|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт перспективных технологий и индустриального программирования (ИПТИП)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | |
| **по дисциплине** | |
| «Технологии создания программного обеспечения» | |
| **Практическая работа №3** | |
| Выполнил студент группы ЭФМО-02-23 | Мурадов Н.Н. |

Москва 2023

**СОДЕРЖАНИЕ**

[Задача 3](#_Toc150790600)

[Решение 5](#_Toc150790601)

[Результаты 13](#_Toc150790602)

Задача

Создать приложение, которое позволит выполнять добавление, извлечение и поиск информации из XML-файла. XML-файл должен описывать ФИО сотрудника, год рождения, домашний адрес, телефон, сведения о работе (название должности, дата начала работы, дата окончания работы, отдел) и заработной плате (год, месяц, итого).

Предусмотреть возможность добавления и редактирования данных о сотруднике, должности и заработной плате.

**Приложение должно осуществлять:**

1. поиск сотрудника по фамилии, выводить историю о его трудовой деятельности (с возможностью сортировки по году начала работы) и начислениях заработной платы сотрудника (с возможностью указания периода) с расчетом максимального, минимального и среднего значения.
2. вывод по каждому отделу количества работающих сотрудников и списка должностей (без повторов). Для каждого отдела определить долю работающих сотрудников из общего контингента. Работающим считается сотрудник, у которого хотя бы для одной должности не указана дата увольнения.
3. вывод сотрудников, которые работают на текущий момент в более чем одном отделе. Для таких сотрудников указать, в каком отделе они получают бо́льшую зарплату.
4. вывод отделов, в которых работает не более 3 сотрудников.
5. вывод годов, в которых было принято и уволено наибольшее и наименьшие количество сотрудников.

**Используя данные XML-файлов:**

1. Построить график изменения курса валюты (пользователь имеет возможность указать диапазон дат и валюту)

*Пример получения динамики котировок доллара США:* <http://www.cbr.ru/scripts/XML_dynamic.asp?date_req1=02/01/2021&date_req> 2=20/02/2021&VAL\_NM\_RQ=R01235

date\_req1 — date\_req2 = Диапазон дат VAL\_NM\_RQ — Код валюты

*Справочник по кодам валют:* <http://www.cbr.ru/scripts/XML_val.asp?d=0>

1. Построить графики изменения курса каждого драгоценного металла (пользователь имеет возможность указать диапазон дат). Определить даты (из указанного диапазона), на которые стоимость каждого драгоценного металла была максимальной и минимальной.

*Пример получения динамики котировок драгоценных металлов:* <http://www.cbr.ru/scripts/xml_metall.asp?date_req1=01/01/2021&date_req2=0> 1/02/2021

date\_req1 — date\_req2 = Диапазон дат

Code = 1 – золото, 2 – серебро, 3 – платина, 4 – палладий

1. Определить санаторно-курортные учреждения, расположенные в Астраханской области

*Перечень санаторно-курортных учреждений (государственной, муниципальной и частной систем здравоохранения):*

https://minzdrav.gov.ru/opendata/7707778246-perechensanatornokurortnyhuchrezhdenij

Решение

*Листинг кода первой части(1-5 задачи) и второй части(1-3 задачи).*

using System;

using System.Linq;

using System.Collections.Generic;

using System.Runtime.Versioning;

using Bogus;

using System.Xml.Linq;

using System.Text;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

using System.Drawing;

public class Pract3 : Form {

private Faker fakerRu = new("ru");

private Random rnd = new();

private Chart myChart;

private Series mySeries1, mySeries2, mySeries3, mySeries4;

public void run() {

Console.WriteLine("Hello World Pract3");

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.InputEncoding = System.Text.Encoding.UTF8;

mainP2();

Console.WriteLine("Конец Hello World Pract3");

}

private void mainP1() {

XDocument doc = genData();

num1P1(doc);

}

private void mainP2() {

Encoding.RegisterProvider(CodePagesEncodingProvider.Instance);

num3P2();

}

private XElement genJob(List<String> jobList, bool nonexpired = false) {

XElement dataExp = nonexpired ? null : new XElement("Дата\_окончания", fakerRu.Date.PastDateOnly(20));

// Console.WriteLine(dataExp);

jobList.Add(fakerRu.Name.JobArea());

return new("Работа",

new XElement("Название", fakerRu.Name.JobTitle()),

new XElement("Дата\_начала", fakerRu.Date.PastDateOnly(100)),

dataExp,

new XElement("Отдел", jobList.Last())

);

}

private XDocument genData() {

XDocument doc;

if(File.Exists("pract3.xml")) {

doc = XDocument.Load("pract3.xml");

} else {

doc = new XDocument();

XElement dep = new("Департамент");

for(int i = 0; i < rnd.Next(15, 26); i++) {

XElement listJob = new("Список\_работ"),

listSalary = new("Список\_зарплат"),

pep = new("Сотрудник",

new XElement("ФИО", fakerRu.Name.LastName() + " " + fakerRu.Name.FirstName() + " -"),

new XElement("Год\_рождения", fakerRu.Date.PastDateOnly(100)),

new XElement("Домашний\_адрес", fakerRu.Address.FullAddress()),

new XElement("Телефон", fakerRu.Phone.PhoneNumber()),

listJob,

listSalary

);

bool unemployed = fakerRu.Random.Bool();

List<String> jobList = new();

for(int i1 = 0; i1 < rnd.Next(5, 10); i1++) {

listJob.Add(genJob(jobList));

}

if(unemployed) {

for(int i1 = 0; i1 < rnd.Next(1, 3); i1++) {

listJob.Add(genJob(jobList, true));

}

};

// if(unemployed) listJob.Add(genJob(true));

for(int i1 = 0; i1 < rnd.Next(5, 10); i1++) {

DateTime d = fakerRu.Date.Past(100);

listSalary.Add(new XElement("Зарплата",

new XElement("Год", d.Year),

new XElement("Месяц", d.Month),

new XElement("Отдел", fakerRu.Random.ListItem(jobList)),

new XElement("Размер", fakerRu.Finance.Amount(50000, 1000000, 0))

