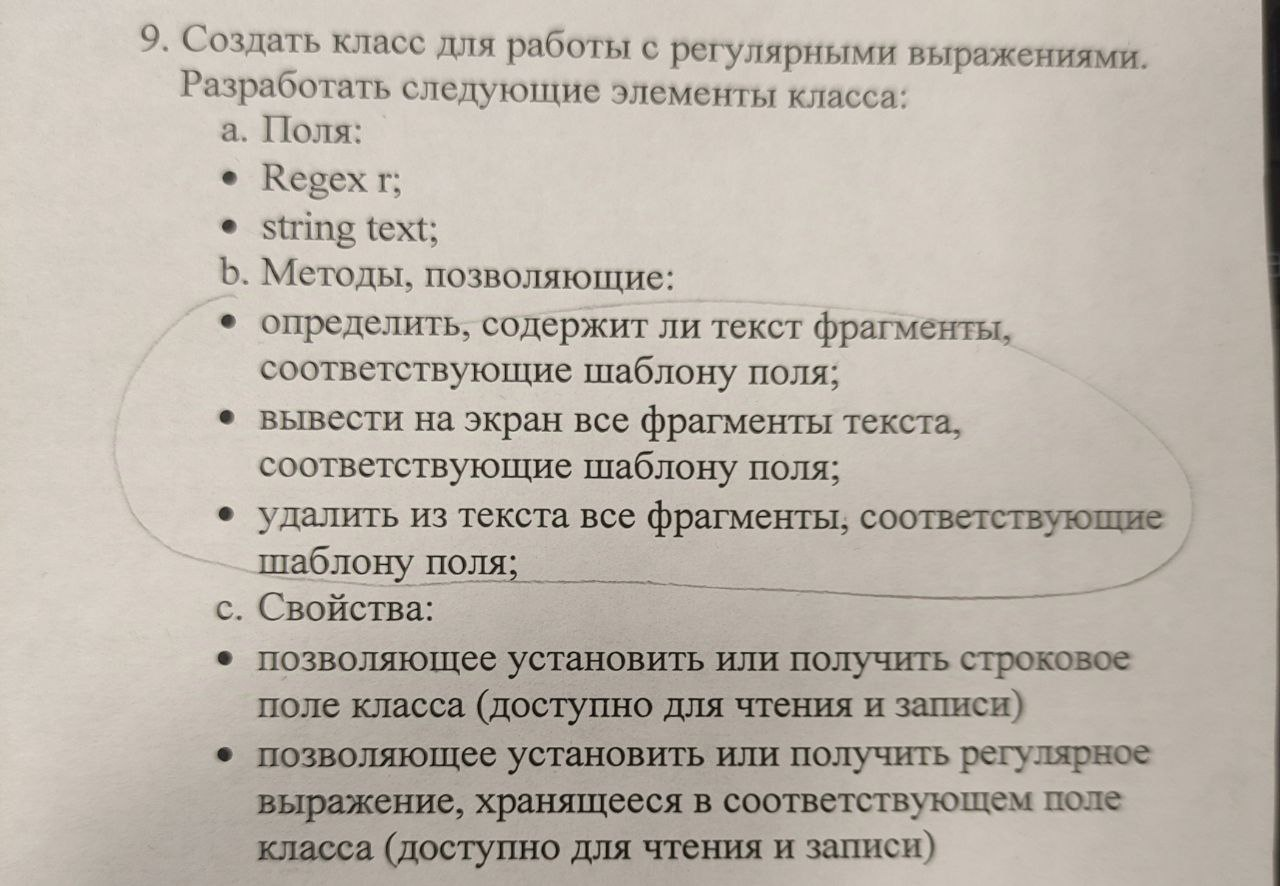
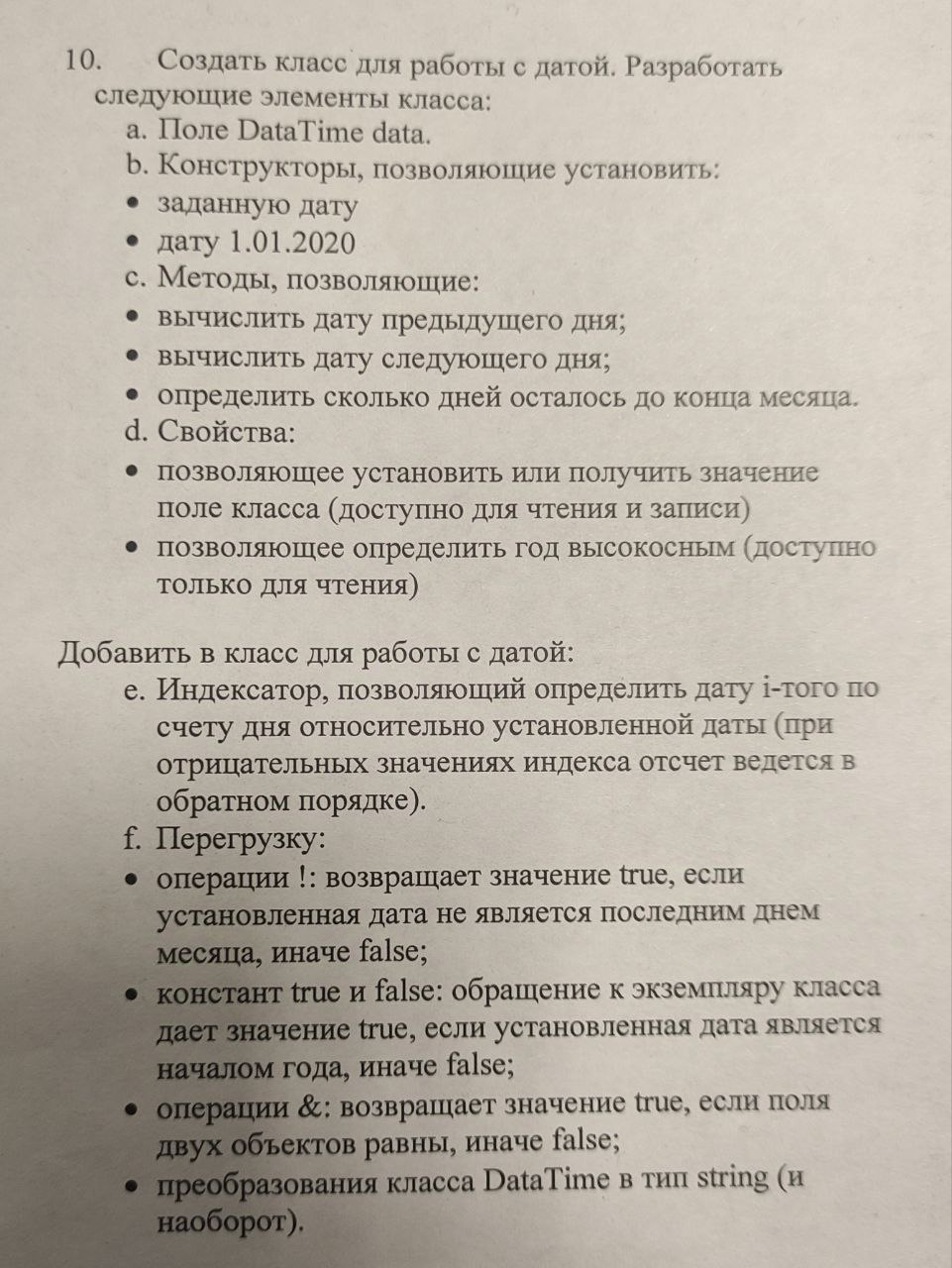
# Задание 1





