МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образование

учреждение высшего образования

«Ижевский государственный технический университет имени М. Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М. Т. Калашникова»)

Институт непрерывного профессионального образования

Лабораторная работа №2

по дисциплине "Прикладное программирование"

(Вариант 2)

|  |  |
| --- | --- |
|  | Выполнил студент группы: 5-19-1 од  Тепляков Георгий Константинович  Принял: преподаватель  Еланцев Михаил Олегович |

Ижевск

2018

**Содержание**

|  |  |
| --- | --- |
| 1 Листинг программы  **1.1 Program.cs**  **1.2 Studio.cs**  **1.3 Сheck\_in.cs**  **1.4 DataSource.cs**  **1.5 MemoryDataSource.cs**  **1.6 Name\_client.cs**  **1.7 Nomer.cs**  **1.8 OnlineBooking.cs**  **1.9 TelephoneBooking.cs**  **1.10 ReportItem.cs** | **3**  **3**  **11**  **13**  **13**  **14**  **15**  **15**  **16**  **16**  **17** |
| 2 Контрольный пример | **18** |
|  |  |
|  |  |
|  |  |
|  |  |

1 Листинг программы

**1.1 Program.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

class Program

{

public static Studio Studio;

static void Main(string[] args)

{

Studio = new Studio(new MemoryDataSource());

int user = 0;

do

{

try

{

Menu();

user = int.Parse(Console.ReadLine());

switch (user)

{

case 1:

Add();

break;

case 2:

Change();

break;

case 3:

Delete();

break;

case 4:

Look();

break;

case 5:

Report();

break;

case 0:

Console.WriteLine(" выход");

break;

default:

Console.WriteLine(" Такого раздела нет!");

break;

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

while (user != 0);

}

/// Добавить ----------------------------------------

static void Add()

{

Console.WriteLine(" Выберите метод регистрации:");

Console.WriteLine(" 1 - онлаин регистрация");

Console.WriteLine(" 2 - регистрация по телефону");

Console.WriteLine(" 0 - Выход в главное меню \n");

Console.Write("> ");

Check\_in check = null;

int type = int.Parse(Console.ReadLine());

if (type == 1)

{

check = new TelephoneBooking();

}

else if (type == 2)

{

check = new OnlineBooking();

}

else if (type == 0)

{

throw new Exception(" выход");

}

else

{

if (type != 1 || (type != 2) || (type != 0))

throw new Exception(" Такого раздела нет!");

}

type = 0;

if (check is TelephoneBooking)

{

Console.WriteLine(" Введите номер телефона(телефон должен содеражать 11 цифр):");

Console.Write("> ");

((TelephoneBooking)check).phone = (Console.ReadLine());

var reg = ((TelephoneBooking)check).phone;

if (reg.Length < 11 || reg.Length > 11)

{

throw new Exception(" Ошибка: телефон введен неверно ");

}

}

else if (check is OnlineBooking)

{

Console.WriteLine(" Введите email (email может содержать от 8 до 21 символа:):");

Console.Write("> ");

((OnlineBooking)check).email = (Console.ReadLine());

var reg = ((OnlineBooking)check).email;

if (reg.Length < 8 || reg.Length > 20)

{

throw new Exception(" Ошибка: email введен неверно");

}

}

var nam = new Name\_client();

Console.WriteLine(" ");

Console.WriteLine(" Введите имя");

Console.Write("> ");

nam.name = (Console.ReadLine());

if (nam.name.Length < 1)

{

throw new Exception(" Имя отсутствует!");

}

check.ime = nam;

Console.WriteLine(" ");

Console.WriteLine(" Введите фамилию");

Console.Write("> ");

nam.famil = (Console.ReadLine());

if (nam.famil.Length < 1)

{

throw new Exception(" Фамилия отсутствует!");

}

check.ime = nam;

var data = new Nomer();

Console.WriteLine(" ");

Console.WriteLine(" Выберите номер:");

Console.WriteLine(" 1 - Junior Suite (500р)");

Console.WriteLine(" 2 - Family Room (1000р)");

Console.WriteLine(" 3 - Standart (1200р)");

Console.WriteLine(" 4 - President Villa (2500р)");