));

}

dep.Add(pep);

}

doc.Add(dep);

doc.Save("pract3.xml");

Console.WriteLine("Конец genData\n");

}

return doc;

}

private Series genGraph(string zag = "График") {

Text = zag;

ClientSize = new Size(1400, 600);

// MicrosoftChart - свойства

myChart = new Chart();

myChart.Parent = this;

myChart.Left = 10;

myChart.Top = 10;

myChart.Width = (ClientSize.Width - 20);

myChart.Height = (ClientSize.Height - 20);

// Область в которой будет построен график// (Их может быть несколько)

ChartArea myChartArea = new ();

myChartArea.Name = "myChartArea";

myChartArea.AxisY.IsStartedFromZero = false;

myChart.ChartAreas.Add(myChartArea);

// График (Их может быть несколько)

mySeries1 = new ();

mySeries1.ChartType = SeriesChartType.Spline;

mySeries1.ChartArea = "myChartArea";

myChart.Series.Add(mySeries1);

return mySeries1;

}

private void num1P1(XDocument doc) {

Console.WriteLine("num1P1:");

Console.WriteLine("Введите Фамилию(Пример | по умолчанию, 'Васильева'):");

string lastName = defConsole(Console.ReadLine(), "Васильева");

Console.WriteLine("Фамилию: " + lastName);

var rez = doc.Element("Департамент").Elements("Сотрудник").Where(e => e.Element("ФИО").Value.IndexOf(lastName) > -1)

.Select(e => new {listJob = e.Element("Список\_работ").Elements("Работа"),

listSalary = e.Element("Список\_зарплат").Elements("Зарплата")

})

.Select(e => new {listJob = string.Join(", \n", e.listJob.OrderBy(e => Int32.Parse(e.Element("Дата\_начала").Value.Split('.')[2]))),

listSalary = string.Join(", \n", e.listSalary),

maxSalary = e.listSalary.MaxBy(el => Int32.Parse(el.Element("Размер").Value)),

minSalary = e.listSalary.MinBy(el => Int32.Parse(el.Element("Размер").Value)),

avgSalary = e.listSalary.Average(el => Int32.Parse(el.Element("Размер").Value))

});

foreach (var n in rez) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num1P1\n");

}

private void num2P1(XDocument doc) {

Console.WriteLine("num2P1:");

var rez = doc.Elements("Департамент").Elements("Сотрудник").Elements("Список\_работ").Elements("Работа").Where(el => el?.Element("Дата\_окончания")?.Value == null).Distinct();

var rezG = rez.GroupBy(e => e.Element("Отдел").Value);

var rez1 = rezG.Select(e => new {name = e.Key, ratio = e.Count() / (float)rez.Count(), count = e.Count(), jobs = string.Join(", \n", e.Elements("Название").Distinct())});

Console.WriteLine("\n Данные: ");

foreach (var n in rezG) {

Console.WriteLine("\n key " + n.Key + "\n value");

foreach (var n1 in n)

Console.WriteLine(string.Join(", \n", n1.Element("Отдел")));

}

Console.WriteLine("\n Ответ: ");

foreach (var n in rez1) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num2P1\n");

}

private void num3P1(XDocument doc) {

Console.WriteLine("num3P1:");

var rez = doc.Elements("Департамент").Elements("Сотрудник")

.Where(el => el.Elements("Список\_работ").Elements("Работа")

.Count(e => e?.Element("Дата\_окончания")?.Value == null) > 1)

.Select(e => new {fio = e.Element("ФИО"), maxSalary = e.Elements("Список\_зарплат").Elements("Зарплата").MaxBy(el => el.Element("Размер").Value)});

Console.WriteLine("\n Данные: ");

foreach (var n in rez) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num3P1\n");

}

private void num4P1(XDocument doc) {

Console.WriteLine("num4P1:");

var rez = doc.Elements("Департамент").Elements("Сотрудник").Elements("Список\_работ").Elements("Работа").Where(el => el?.Element("Дата\_окончания")?.Value == null);

var rezG = rez.GroupBy(e => e.Element("Отдел").Value).Where(el => el.Count() < 4);

var rez1 = rezG.Select(e => new {name = e.Key, ratio = e.Count() / (float)rez.Count(), count = e.Count()});

Console.WriteLine("\n Данные: ");

foreach (var n in rezG) {

Console.WriteLine("\n key " + n.Key + "\n value");

foreach (var n1 in n)

Console.WriteLine(string.Join(", \n", n1.Element("Отдел")));

}

Console.WriteLine("\n Ответ: ");

foreach (var n in rez1) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num4P1\n");

}

private void num5P1(XDocument doc) {

Console.WriteLine("num5P1:");

var rezG = doc.Descendants("Работа")

.Select(e => new {id = 0, acc = e.Element("Дата\_начала").Value.Split(".")[2], dis = e?.Element("Дата\_окончания")?.Value?.Split(".")[2]})

.GroupBy(e => e.id,

(k, v) => new {accList = v.GroupBy(el => el.acc).Select(el => new {acc = el.Key, accCount = el.Count()}),

disList = v.GroupBy(el => el.dis).Where(el => el.Key != null).Select(el => new {dis = el.Key, disCount = el.Count()})});

var rez = rezG.Select(e => new {accMax = e.accList.MaxBy(el => el.accCount), accMin = e.accList.MinBy(el => el.accCount),

disMax = e.disList.MaxBy(el => el.disCount), disMin = e.disList.MinBy(el => el.disCount)});

Console.WriteLine("\n Данные: ");

foreach (var n in rezG) {

Console.WriteLine("\n accList: ");

foreach (var n1 in n.accList)

Console.WriteLine(n1);

Console.WriteLine("\n disList: ");

foreach (var n1 in n.disList)

Console.WriteLine(n1);

}

Console.WriteLine("\n Ответ: ");

foreach (var n in rez) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num5P1\n");

