Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №9

по дисциплине «Технологии программирования»

Тема: «Разработка многопоточных приложений»

Выполнил:

студент гр. ПРИ-123

А.Ц. Нямаа

Приняла:

Ст. пр. кафедры ИСПИ

В.В. Данилов

Владимир, 2024 г.

ЦЕЛЬ РАБОТЫ:

Познакомиться с возможностями языка программирования C# при разработке многопоточных приложений.

ЗАДАНИЕ 1(вариант 13).

Выполнить демонстрационные примеры, показывающий работу с лямбда потоками.

ВЫПОЛНЕНИЕ РАБОТЫ:

Листинг примера 1. Демонстрация потоковой небезопасности:

using System;

using System.Threading;

class ThreadSafe

{

static int maxi = 100;

static int i;

static void Main()

{

new Thread(GoA).Start();

GoB();

Console.ReadLine();

}

static void GoA()

{

for (; i < maxi; i++)

Console.Write("A" + i + " ");

}

static void GoB()

{

for (; i < maxi; i++)

Console.Write("B" + i + " ");

}

}

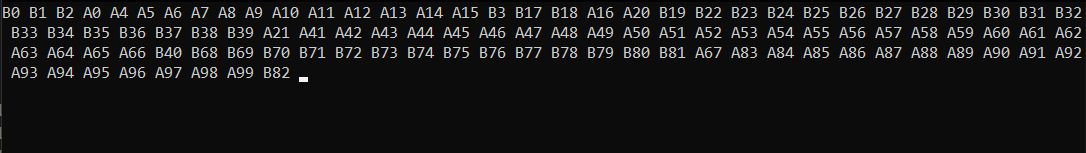


Рисунок 1. Выполнение программы примера 1.

Листинг примера 2. Демонстрация метода блокировок:

using System;

using System.Threading;

class ThreadSafe

{

static int maxi = 100;

static int i;

static object locker = new object();

static void Main()

{

new Thread(GoA).Start();

GoB();

Console.ReadLine();

}

static void GoA()

{

for (; i < maxi;)

{

lock (locker)

{

Console.Write("A" + i + " ");

i++;

}

}

}

static void GoB()

{

for (; i < maxi;)

{

lock (locker)

{

Console.Write("B" + i + " ");

i++;

}

}

}

}

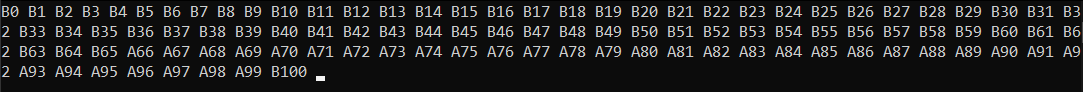


Рисунок 2. Выполнение программы примера 2.

ЗАДАНИЕ 2(вариант 13).

Реализовать программу на C# в соответствии с вариантом задания из предыдущих работ с использованием многопоточности.

ВЫПОЛНЕНИЕ РАБОТЫ:

Листинг Form1.cs:

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Windows.Forms;

namespace lab\_9

{

public partial class Form1 : Form

{

Random random = new Random();

List<Person> people = new List<Person>();

List<IDoc> documents = new List<IDoc>();

List<subject> subjects = new List<subject>();

Stopwatch time = new Stopwatch();

DateTime start\_time = DateTime.Now;

int[] quantity\_methods = new int[3];

public Form1()

{

InitializeComponent();

time.Start();

}

private void button1\_Click(object sender, EventArgs e)

{

string input = textBox1.Text;

if (int.TryParse(input, out int count\_person))

{

for (int i = 0; i < count\_person; i++)

{

Persons newPerson = new Persons();

newPerson.ShowDialog();

if (newPerson.radioValue == "Профессор")

{

people.Add(new Proffesor {

Id = people.Count + 1,

Name = newPerson.name,

Surname = newPerson.second\_name,

BirthDay = newPerson.date,

workBook = random.Next(100000, 999999)