Console.WriteLine(" 0 - Выход в главное меню");

Console.WriteLine(" ");

Console.Write("> ");

data.room\_Number = int.Parse(Console.ReadLine());

if (data.room\_Number == 0)

{

throw new Exception(" выход");

}

else if (data.room\_Number == 1)

{

data.price\_Number = +1000;

}

else if (data.room\_Number == 2)

{

data.price\_Number = +1200;

}

else if (data.room\_Number == 3)

{

data.price\_Number = +1600;

}

else if (data.room\_Number == 4)

{

data.price\_Number = +2500;

}

else if (data.room\_Number != 0 || data.room\_Number != 1 || data.room\_Number != 2 || data.room\_Number != 3 || data.room\_Number != 4)

{

throw new Exception(" Такого раздела нет! ");

}

Console.WriteLine(" ");

Console.WriteLine(" Введите дату (1 января 2000 или 1.1.2000 или 1 1 2000 )");

Console.Write("> ");

data.date = Convert.ToDateTime(Console.ReadLine());

check.nome = data;

Studio.Save(check);

}

/// Изменение записи ----------------------------------------

static void Change()

{

Console.WriteLine("Введите id: ");

Console.WriteLine("> ");

int id = int.Parse(Console.ReadLine());

Console.WriteLine(" Выберите метод регистрации:");

Console.WriteLine(" 1 - онлаин регистрация");

Console.WriteLine(" 2 - регистрация по телефону");

Console.WriteLine(" 0 - Выход в главное меню \n");

Console.Write("> ");

Check\_in check = null;

int type = int.Parse(Console.ReadLine());

if (type == 1)

{

check = new TelephoneBooking();

}

else if (type == 2)

{

check = new OnlineBooking();

}

else if (type == 0)

{

throw new Exception(" выход");

}

else

{

if (type != 1 || (type != 2) || (type != 0))

throw new Exception(" Такого раздела нет!");

}

type = 0;

if (check is TelephoneBooking)

{

Console.WriteLine(" Введите номер телефона(телефон должен содеражать 11 цифр):");

Console.Write("> ");

((TelephoneBooking)check).phone = (Console.ReadLine());

var reg = ((TelephoneBooking)check).phone;

if (reg.Length < 11 || reg.Length > 11)

{

throw new Exception(" Ошибка: телефон введен неверно ");

}

}

else if (check is OnlineBooking)

{

Console.WriteLine(" Введите email (email может содержать от 8 до 21 символа:):");

Console.Write("> ");

((OnlineBooking)check).email = (Console.ReadLine());

var reg = ((OnlineBooking)check).email;

if (reg.Length < 8 || reg.Length > 20)

{

throw new Exception(" Ошибка: email введен неверно");

}

}

var nam = new Name\_client();

Console.WriteLine(" ");

Console.WriteLine(" Введите имя");

Console.Write("> ");

nam.name = (Console.ReadLine());

if (nam.name.Length < 1)

{

throw new Exception(" Имя отсутствует!");

}

check.ime = nam;

Console.WriteLine(" ");

Console.WriteLine(" Введите фамилию");

Console.Write("> ");

nam.famil = (Console.ReadLine());

if (nam.famil.Length < 1)

{

throw new Exception(" Фамилия отсутствует!");

}

check.ime = nam;

var data = new Nomer();

Console.WriteLine(" ");

Console.WriteLine(" Выберите номер:");

Console.WriteLine(" 1 - Junior Suite (500р)");

Console.WriteLine(" 2 - Family Room (1000р)");

Console.WriteLine(" 3 - Standart (1200р)");

Console.WriteLine(" 4 - President Villa (2500р)");

Console.WriteLine(" 0 - Выход в главное меню");

Console.WriteLine(" ");

Console.Write("> ");

data.room\_Number = int.Parse(Console.ReadLine());

if (data.room\_Number == 0)

{

throw new Exception(" выход");

}

else if (data.room\_Number == 1)

{

data.price\_Number = +1000;

}

else if (data.room\_Number == 2)

{

data.price\_Number = +1200;

}

else if (data.room\_Number == 3)

{

data.price\_Number = +1600;

}

else if (data.room\_Number == 4)

{

data.price\_Number = +2500;