}

private string defConsole(string con, string def) {

return con == String.Empty ? def : con;

}

private void num1P2() {

genGraph("График");

Console.WriteLine("num1P2:");

Console.WriteLine("Введите date\_req1(Пример | по умолчанию, '02/01/2021'):");

string date\_req1 = defConsole(Console.ReadLine(), "02/01/2021");

Console.WriteLine("Введите date\_req2(Пример | по умолчанию, '20/02/2021'):");

string date\_req2 = defConsole(Console.ReadLine(), "20/02/2021");

Console.WriteLine("Введите VAL\_NM\_RQ(Пример | по умолчанию, 'R01235'):");

string VAL\_NM\_RQ = defConsole(Console.ReadLine(), "R01235");

// XDocument doc = XDocument.Load("http://www.cbr.ru/scripts/XML\_dynamic.asp?date\_req1=02/01/2021&date\_req2=20/02/2021&VAL\_NM\_RQ=R01235");

XDocument doc = XDocument.Load($"http://www.cbr.ru/scripts/XML\_dynamic.asp?date\_req1={date\_req1}&date\_req2={date\_req2}&VAL\_NM\_RQ={VAL\_NM\_RQ}");

// var rez = doc.Element("ValCurs").Elements("Record").Select((e, i) => new {i = i, e = e});

var rez = doc.Element("ValCurs").Elements("Record");

Console.WriteLine("Count: " + rez.Count());

// var rezFin = fakerRu.Random.ListItems(rez.ToList(), 10).OrderBy(e => e.i).Select(e => e.e);

foreach (var n in rez) {

Console.WriteLine(n);

}

// var yval = new double[] { 5, 6, 4, 6, 3 };

// var xval = new string[] { "Январь", "Февраль", "Март", "Апрель", "Май" };

var yval = rez.Elements("Value").Select(e => float.Parse(e.Value)).ToArray();

var xval = rez.Attributes("Date").Select(e => e.Value).ToArray();

// Console.WriteLine("TestX: ");

// foreach (var n in xval) {

// Console.WriteLine(n);

// }

// Console.WriteLine("TestY: ");

// foreach (var n in yval) {

// Console.WriteLine(n);

// }

mySeries1.Points.DataBindXY(xval, yval);

Console.WriteLine("Конец num1P2\n");

this.ShowDialog();

}

private void genGraphNum2() {

// Область в которой будет построен график// (Их может быть несколько)

ChartArea myChartArea1 = new(){Name = "myChartArea1"};

myChartArea1.AxisY.IsStartedFromZero = false;

myChart.ChartAreas.Add(myChartArea1);

// Область в которой будет построен график// (Их может быть несколько)

ChartArea myChartArea2 = new(){Name = "myChartArea2"};

myChartArea2.AxisY.IsStartedFromZero = false;

myChart.ChartAreas.Add(myChartArea2);

// Область в которой будет построен график// (Их может быть несколько)

ChartArea myChartArea3 = new(){Name = "myChartArea3"};

myChartArea3.AxisY.IsStartedFromZero = false;

myChart.ChartAreas.Add(myChartArea3);

myChart.Titles.Add(new Title () {

Text = "Золото",

DockedToChartArea = "myChartArea"

});

myChart.Titles.Add(new Title () {

Text = "Серебро",

DockedToChartArea = "myChartArea1"

});

myChart.Titles.Add(new Title () {

Text = "Платина",

DockedToChartArea = "myChartArea2"

});

myChart.Titles.Add(new Title () {

Text = "Палладий",

DockedToChartArea = "myChartArea3"

});

// График (Их может быть несколько)

mySeries2 = new() {

ChartType = SeriesChartType.Spline,

ChartArea = "myChartArea1"

};

myChart.Series.Add(mySeries2);

// График (Их может быть несколько)

mySeries3 = new() {

ChartType = SeriesChartType.Spline,

ChartArea = "myChartArea2"

};

myChart.Series.Add(mySeries3);

// График (Их может быть несколько)

mySeries4 = new() {

ChartType = SeriesChartType.Spline,

ChartArea = "myChartArea3"

};

myChart.Series.Add(mySeries4);

}

private void num2P2() {

genGraph("График'и");

genGraphNum2();

Console.WriteLine("num2P2:");

Console.WriteLine("Введите date\_req1(Пример | по умолчанию, '02/01/2021'):");

string date\_req1 = defConsole(Console.ReadLine(), "02/01/2021");

Console.WriteLine("Введите date\_req2(Пример | по умолчанию, '20/02/2021'):");

string date\_req2 = defConsole(Console.ReadLine(), "20/02/2021");

// XDocument doc = XDocument.Load("https://www.cbr.ru/scripts/xml\_metall.asp?date\_req1=01/01/2021&date\_req2=01/02/2021");

XDocument doc = XDocument.Load($"http://www.cbr.ru/scripts/xml\_metall.asp?date\_req1={date\_req1}&date\_req2={date\_req2}");

var rez = doc.Element("Metall").Elements("Record");

var rez1 = rez.Where(e => e.Attribute("Code").Value == "1");

var rez2 = rez.Where(e => e.Attribute("Code").Value == "2");

var rez3 = rez.Where(e => e.Attribute("Code").Value == "3");

var rez4 = rez.Where(e => e.Attribute("Code").Value == "4");

Console.WriteLine("Count: " + rez.Count());

var xval = rez1.Attributes("Date").Select(e => e.Value).ToArray();

var yval1 = rez1.Select(e => new {k = e.Attribute("Date").Value, v = float.Parse(e.Element("Sell").Value)});

var yval2 = rez2.Select(e => new {k = e.Attribute("Date").Value, v = float.Parse(e.Element("Sell").Value)});

var yval3 = rez3.Select(e => new {k = e.Attribute("Date").Value, v = float.Parse(e.Element("Sell").Value)});

var yval4 = rez4.Select(e => new {k = e.Attribute("Date").Value, v = float.Parse(e.Element("Sell").Value)});

mySeries1.Points.DataBindXY(xval, yval1.Select(e => e.v).ToArray());

mySeries2.Points.DataBindXY(xval, yval2.Select(e => e.v).ToArray());

mySeries3.Points.DataBindXY(xval, yval3.Select(e => e.v).ToArray());

mySeries4.Points.DataBindXY(xval, yval4.Select(e => e.v).ToArray());