});

}

else if (newPerson.radioValue == "Студент")

{

people.Add(new Student {

Id = people.Count + 1,

Name = newPerson.name,

Surname = newPerson.second\_name,

BirthDay = newPerson.date,

recordBook = random.Next(1000000, 9999999)

});

}

quantity\_methods[0] += 1;

}

}

else

{

MessageBox.Show("Ошибка\nВведите целочисленное число");

}

}

private void button2\_Click(object sender, EventArgs e)

{

string input = textBox2.Text;

if (int.TryParse(input, out int count\_docs))

{

for (int i = 0; i < count\_docs; i++)

{

Docs newDoc = new Docs();

newDoc.ShowDialog();

if (newDoc.radioValue == "Результаты экзамена")

{

documents.Add(new Exam {

Id = documents.Count + 1,

Document = "экзамена",

StudentsQuant = newDoc.students\_quant,

Group = newDoc.group,

Date = newDoc.date

});

}

else if (newDoc.radioValue == "Результаты зачета")

{

documents.Add(new Test {

Id = documents.Count + 1,

Document = "зачета",

StudentsQuant = newDoc.students\_quant,

Group = newDoc.group,

Date = newDoc.date

});

}

else if (newDoc.radioValue == "Информация о ведомостях")

{

documents.Add(new State {

Id = documents.Count + 1,

Document = "ведомости",

StudentsQuant = newDoc.students\_quant,

Group = newDoc.group,

Date = newDoc.date

});

}

quantity\_methods[1] += 1;

}

}

else

{

MessageBox.Show("Ошибка\nВведите целочисленное число");

}

}

private void button3\_Click(object sender, EventArgs e)

{

string input = textBox3.Text;

if (int.TryParse(input, out int count\_subjects))

{

for (int i = 0; i < count\_subjects; i++)

{

Subjects newSubjects = new Subjects();

newSubjects.ShowDialog();

subjects.Add(new subject {

Id = subjects.Count + 1,

Subject = newSubjects.subject

});

quantity\_methods[2] += 1;

}

}

else

{

MessageBox.Show("Ошибка\nВведите целочисленное число");

}

}

private void button4\_Click(object sender, EventArgs e)

{

Result result = new Result(people.ToArray(), documents.ToArray(), subjects.ToArray(), time, start\_time, quantity\_methods);

result.Show();

}

}

}

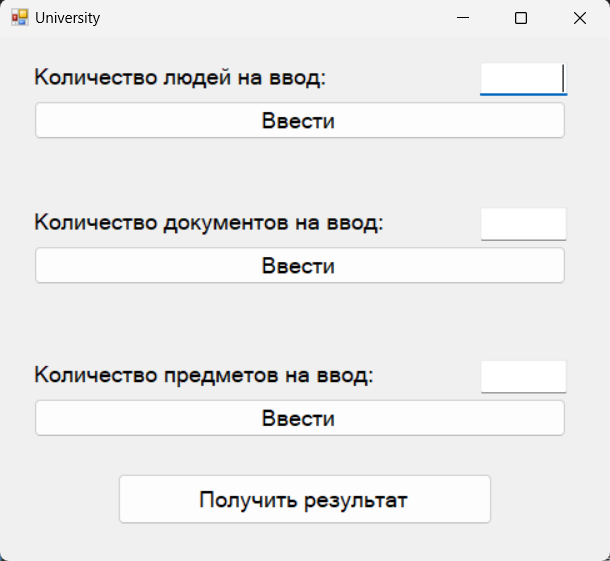


Рисунок 3 – Скриншот главной формы.