}

else if (data.room\_Number != 0 || data.room\_Number != 1|| data.room\_Number != 2|| data.room\_Number != 3|| data.room\_Number != 4)

{

throw new Exception(" Такого раздела нет! ");

}

Console.WriteLine(" ");

Console.WriteLine(" Введите дату (1 января 2000 или 1.1.2000 или 1 1 2000 )");

Console.Write("> ");

data.date = DateTime.Parse(Console.ReadLine());

check.nome = data;

check.id = id;

var che = Studio.Change(id, check);

if (che == null) throw new NotImplementedException();

}

/// Удалить ----------------------------------------

static void Delete()

{

Console.WriteLine(" Введите id для удаления: ");

Console.Write("> ");

int id = int.Parse(Console.ReadLine());

var act = Studio.Delete(id);

if (act == false) throw new NotImplementedException();

}

///Посмотр списка ----------------------------------------

static void Look()

{

var check = Studio.GetList();

foreach (var chec in check)

{

Console.Write(" id:");

Console.Write(chec.id);

Console.Write(" ");

Console.Write(" Дата заезда:");

Console.Write (chec.nome.date.ToString("d MMMM yyyy"));

Console.Write(" ");

Console.Write(" №\_номера:");

Console.Write(chec.nome.room\_Number);

Console.Write(" ");

Console.Write("+");

Console.Write(chec.nome.price\_Number);

Console.Write("р");

Console.Write(" ");

if (chec is TelephoneBooking)

{

Console.Write(" номер\_телефона:");

Console.Write(((TelephoneBooking)chec).phone);

}

else if (chec is OnlineBooking)

{

Console.Write(" email:");

Console.Write(((OnlineBooking)chec).email);

}

Console.Write(" ");

Console.Write(" Имя:");

Console.Write(chec.ime.name);

Console.Write(" ");

Console.Write(" Фамилия:");

Console.Write(chec.ime.famil);

Console.Write("\n");

}

}

/// Отчет ---------------------------------------- --------------------------

static void Report()

{

Console.WriteLine(" 1 - Доход от номерного фонда за отчетный период по дням");

Console.WriteLine(" 2 - Доход от номерного фонда за все время");

Console.WriteLine(" 0 - Выход в главное меню");

Console.Write("> ");

int type = int.Parse(Console.ReadLine());

if (type == 0)

{

throw new Exception(" выход");

}

else if (type == 2)

{

var report2 = Studio.Report2(null);

}

else if (type == 1)

{

var dataRep = new ReportItem();

Console.WriteLine(" Введите дату за какой период нужно сформировать отчет (1 января 2000 или 1.1.2000 или 1 1 2000 )");

Console.Write("> ");

dataRep.dateRep = DateTime.Parse(Console.ReadLine());

var report2 = Studio.Report2(dataRep.dateRep);

}

else if (type != 2 || type != 1)

{

Console.WriteLine("Ошибка: тип не выбран");

}

Console.WriteLine(" Отчет готов");

/// Меню ----------------------------------------

public static void Menu()

{

Console.WriteLine("--------------------------------------------------------------");

Console.WriteLine("Учет\_номеров\_в\_гостинице");

Console.WriteLine(" ");

Console.WriteLine("Выберите раздел:");

Console.WriteLine("1.Добавить");

Console.WriteLine("2.Изменение записи ");

Console.WriteLine("3.Удалить ");

Console.WriteLine("4.Посмотреть записи");

Console.WriteLine("5.Отчет");

Console.WriteLine("0.Выход");

Console.WriteLine(" ");

Console.Write("> ");

}

}

}

**1.2 Studio.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class Studio

{

private DataSource dataSource;

public Studio(DataSource data\_source)

{

dataSource = data\_source;

}

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

public List<Check\_in> GetList()

{

var check = dataSource.GetAll();

check.Sort((a, b) =>

{

if (a.nome.date < b.nome.date) return -1;

else if (a.nome.date > b.nome.date) return 1;

else

{

if (a.nome.room\_Number < b.nome.room\_Number) return -1;

else if (a.nome.room\_Number > b.nome.room\_Number) return 1;

else return 0;

}

});

return check;

}

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

public bool Delete(int id)