Console.WriteLine($"Золото, цены: минимальная: {yval1.MinBy(e => e.v).k}/{yval1.MinBy(e => e.v).v}, максимальная: {yval1.MaxBy(e => e.v).k}/{yval1.MaxBy(e => e.v).v}");

Console.WriteLine($"Серебро, цены: минимальная: {yval2.MinBy(e => e.v).k}/{yval2.MinBy(e => e.v).v}, максимальная: {yval2.MaxBy(e => e.v).k}/{yval2.MaxBy(e => e.v).v}");

Console.WriteLine($"Платина, цены: минимальная: {yval3.MinBy(e => e.v).k}/{yval3.MinBy(e => e.v).v}, максимальная: {yval3.MaxBy(e => e.v).k}/{yval3.MaxBy(e => e.v).v}");

Console.WriteLine($"Палладий, цены: минимальная: {yval4.MinBy(e => e.v).k}/{yval4.MinBy(e => e.v).v}, максимальная: {yval4.MaxBy(e => e.v).k}/{yval4.MaxBy(e => e.v).v}");

Console.WriteLine("Конец num2P2\n");

this.ShowDialog();

}

private void num3P2() {

Console.WriteLine("num3P2:");

XDocument doc = XDocument.Load("data-20210603T1050-structure-20150929T0000.xml");

foreach (var n in doc.Element("data").Elements("record").Where(e => e.Element("unit").Value == "Астраханская область")) {

Console.WriteLine(n);

}

Console.WriteLine("Конец num3P2\n");

}

}

Результаты

Результаты выполнения кода первой части(1-5 задачи) и второй части(1-3 задачи) представлены на Рисунке 1.