Листинг Docs.cs:

using System;

using System.Linq;

using System.Windows.Forms;

using System.Threading;

namespace lab\_9

{

delegate bool isChecked(int toCheckValue);

public partial class Docs : Form

{

public string radioValue;

public string group;

public DateTime date;

public int students\_quant;

public Docs()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

int[] errors = new int[2];

string radioValueTemp = checker();

string groupTemp = textBox1.Text;

int studentsQuantTemp = 0;

var thread = new Thread(() =>

{

(int studentsQuantThread, int error0) = errorCheckGroup(textBox2.Text);

studentsQuantTemp = studentsQuantThread;

errors[0] = error0;

});

thread.Start();

(DateTime dateTemp, int error1) = errorCheckDate(maskedTextBox1.Text);

errors[1] = error1;

thread.Join();

if (errors.Sum() == 0)

{

radioValue = radioValueTemp;

group = groupTemp;

students\_quant = studentsQuantTemp;

date = dateTemp;

this.Close();

}

}

private string checker()

{

string selected = "";

if (radioButton1.Checked)

{

selected = radioButton1.Text;

}

else if (radioButton2.Checked)

{

selected = radioButton2.Text;

}

else if (radioButton3.Checked)

{

selected = radioButton3.Text;

}

return selected;

}

private static (int, int) errorCheckGroup(string textBox2)

{

isChecked Checker = countToCheck => countToCheck >= 0 && countToCheck <= 30;

int error0;

int studentsQuant = 0;

if (Int32.TryParse(textBox2, out int count) && Checker(count))

{

studentsQuant = count;

error0 = 0;

}

else

{

MessageBox.Show("Ошибка\nВ группе от 1 до 30 человек");

error0 = 1;

}

return (studentsQuant, error0);

}

private static (DateTime, int) errorCheckDate(string input)

{

DateTime dateThread = DateTime.Now;

int error1;

if (DateTime.TryParse(input, out DateTime dateTemp))

{

if (dateTemp.Year <= DateTime.Now.Year && dateTemp.Year >= 1924)

{

dateThread = dateTemp;

error1 = 0;

}

else

{

MessageBox.Show("Ошибка\nДата не реальная!");

error1 = 1;

}

}

else

{

MessageBox.Show("Ошибка\nВведена неправильная дата");

error1 = 1;

}

return (dateThread, error1);

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

label4.Text = ("Сколько человек сдало?");

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

label4.Text = ("Сколько человек получило зачет?");

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

label4.Text = ("Сколько человек с идеальной посещаемостью?");

}

}

}

Листинг Docs\_class.cs

using System;

namespace lab\_9

{

interface IDoc

{

int Id { get; set; }

string Document { get; set; }

string Group { get; set; }

DateTime Date { get; set; }

int StudentsQuant { get; set; }

string WhatADoc();

string results();

}

class Exam : IDoc

{

public int Id { get; set; }

public string Document { get; set; }

public string Group { get; set; }

public DateTime Date { get; set; }

public int StudentsQuant { get; set; }

public string WhatADoc()

{

return "экзаменационный документ";

}

public string results()

{

return $"количество сдавших экзамен - {StudentsQuant}";

}

}

class Test : IDoc

{

public int Id { get; set; }

public string Document { get; set; }

public string Group { get; set; }

public DateTime Date { get; set; }

public int StudentsQuant { get; set; }

public string WhatADoc()

{

return "документ о зачетах";

}

public string results()

{

return $"количество сдавших зачет - {StudentsQuant}";

}

}

class State : IDoc

{

public int Id { get; set; }

public string Document { get; set; }

public string Group { get; set; }

public DateTime Date { get; set; }

public int StudentsQuant { get; set; }

public string WhatADoc()

{

return "ведомость";

}

public string results()

{

return $"количество идеальных посещений - {StudentsQuant}";