{

if (id < 1)

{

throw new Exception(" Отсутствует id!");

}

return dataSource.Delete(id);

}

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

public Check\_in Change(int id, Check\_in check)

{

if (id < 1)

{

throw new Exception(" Отсутствует id!");

}

return dataSource.Get(id, check);

}

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

public Check\_in Save(Check\_in check)

{

return dataSource.Save(check);

}

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

public List<Check\_in> Report2(DateTime? dataRep = null)

{

var re = dataSource.GetAll();

string str = "";

int rep = 0;

foreach (Check\_in r in re)

{

if (dataRep == null || r.nome.date == dataRep.Value)

{

string rea = " ";

if (r is TelephoneBooking) rea = "номер\_телефона:" + ((TelephoneBooking)r).phone;

else if (r is OnlineBooking) rea = "email:" + ((OnlineBooking)r).email;

str = str + "+" + r.nome.price\_Number + " "

+ "Дата заезда:" + r.nome.date.ToString("d MMMM yyyy") + " "

+ "№\_номера:" + r.nome.room\_Number + " "

+ "Имя:" + r.ime.name + " "

+ "Фамилия:" + r.ime.famil + " "

+ rea + " "

+ "id:" + r.id + ' '

+ " " + Environment.NewLine;

rep += r.nome.price\_Number;

}

else if (r.nome.date != dataRep.Value)

{

throw new Exception(" Дата не найдена!");

}

}

str += Environment.NewLine + Environment.NewLine + "Итог за:" + " " + rep + "р";

System.IO.Directory.CreateDirectory("D:\\Прикладное\_программирование\\lab2");

System.IO.File.WriteAllText("D:\\Прикладное\_программирование\\lab2\\Report.txt", str);

return null;

}

}

}

**1.3 Сheck\_in.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public abstract class Check\_in

{

public int? id;

public Name\_client ime;

public Nomer nome;

abstract public Check\_in Copy();

}

}

**1.4 DataSource.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public abstract class DataSource

{

abstract public Check\_in Save(Check\_in record);

abstract public bool Delete(int id);

abstract public Check\_in Get(int id, Check\_in record);

abstract public List<Check\_in> GetAll();

}

}

**1.5 MemoryDataSource.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class MemoryDataSource : DataSource

{

private List<Check\_in> check\_ins = new List<Check\_in>();

private int idCounter = 0;

private int FindIndex(int id)

{

for (var i = 0; i < check\_ins.Count; i++)

{

if (check\_ins[i].id == id) return i;

}

return -1;

}

public override Check\_in Get(int id, Check\_in record)

{

var nw = record;

int ind = FindIndex(id);

check\_ins[ind] = nw;

return nw;

}

public override Check\_in Save(Check\_in record)

{

var copy = record.Copy();

copy.id = ++idCounter;

check\_ins.Add(copy);

return copy;

}

public override bool Delete(int id)

{

int ind = FindIndex(id);

if (ind != -1)

{

check\_ins.RemoveAt(ind);

return true;

}

return false;

}

public override List<Check\_in> GetAll()

{

var new\_list = new List<Check\_in>();

foreach (var check in check\_ins)

{

new\_list.Add(check.Copy());

}

return new\_list;

}

}

}

**1.6 Name\_client.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class Name\_client

{

public string name;

public string famil;

}

}

**1.7 Nomer.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class Nomer

{

public int room\_Number;

public DateTime date;

public int price\_Number;

}

}

**1.8 OnlineBooking.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class OnlineBooking : Check\_in

{

public string email;

public override Check\_in Copy()

{

return new OnlineBooking

{

id = id,

ime = ime,

nome = nome,

email = email,

};

}

}

}

**1.9 TelephoneBooking.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class TelephoneBooking : Check\_in

{

public string phone;

public override Check\_in Copy()

{

return new TelephoneBooking

{

id = id,

ime = ime,

nome = nome,

phone = phone,

};

}

}

}

**1.10 ReportItem.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Accounting\_for\_hotel\_rooms

{

public class ReportItem

{

public DateTime dateRep;

}

}

2 Контрольный пример

На рисунке 1 произведено добавление новой записи.