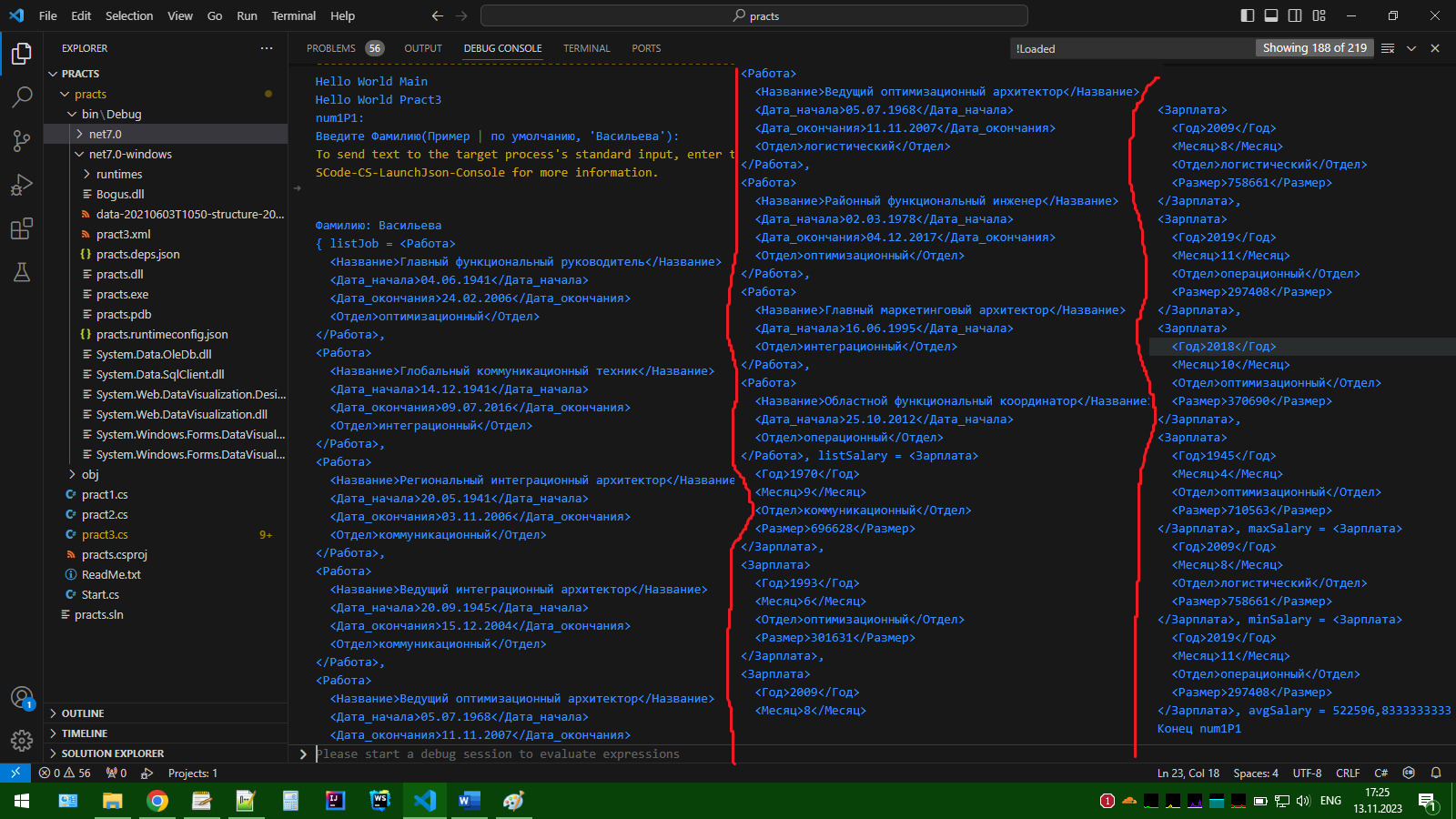


Рисунок 1 – Результаты компиляции кода 1 задачи из первой части

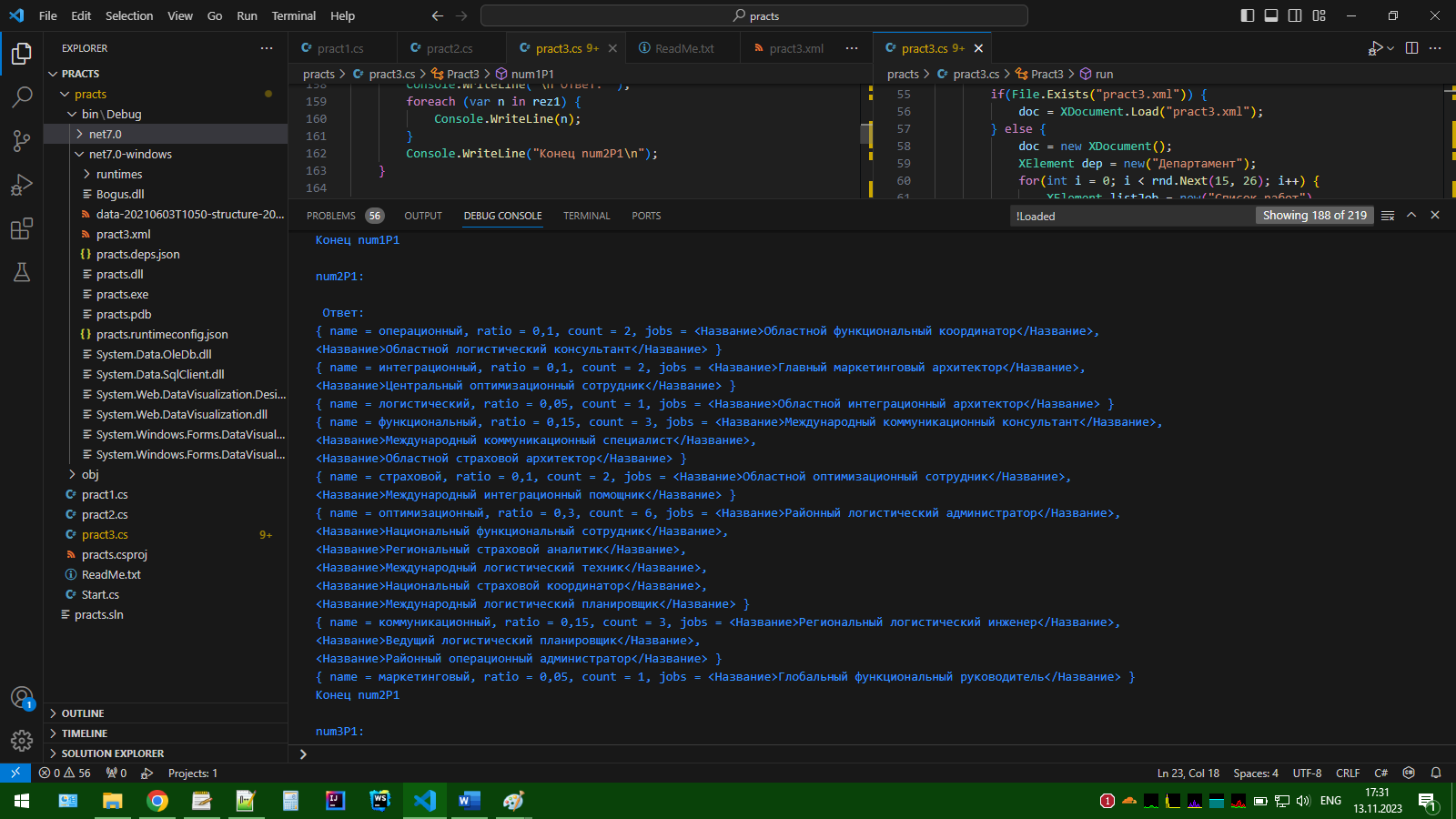


Рисунок 2 – Результаты компиляции кода 2 задачи из первой части