}

}

}

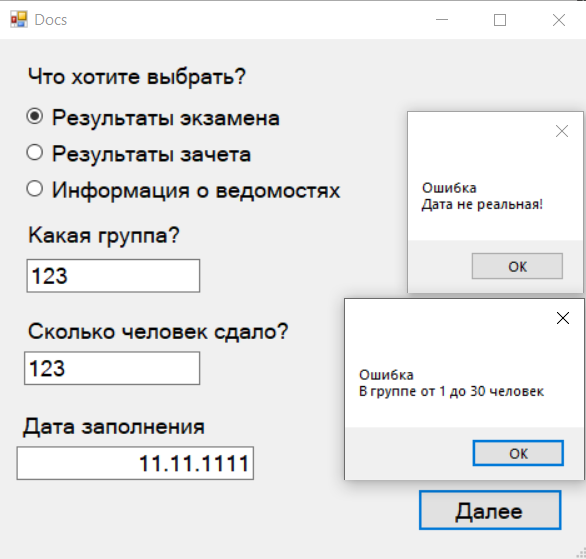


Рисунок 4 – Скриншот формы Docs.

Листинг Persons.cs

using System;

using System.Windows.Forms;

using System.Threading;

namespace lab\_9

{

public partial class Persons : Form

{

public string name;

public string second\_name;

public DateTime date;

public string radioValue;

public Persons()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

int error = 1;

var thread = new Thread(() =>

{

(DateTime dateTemp, int errorTemp) = errorCheckDate(maskedTextBox1.Text);

date = dateTemp;

error = errorTemp;

});

thread.Start();

string radioValueTemp = checker();

string nameTemp = textBox1.Text;

string second\_nameTemp = textBox2.Text;

DateTime dateThread = DateTime.Now;

thread.Join();

if (error == 0)

{

radioValue = radioValueTemp;

name = nameTemp;

second\_name = second\_nameTemp;

date = dateThread;

this.Close();

}

}

private static (DateTime, int) errorCheckDate(string input)

{

DateTime dateThread = DateTime.Now;

int error;

if (DateTime.TryParse(input, out DateTime dateTemp))

{

if (dateTemp.Year <= DateTime.Now.Year && dateTemp.Year >= 1924)

{

dateThread = dateTemp;

error = 0;

}

else

{

MessageBox.Show("Ошибка\nДата не реальная!");

error = 1;

}

}

else

{

MessageBox.Show("Ошибка\nВведена неправильная дата");

error = 1;

}

return (dateThread, error);

}

private string checker()

{

string selected = "";

if (radioButton1.Checked)

{

selected = radioButton1.Text;

}

else if (radioButton2.Checked)

{

selected = radioButton2.Text;

}

return selected;

}

}

}

Листинг Persons\_class.cs

using System;

namespace lab\_9

{

class Person

{

public int Id;

public string Name;

public string Surname;

public DateTime BirthDay;

}

class Student : Person

{

public int recordBook;

}

class Proffesor : Person

{

public int workBook;

}

}

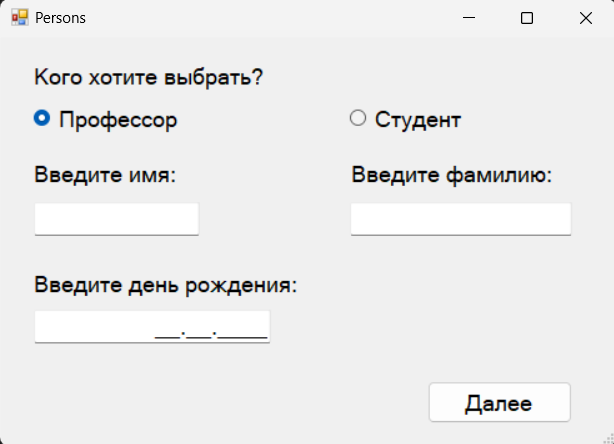


Рисунок 5 – Скриншот формы Person.

