Отчет по практической работе №13

Наименование работы: делегаты.

Индивидуальные задания (вариант №9):

1. Создайте приложение C# в среде MS Visual Studio.

В соответствии с вариантом индивидуального задания реализуйте пользовательский тип делегата требуемой сигнатуры и выполните с его использованием вызов нескольких методов (с корректной сигнатурой).

|  |  |
| --- | --- |
| 9 | Action<Func<int>, char, char> |

1. Создайте приложение на языке C# в MS Visual Studio.

В соответствии с индивидуальным вариантом разработайте требуемый тип делегата (пользовательский, библиотечный или лямбдавыражение).

Реализуйте асинхронное выполнение метода на основе разработанного делегата с возможностью мониторинга процесса выполнения, передачи параметров в метод и получения результата работы метода.

Библиотечные делегаты это Action, Predicate, Func.

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | лямбда-выражение | Метод возвращает результат шифрования строки: каждый исходный символ строки заменяется шифрованным символом, код которого на n больше кода исходного символа. | Два параметра: исходная  строка, число  сдвига n |

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы методы, делегаты.

Листинг программы:

using System;

namespace Lab13.\_1

{

class Program

{

static void Main(string[] args)

{

char x1 = Convert.ToChar("Д");

char x2 = Convert.ToChar("р");

Console.WriteLine("Hello World!");

Action < Func<int>, char, char> op = Test2;

Func<int> fc = Test1;

fc.Invoke();

op.Invoke(fc, x1, x2);

}

static public void Test2(Func<int> t, char ch, char ch2)

{

Console.WriteLine($"{ch}елегат - {ch2}еализован!");

}

static public int Test1()

{

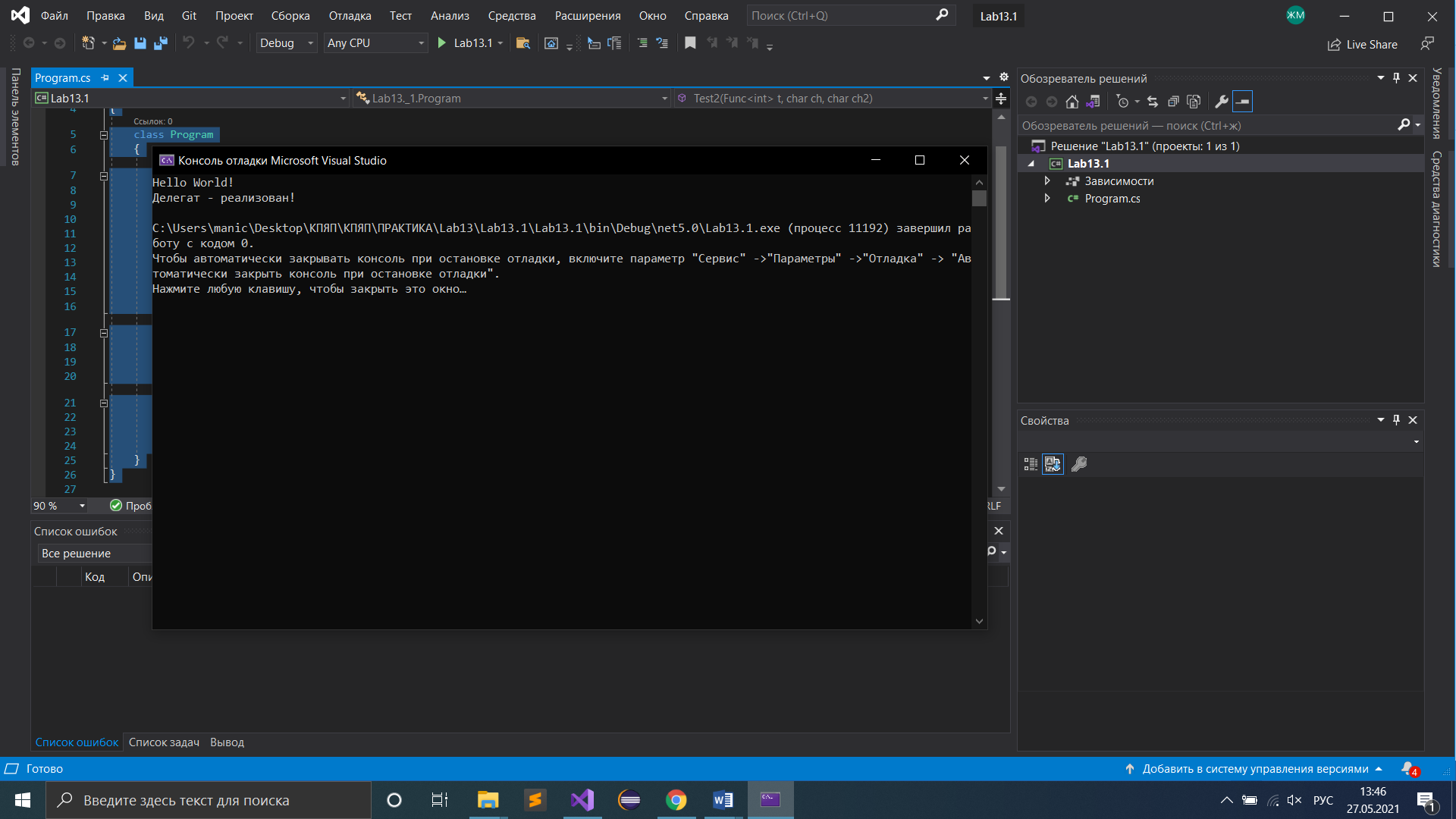
return 2;

}

}

}

Результат выполнения программы Рис 1.1.