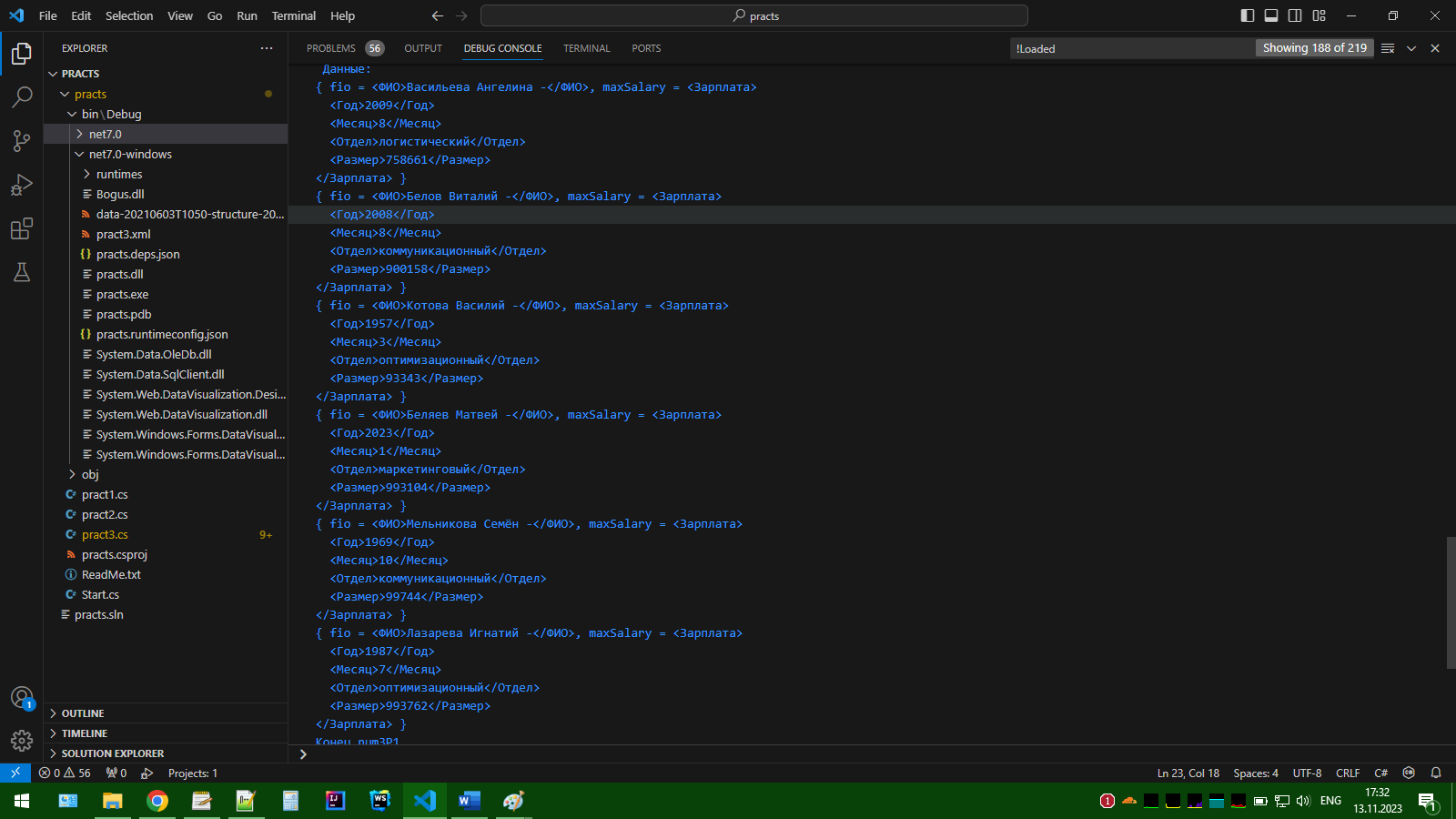


Рисунок 3 – Результаты компиляции кода 3 задачи из первой части

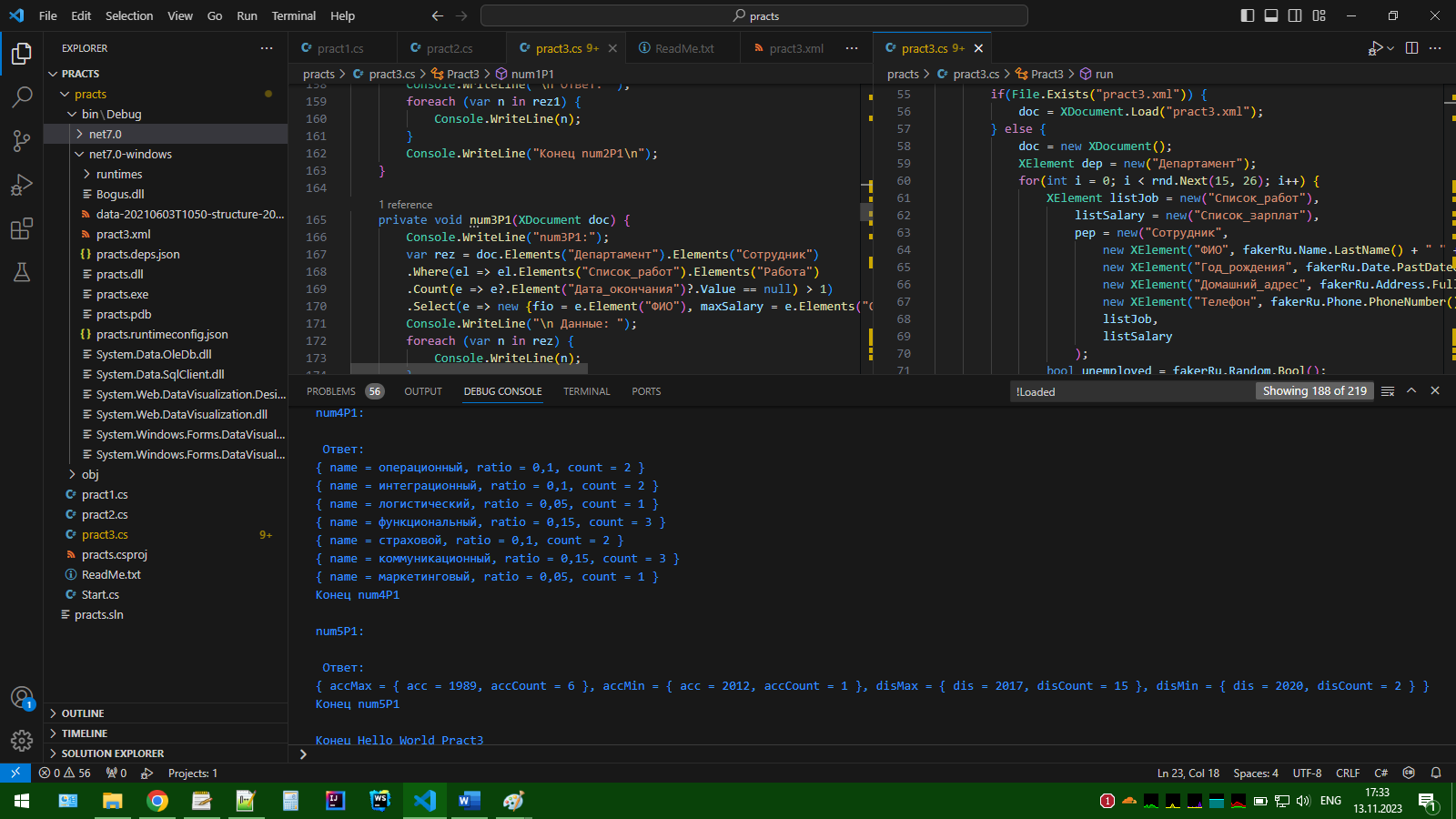