Листинг Subjects.cs

using System;

using System.Windows.Forms;

namespace lab\_9

{

public partial class Subjects : Form

{

public string subject;

public Subjects()

{

InitializeComponent();

}

private void button4\_Click(object sender, EventArgs e)

{

subject = textBox1.Text;

this.Close();

}

}

}

Листинг Subjects\_class.cs

using System.Linq;

using System;

namespace lab\_9

{

delegate string SubCheck(string subject);

class subject

{

private string[] checker = { "Математика", "Иностранный", "Программирование", "Философия" };

private string[] teachers = { "Иванов", "Петров", "Сидоров", "Федоров" };

Random rand = new Random();

public int Id;

public string Subject;

public string teach(string Subject)

{

SubCheck checkSub = subject => checker.Contains(subject) ? $"{teachers[rand.Next(0, 3)]}" : "раз такого предмета нет, то и преподавателя тоже нет";

return checkSub(Subject);

}

public string Check(string Subject)

{

SubCheck checkSub = subject => checker.Contains(subject) ? "такой предмет существует" : "такого предмета не существует";

return checkSub(Subject);

}

}

}

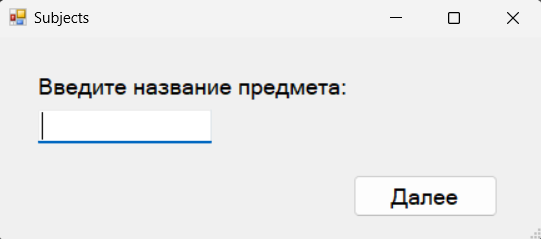


Рисунок 6 – Скриншот формы Subjects.

Листинг Result.cs

using System;

using System.Diagnostics;

using System.Windows.Forms;

using lab\_9;

namespace lab\_9

{

public partial class Result : Form

{

Stopwatch time;

DateTime start\_time;

int[] quantity\_methods;

internal Result(Person[] people, IDoc[] documents, subject[] subjects, Stopwatch time, DateTime start\_time, int[] quantity\_methods)

{

InitializeComponent();

this.time = time;

this.start\_time = start\_time;

this.quantity\_methods = quantity\_methods;

if (people != null)

{

for (int i = 0; i < people.Length; i++)

{

if (people[i] != null && people[i] is Proffesor prof)

{

label1.Text += ($"id преподавателя - {prof.Id}, его имя - {prof.Name}, фамилия - {prof.Surname}, родился {prof.BirthDay.ToShortDateString()}, номер трудовой книжки - {prof.workBook}\n\n");

}

else if (people[i] != null && people[i] is Student student)

{

label1.Text += ($"id студента - {student.Id}, его имя - {student.Name}, фамилия - {student.Surname}, родился {student.BirthDay.ToShortDateString()}, номер зачетки - {student.recordBook}\n\n");

}

}

}

if (documents != null)

{

for (int i = 0; i < documents.Length; i++)

{

if (documents[i] != null)

{

label2.Text += ($"id документа - {documents[i].Id}, название документа - {documents[i].WhatADoc()}, необходим для {documents[i].Document}, группа - {documents[i].Group}, {documents[i].results()}, дата подписания документа {documents[i].Date.ToShortDateString()}\n\n");

}

}

}

if (subjects != null)

{

for (int i = 0; i < subjects.Length; i++)

{

label3.Text += ($"id предмета - {subjects[i].Id}, предмет - {subjects[i].Subject}, {subjects[i].Check(subjects[i].Subject)}, преподаватель - {subjects[i].teach(subjects[i].Subject)}\n\n");

}

}

}

private void button1\_Click(object sender, System.EventArgs e)

{

add\_info add\_Info = new add\_info(time, start\_time, quantity\_methods);

add\_Info.Show();