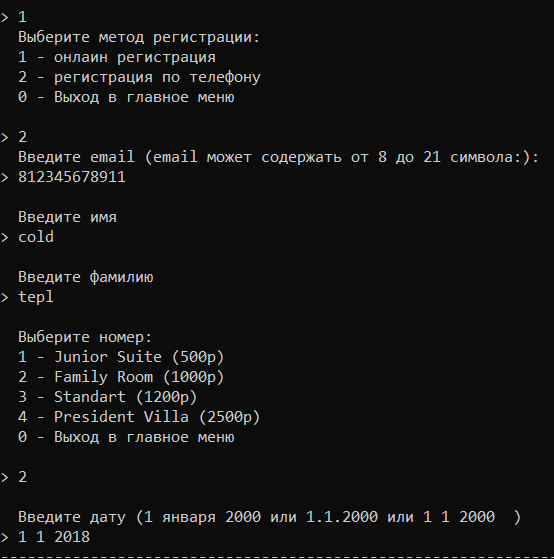


Рис. 1 – добавление записи

На рисунке 2 представлен вывод всех записей на текущий момент.

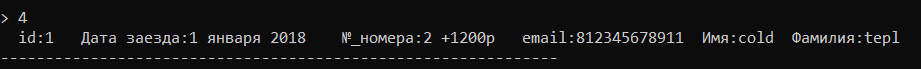


Рис. 2 – вывод всех записей на текущий момент

Аналогично добавим еще несколько записей. После добавления на рисунке 3 происходит сортировка по полям с датой заезда и №\_номера.

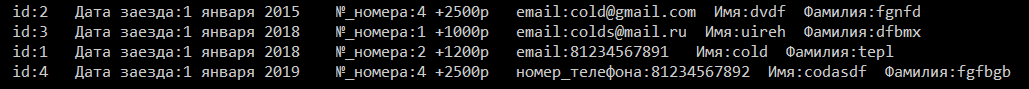


Рис. 3 – сортировка записей на текущий момент

На рисунке 4 показано изменение записи с id = 1.

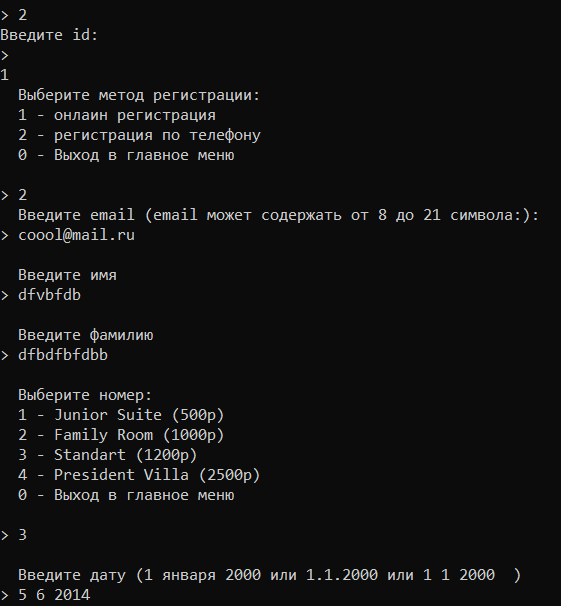


Рис. 4 – изменение записи с id = 1

На рисунке 5 показано запись до изменения и сравнение ее с рисунком 6 после изменение записи с id = 1.

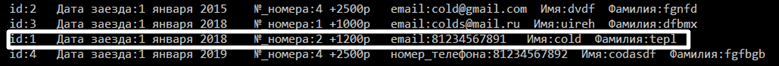


Рис. 5 – запись до изменения

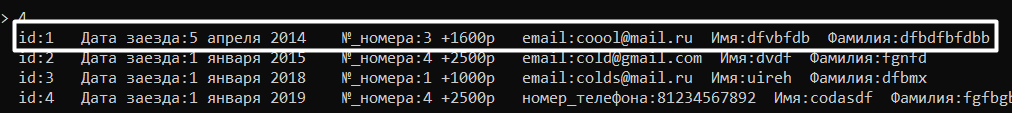


Рис. 6 – запись после изменение

На рисунке 7 показано удаление записи с id = 1.