Рисунок 4 – Результаты компиляции кодов 4 и 5 задач из первой части

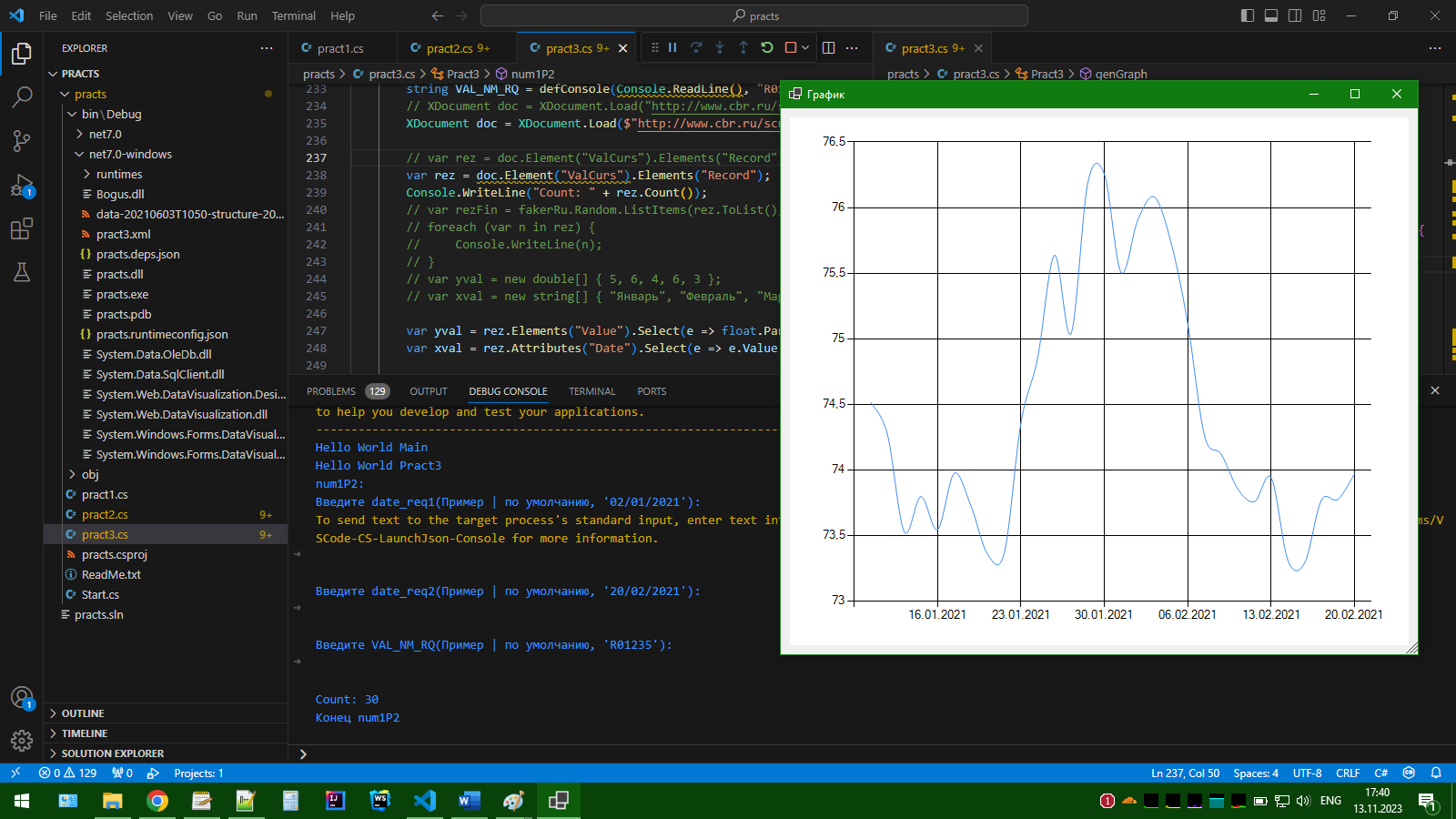


Рисунок 5 – Результаты компиляции кода 1 задачи из второй части

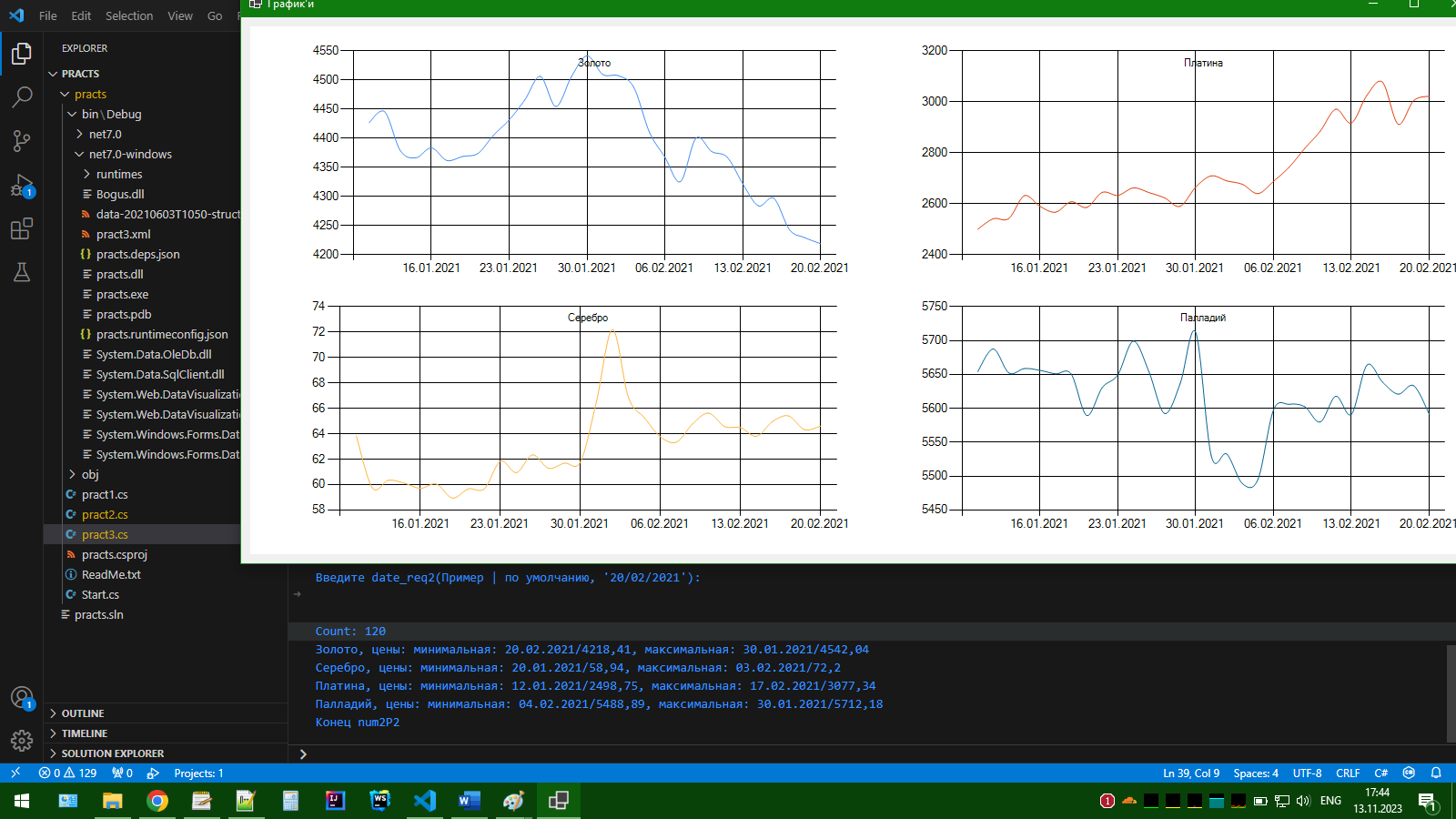