}

}

}

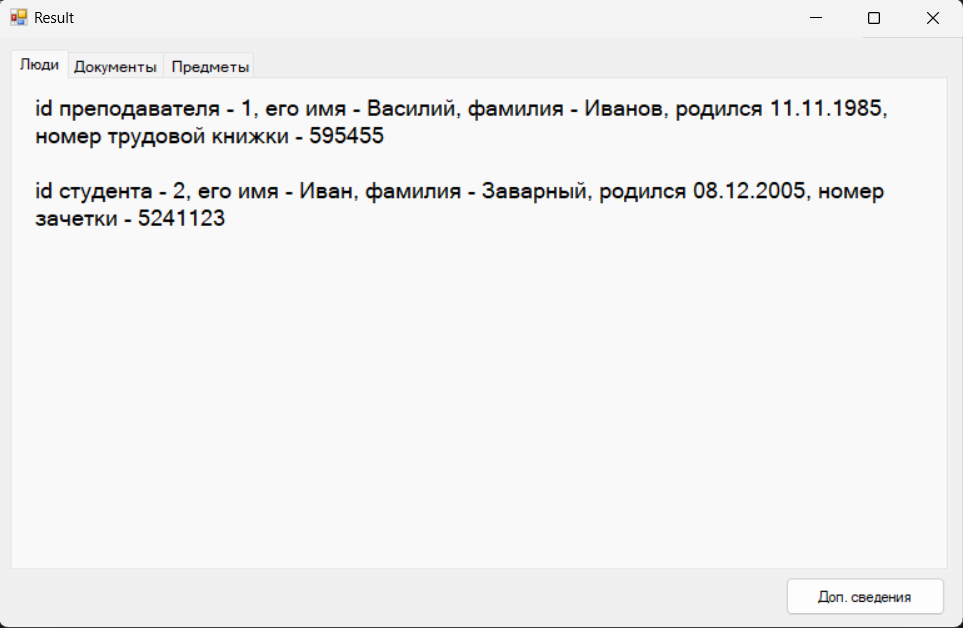


Рисунок 7 – Скриншот формы Result. Вкладка 1.

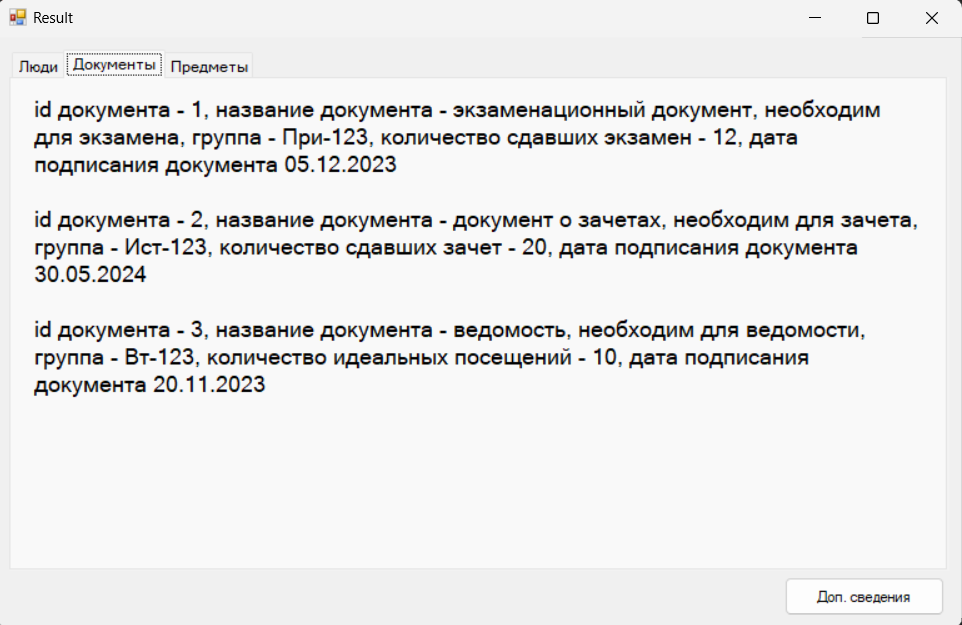


Рисунок 8 – Скриншот формы Result. Вкладка 2.