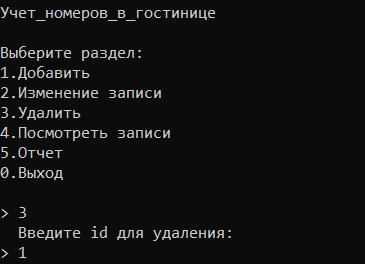


Рис. 7 – удаление записи

На рисунке 8 показан вывод записи после удаления с id = 1.

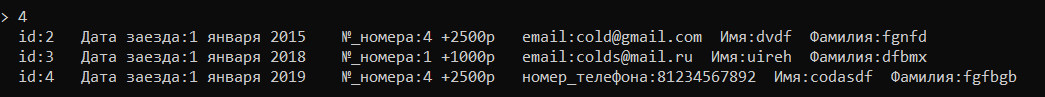


Рис. 8 – вид после удаления записи с id = 1

На рисунке 9 был создан отчёт за все время. Сам отчёт сохраняется в виде .txt файла, содержащий в себе все имеющиеся записи и подсчет их сумму.

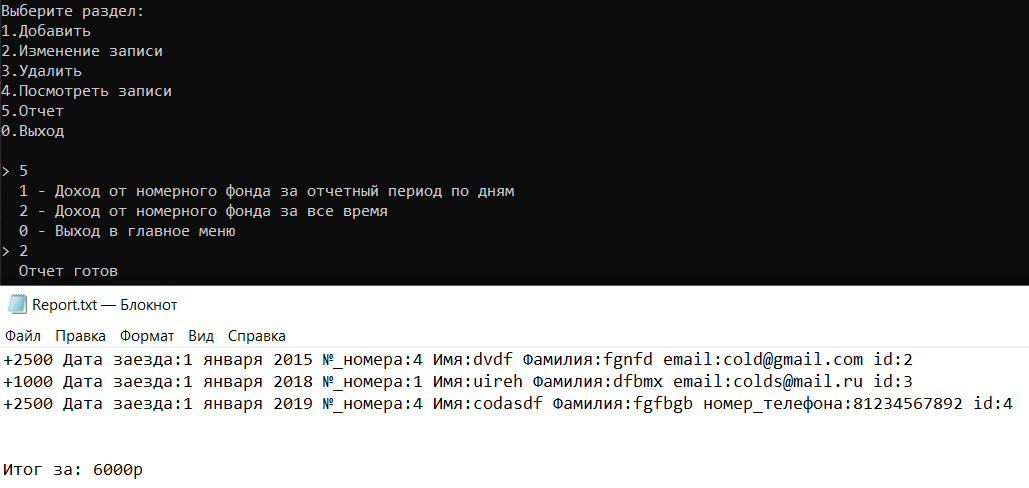


Рис. 9 – Создание отчёта и его вид в .txt файле за все время

На рисунке 10 был создан отчёт за отчетный период с заданной датой 1 января 2015. Сам отчёт сохраняется в виде .txt файла, и содержит записи за введённый отчетный период, а так же подсчет этой суммы за этот период.

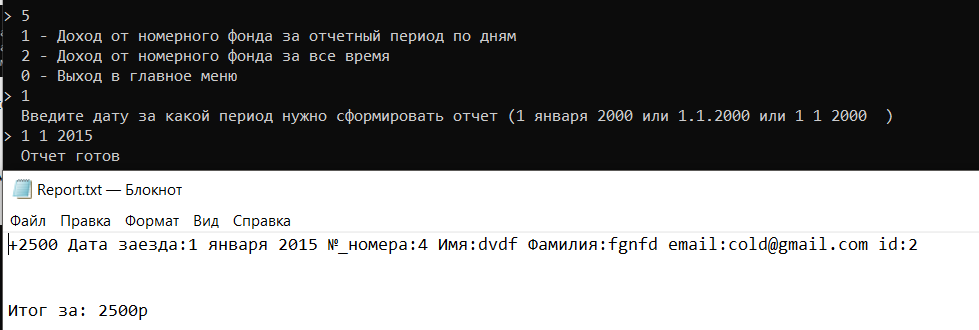


Рис. 10 – Создание отчёта и его вид в .txt файле за отчетный период