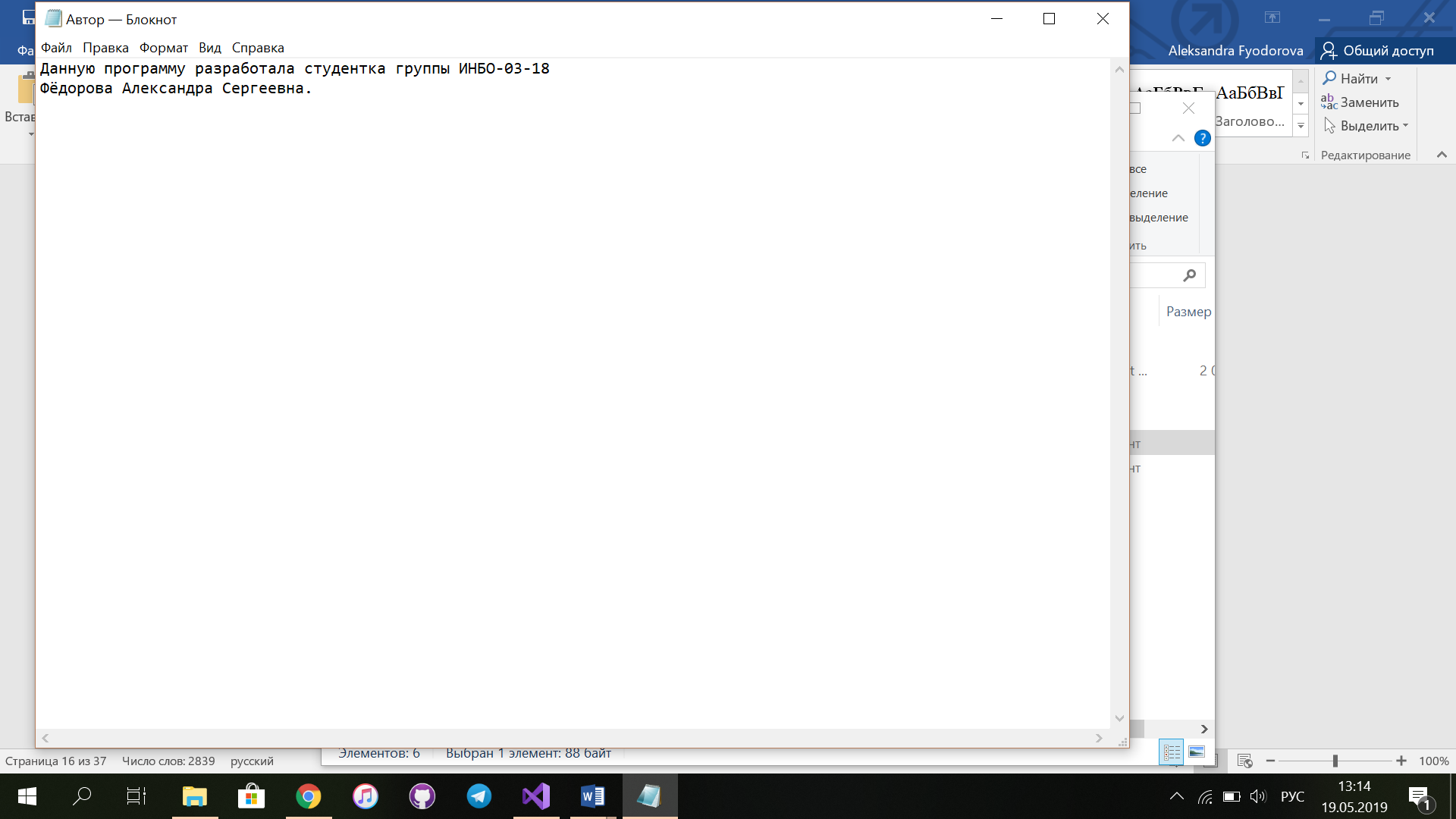


Рисунок 10 Файл с данными о студенте