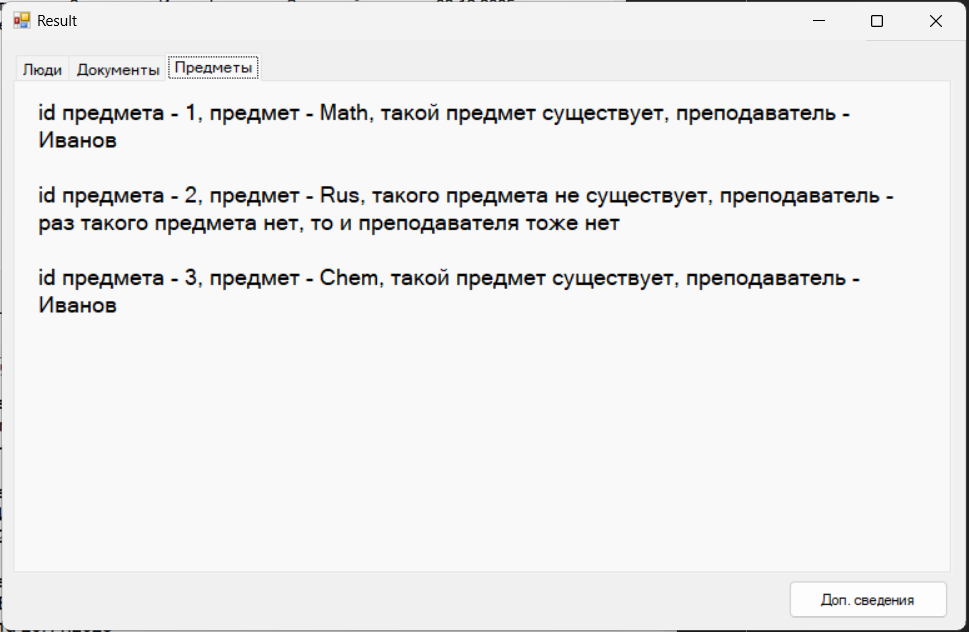


Рисунок 9 – Скриншот формы Result. Вкладка 3.

Листинг add\_info.cs

using System;

using System.Diagnostics;

using System.Windows.Forms;

namespace lab\_9

{

public partial class add\_info : Form

{

public add\_info(Stopwatch time, DateTime start\_time, int[] quantity\_methods)

{

InitializeComponent();

label1.Text = ($"Время выполнения программы - {time.ElapsedMilliseconds}ms");

label2.Text = ($"Время запуска программы - {start\_time.ToLongTimeString()}");

label5.Text = ($"Количество выполнения метода Persons - {quantity\_methods[0]} раз");

label4.Text = ($"Количество выполнения метода IDoc - {quantity\_methods[1]} раз");

label3.Text = ($"Количество выполнения метода Subjects - {quantity\_methods[2]} раз");

label6.Text = ($"Операционная система - {Environment.OSVersion}");

}

}

}

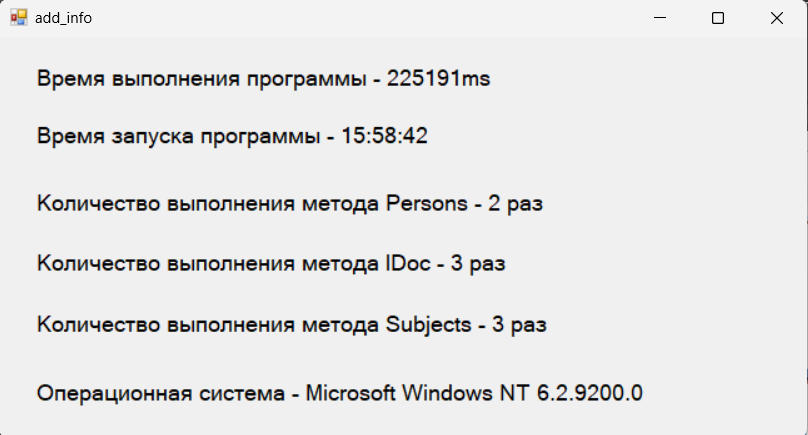


Рисунок 10 – Скриншот формы add\_info.

ВЫВОД К РАБОТЕ:

Я познакомился с возможностями языка программирования C# при разработке многопоточных приложений.