**Рисунок 1.1 – Пример работы программы c делегатами**

1. Разработал программу в соответствии с заданием №2. Были реализованы методы, делегаты.

Листинг программы:

using System;

using System.Text;

namespace Lab13.\_2

{

delegate void Meth();

class Program

{

const string alfabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

static void Main(string[] args)

{

Console.Write("Введите строку для шифровки: ");

string stroka = Console.ReadLine();

Console.Write("Введите ключ для сдвига: ");

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

Meth mt = () => Method(stroka, n);

mt();

}

public static void Method(string str, int n)

{

var rez = new StringBuilder();

string alf = alfabet + alfabet.ToLower();

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

{

for(int j = 0; j < alf.Length; j++)

{

if(str[i] == alf[j])

{

rez.Append(alf[j + n]);

}

}

}

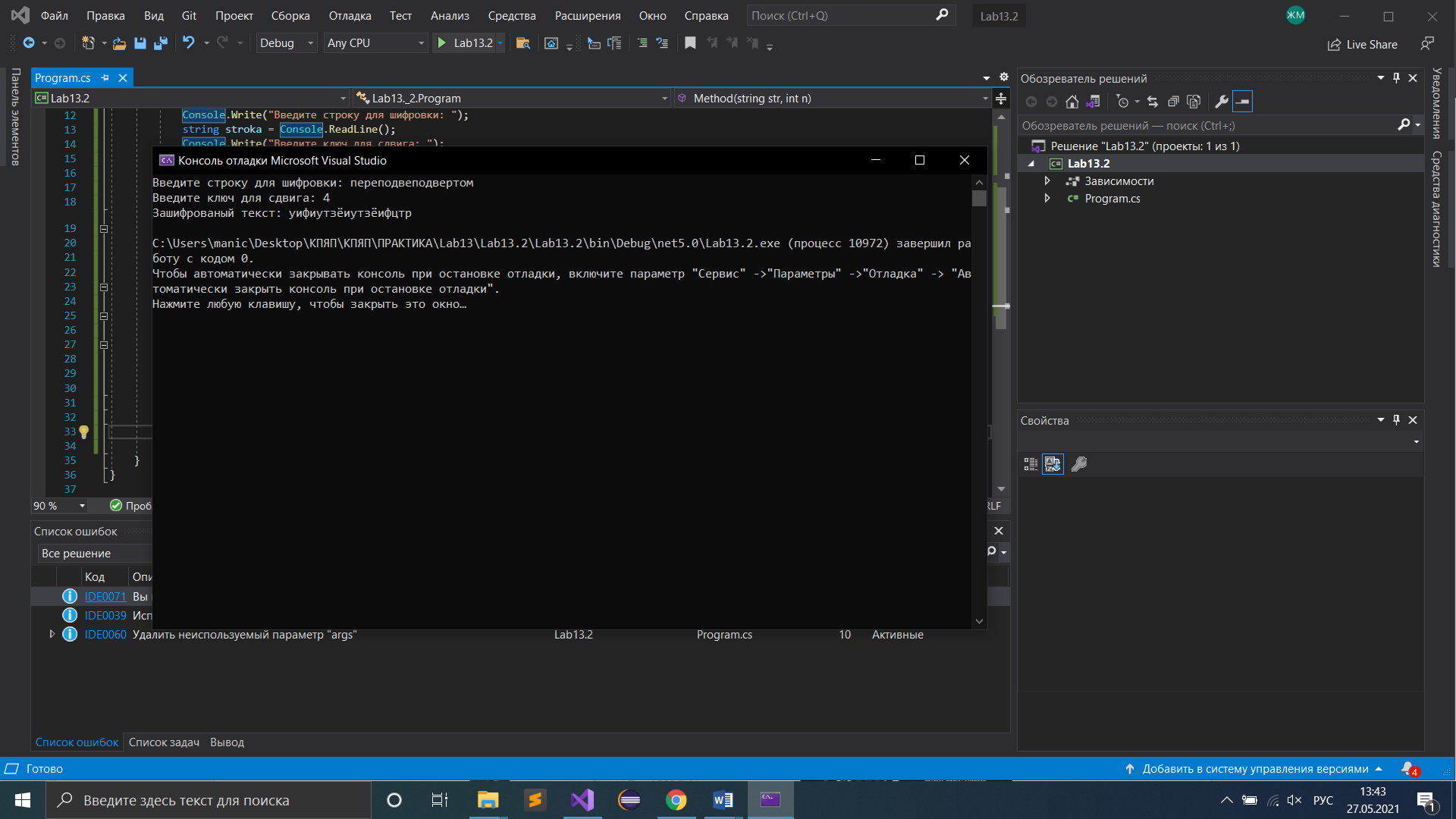
Console.WriteLine($"Зашифрованый текст: {rez.ToString()}");

}

}

}

Результат выполнения программы Рис 2.1.



**Рисунок 2.1 – Пример работы программы c делегатами**

Отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_