Рисунок 6 – Результаты компиляции кода 2 задачи из второй части

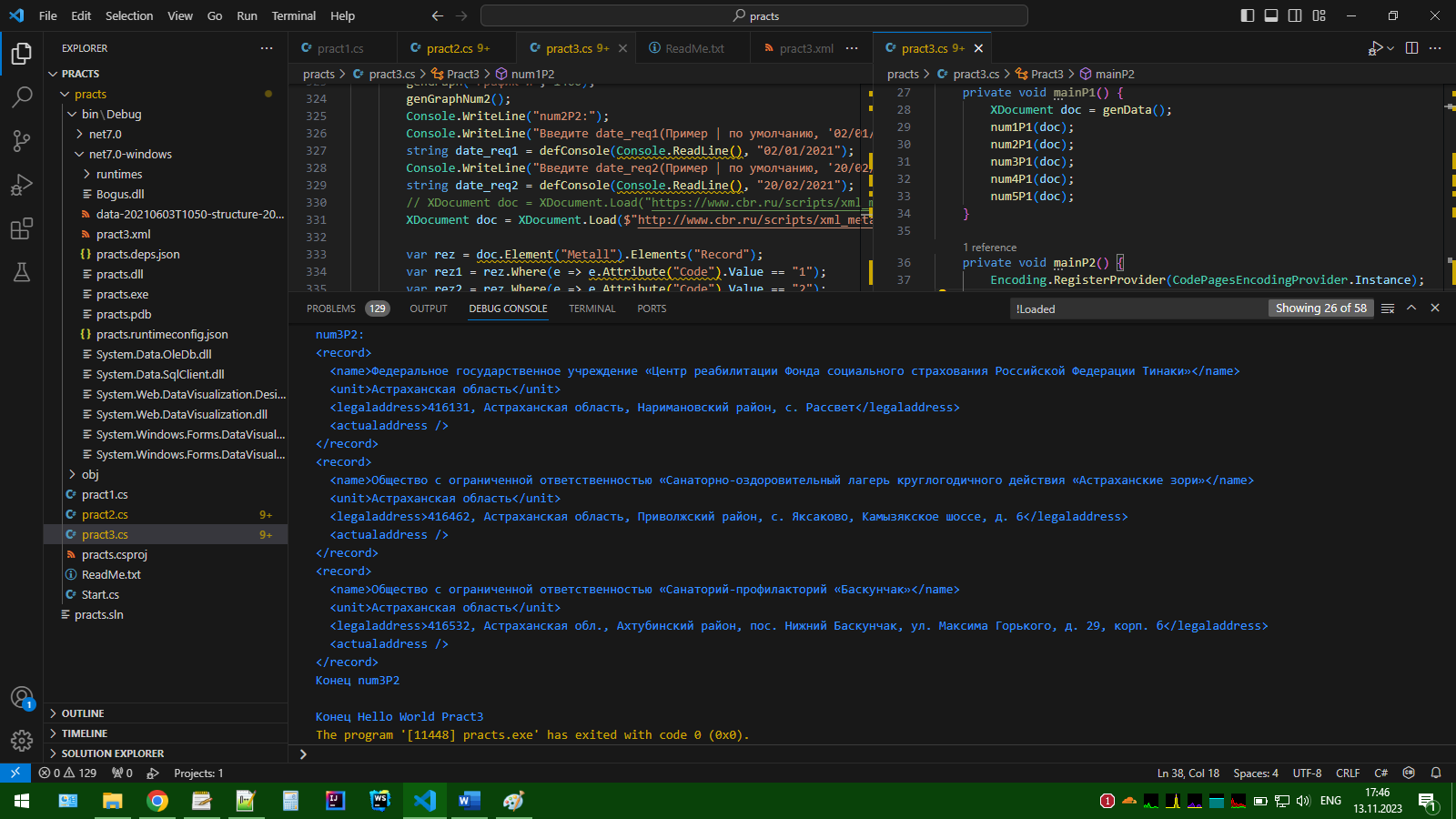


Рисунок 7 – Результаты компиляции кода 3 задачи из второй части