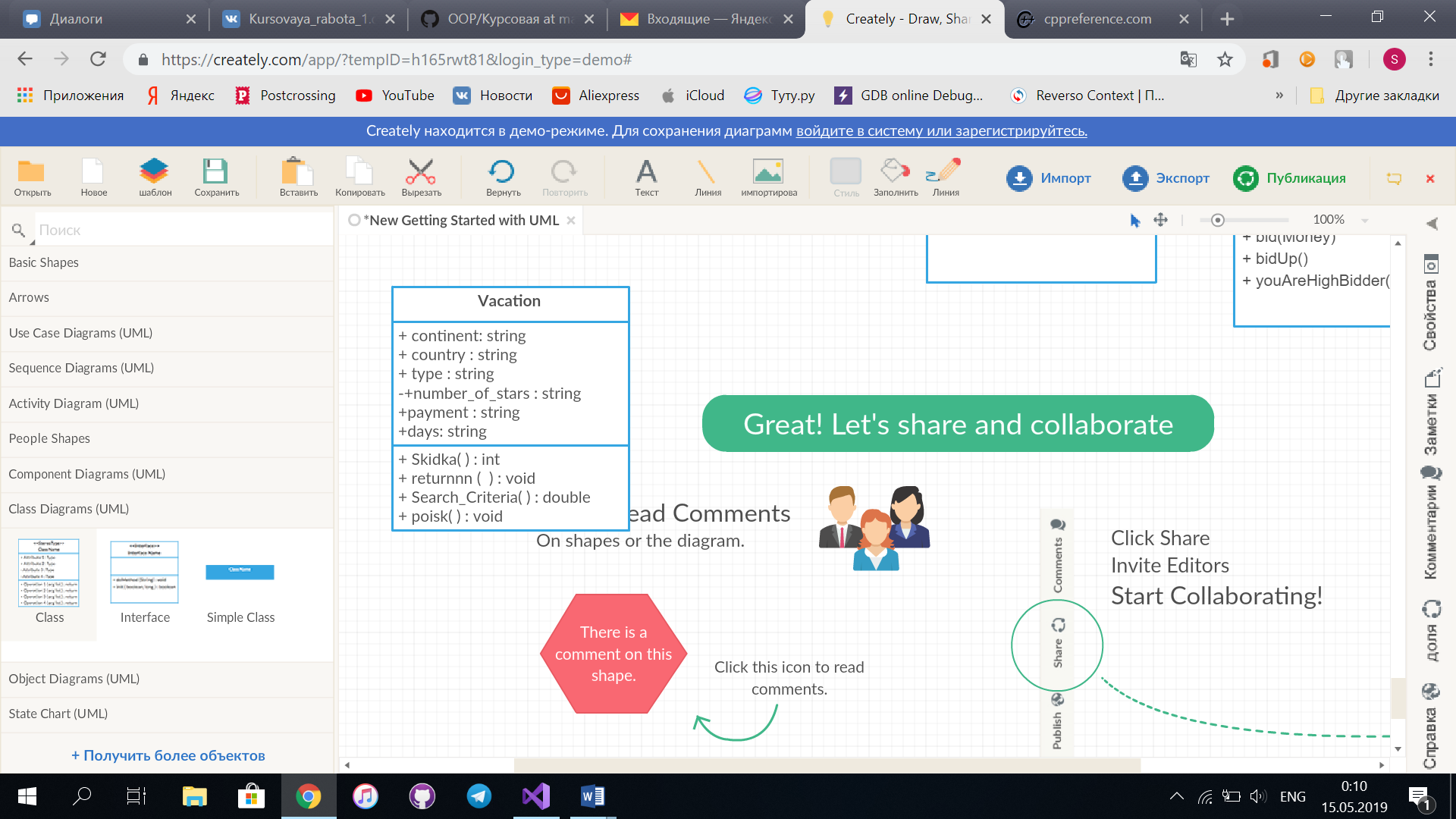


Рисунок 11 UML – диаграмма «Vacation»

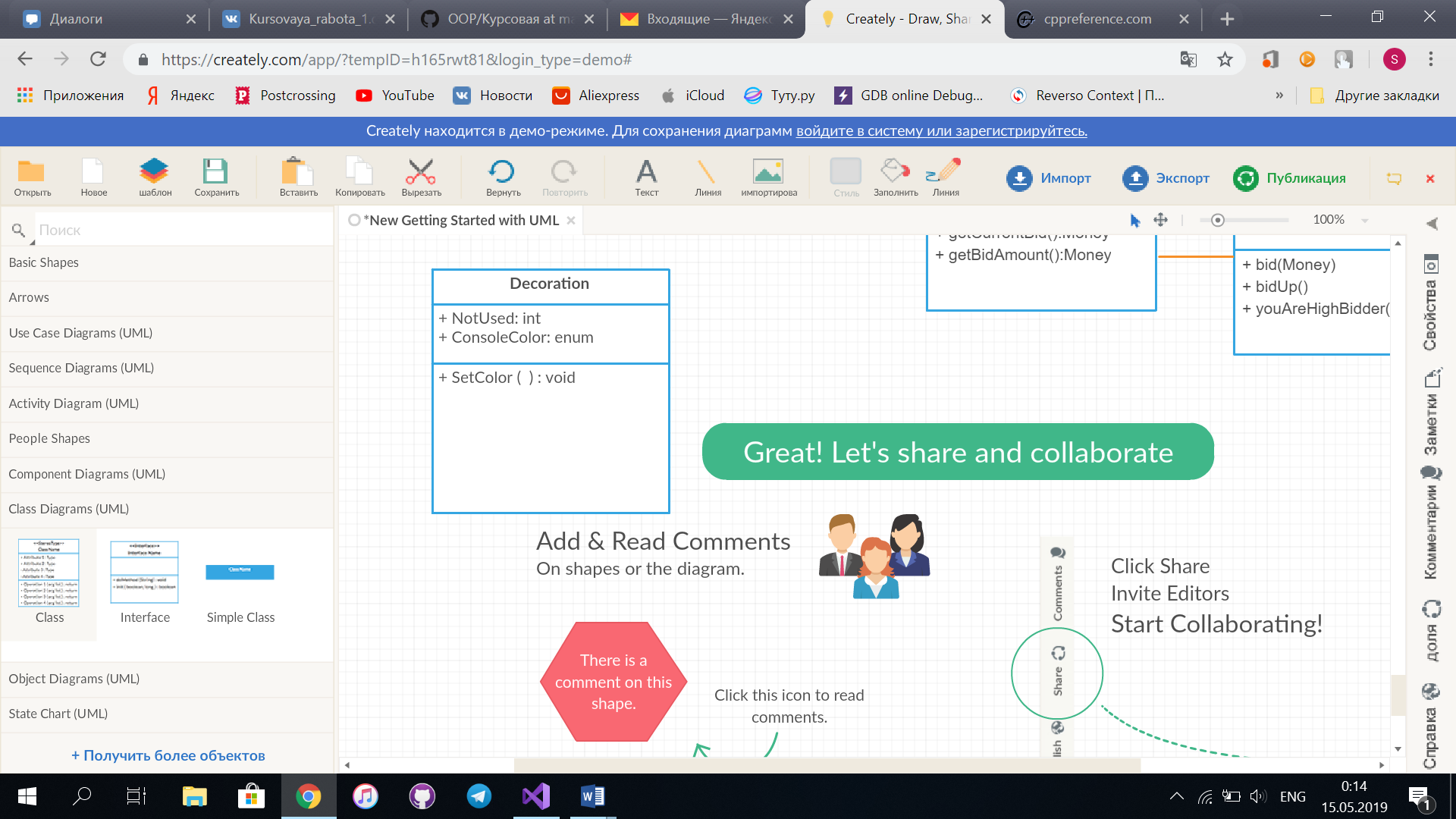


Рисунок 12 UML – диаграмма «Decoration»

# **Приложение Б**

**Main.cpp:**

#include "Headerrrr.h"

int main()

{

system("title Программа продажи путёвок");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Rus");

system("mode con cols=80 lines=30");

Decoration d;

d.hello();

d.setCursorPosition(0, 19); //вывод фраз с указаниями на главный экран

cout << " Нажмите клавишу F1, чтобы открыть инструкцию;" << endl;

cout << " Нажмите клавишу F2, чтобы открыть информацию об авторе;" << endl;

cout << " или нажмите любую другую клавишу, чтобы начать поиск . . .";

int c;

while (true) {

c = \_getch();

if (c == 0) {

c = \_getch();

if (c == 59) {

system("Инструкция.txt");

}

else if (c == 60) {

system("Автор.txt");

}

}

else break;

}

system("cls");

Vacation a[16]; // «база данных» путевок

a[0] = { "Европа", "Германия", "Экскурсия", "3", "15", "14" };

a[1] = { "Европа", "Англия", "Экскурсия", "4", "25", "10" };

a[2] = { "Европа", "Шотландия", "Экскурсия", "2", "10", "14" };

a[3] = { "Европа", "Финляндия", "Горнолыжный курорт", "5", "30", "10" };

a[4] = { "Европа", "Швейцария", "Горнолыжный курорт", "5", "30", "14" };

a[5] = { "Европа", "Норвегия", "Горнолыжный курорт", "4", "25", "14" };

a[6] = { "Азия", "Бали", "Море", "3", "10", "14" };

a[7] = { "Азия", "Шри-Ланка", "Море", "5", "20", "10" };

a[8] = { "Азия", "Таиланд", "Море", "4", "20", "14" };

a[9] = { "Азия", "Китай", "Экскурсия", "4", "25", "14" };

a[10] = { "Европа", "Италия", "Море", "5", "20", "14" };

a[11] = { "Европа", "Италия", "Экскурсия", "3", "25", "10" };

a[12] = { "Европа", "Греция", "Море", "4", "10", "14" };

a[13] = { "Европа","Греция", "Экскурсия", "3", "20", "10" };

a[14] = { "Европа", "Испания", "Море", "5", "20", "14" };

a[15] = { "Европа", "Испания", "Экскурсия", "4", "25", "10" };

Vacation b[2];

b[0] = { "Азия","Египет","Море","3","10","14" };

b[1] = { "Азия","Турция","Море","4","12","14" };

vector <Vacation> res;

while (true) {

res = poisk(a, b);

cout << endl << "Желаете поискать еще?(1/0)";

int ans;

cin >> ans;

if (ans == 0) break;

string s1; //очищаем последующие поиски от ентера, чтобы часть света не скипалась

getline(cin, s1);

system("cls");

}

cout << "Пожалуйста, введите номер более подходящей вам путевки." << endl;//вот тут пользователь

//должен выбрать 1 путевку по которой потом считается скидка Skidka() если он постоянный пользователь

int num = 0;

cin >> num;

Vacation result;

result = res[num - 1];

system("cls");

cout << "Вы постоянный клиент? (1 - да, 0 - нет)";

int post;

cin >> post;

switch (post) {

case 1:

cout << "Ваш выбор:" << endl;

result.returnnn();

cout << endl << "Стоимость Вашей поездки с учетом 10% скидки будет = " << result.Skidka() << "$" << endl << endl;

break;

case 0:

cout << "Ваш выбор:" << endl;

result.returnnn();

cout << endl << "Оформите карту постоянного клиента для получения скидок и бонусов!" << endl << endl;

cout << "Стоимость поездки составит " << stoi(result.days) \* stoi(result.payment) << "$" << endl;

break;

}

cout << endl << "Спасибо за использование программы!" << endl;

cout << "Для выхода нажмите любую клавишу...";

int g;

g = \_getch();

return 0;

}

**Source.cpp:**

#include "Headerrrr.h"

void Decoration::setCursorPosition(int x, int y)

{

static const HANDLE hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

std::cout.flush();

COORD coord = { (SHORT)x, (SHORT)y };

SetConsoleCursorPosition(hOut, coord);

}

void Decoration::SetColor(int text, ConsoleColor background)

{

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

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

void Decoration::x(int x, int y) {

setCursorPosition(x, y);

cout << "||";

}

void Decoration::hello() { //пишется надпись «тур агенство» вначале

SetColor(Yellow, Yellow);

x(0, 0); x(2, 0); x(4, 0); x(8, 0); x(12, 0); x(16, 0); x(18, 0);

x(19, 0);

x(2, 1); x(9, 1), x(12, 1); x(16, 1); x(20, 1);

x(2, 2); x(10, 2); x(12, 2); x(16, 2); x(18, 2); x(19, 2);

x(2, 3); x(12, 3); x(16, 3);

x(2, 4); x(10, 4); x(11, 4); x(16, 4);

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

x(1, 6); x(3, 6); x(8, 6); x(10, 6); x(12, 6); x(16, 6); x(18, 6); x(20, 6);

x(24, 6); x(28, 6); x(33, 6); x(35, 6); x(40, 6);

x(42, 6); x(44, 6); x(48, 6); x(50, 6); x(51, 6);

x(57, 6); x(59, 6);

x(0, 7); x(4, 7); x(8, 7); x(16, 7), x(24, 7); x(28, 7);

x(32, 7); x(36, 7); x(42, 7); x(48, 7); x(52, 7);

x(56, 7); x(60, 7);

x(0, 8); x(2, 8); x(4, 8); x(8, 8); x(16, 8); x(18, 8);

x(24, 8); x(26, 8); x(28, 8); x(32, 8); x(42, 8);

x(48, 8); x(50, 8); x(51, 8); x(56, 8); x(60, 8);

x(0, 9); x(4, 9); x(8, 9); x(16, 9); x(24, 9); x(28, 9);

x(32, 9); x(36, 9); x(42, 9); x(48, 9); x(52, 9);

x(56, 9); x(60, 9);

x(0, 10); x(4, 10); x(8, 10); x(16, 10); x(18, 10); x(20, 10);

x(24, 10); x(28, 10); x(33, 10); x(35, 10); x(42, 10);

x(48, 10); x(50, 10); x(51, 10); x(57, 10); x(59, 10);

cout << endl << endl << endl;

SetColor(Yellow, Green);

}

//вывод поисковых вопросов для пользователя

vector <Vacation> poisk(Vacation a[16], Vacation b[2]) {

vector <Vacation> v1;

vector <Vacation> v2;

v2.push\_back(b[0]);

v2.push\_back(b[1]);

Vacation v = {};

string s = "";

cout << "Введите критерии поиска" << endl;

cout << "Часть света(Европа, Азия): ";

getline(cin, s);

v.continent = s;

cout << "Страна: Германия, Англия, Шолтандия, Финляндия, Швейцария, Норвегия, Италия, Греция, Испания - для Европы" << endl;

cout << "Бали, Шри-Ланка, Таиланд, Китай - для Азии" << "" << endl;

getline(cin, s);

v.country = s;

cout << "Вид отдыха(Море, Экскурсия, Горнолыжный курорт): ";

getline(cin, s);

v.type = s;

cout << "Количество звезд(2-5): ";

getline(cin, s);

v.number\_of\_stars = s;

cout << "Плата в день (10-30$): ";

getline(cin, s);

v.payment = s;

cout << "Количество дней (10/14): ";

getline(cin, s);

v.days = s;

// сортировка введенных данных

for (int i = 0; i < 16; i++) {

if (a[i].Search\_Criteria(v) >= 0.8) {

v1.push\_back(a[i]);

}

}

int h = v1.size();

for (int j = 0; j < h; j++) {

int max = -1;

int max\_ind = -1;

for (int i = 0; i < v1.size(); i++) {

if (v1[i].Search\_Criteria(v) > max) {

max = v1[i].Search\_Criteria(v);

max\_ind = i;

}

}

v2.push\_back(v1[max\_ind]);

v1.erase(v1.begin() + max\_ind);

}

system("cls");

cout << "Вы искали: " << endl;

v.returnnn();

cout << endl;

cout << "Обратите внимание, сейчас есть горящие путевки. Поспешите, осталось мало мест!:" << endl;

b[0].returnnn(); b[1].returnnn();

cout << endl;

if (v2.size() == 2) {

cout << "Извините, по вашему запросу нет совпадений! Попробуйте начать заново." << endl;

cout << "Либо можете воспользоваться нашими горячими путевками!" << endl;

}

else {

cout << "Мы нашли для Вас " << v2.size() - 2 << " направлений(-я)!" << endl << endl;

for (int i = 2; i < v2.size(); i++) {

v2[i].returnnn();

}

}

return v2;

}

//функция подсчета скидки постоянного клиента

int Vacation::Skidka() {

int sale = stoi(payment) \* stoi(days) \* 0.9;

return sale;

}

// функция вывода подобранной путевки

void Vacation::returnnn() {

cout << endl << "Часть света: " << continent << ";";

cout << " Страна: " << country << ";";

cout << " Тип отдыха: " << type << ";" << endl;

cout << "Количество звёзд: " << number\_of\_stars << ";";

cout << " Плата/день: " << payment << "$" << ";";

cout << " Количество дней: " << days << ";" << endl;

}

//функция «начисления» баллов за совпадение между критериями введенными пользователем и «базой данных» тур агенства

double Vacation::Search\_Criteria(Vacation v) {

int max = 0;

double rate = 0;

if (v.continent != "") max += 20;

if (v.continent == continent) {

rate += 20;

}

if (v.country != "") max += 10;

if (v.country == country) {

rate += 10;

}

if (v.type != "") max += 1;

if (v.type == type) {

rate++;

}

if (v.number\_of\_stars != "") max += 1;

if (v.number\_of\_stars == number\_of\_stars) {

rate++;

}

if (v.payment != "") max += 1;

if (v.payment == payment) {

rate++;

}

if (v.days != "") max += 1;

else if (v.days == days) {

rate++;

}

return rate / max;

}

**Headerrrr.h:**

#include <algorithm>

using namespace std;

const int NotUsed = system("color 2E");

const enum ConsoleColor

{

Black = 0,

Blue = 1,

Green = 2,

Cyan = 3,

Red = 4,

Magenta = 5,

Brown = 6,

LightGray = 7,

DarkGray = 8,

LightBlue = 9,

LightGreen = 10,

LightCyan = 11,

LightRed = 12,

LightMagenta = 13,

Yellow = 14,

White = 15

};

struct Vacation {

string continent;

string country;

string type;

string number\_of\_stars;

string payment;

string days;

int Skidka();

void returnnn();

double Search\_Criteria(Vacation v);

};

class Decoration {

public:

void setCursorPosition(int x, int y);

void SetColor(int text, ConsoleColor background);

void x(int x, int y);

void hello();

};

vector <Vacation> poisk(Vacation a[16], Vacation b[2]);

#endif // !HEADER\_H