Код класса Reg представлен ниже.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.Json.Serialization;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Day10

{

internal class Reg

{

private Regex r;

private string text;

public Regex R

{

set

{

if(value.ToString() == "")

{

throw new InvalidOperationException();

}

else r = value;

}

get { return r; }

}

public string T

{

set

{

if(value == "")

{

throw new InvalidOperationException();

}

else text = value;

}

get { return text; }

}

public void match()

{

MatchCollection matches = this.r.Matches(text);

Console.WriteLine("Фрагменты текста, соответствующие шаблону поля: ");

foreach(Match t in matches)

{

Console.WriteLine(t.ToString());

}

}

public void delete()

{

this.T = Regex.Replace(this.T, this.R.ToString(), "");

Console.WriteLine("Изменённая строка: ");

Console.WriteLine(this.T);

}

public void is\_match()

{

Console.WriteLine($"Соостветствие строки шаблону: {this.R.IsMatch(this.T)}");

}

public Reg(){}

public Reg(string text, string reg)

{

this.T = text;

this.R = new Regex(reg);

}

public override string ToString() => $"Регулярное выражение: {this.R}\nСтрока: {this.T}";

}

}

Код класса Data представлен ниже.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection.Metadata.Ecma335;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

namespace Day10

{

internal class Data

{

private DateTime date;

public DateTime Date

{

set

{

if (value.ToString() == "") throw new ArgumentException();

else date = value;

}

get { return date; }

}

public DateTime this[int i]

{

get

{

if (i == 0) return this.Date.AddDays(i);

else return this.Date.AddDays(i);

}

}

public override string ToString() => $"Заданная дата: {this.Date.ToShortDateString()}";

public Data()

{

this.Date = new DateTime(2020,1,1);

}

public Data(DateTime date)

{

this.Date = date;

}

public void prev()

{

Console.WriteLine($"Предыдущий день: {this.Date.AddDays(-1).ToShortDateString()}");

}

public void next()

{

Console.WriteLine($"Следующий день: {this.Date.AddDays(1).ToShortDateString()}");

}

public void days()

{

Console.WriteLine($"До конца месяца осталось {DateTime.DaysInMonth(Date.Year, Date.Month) - this.Date.Day} дней");

}

public static bool operator !(Data data)

{

if (data.Date.Day == DateTime.DaysInMonth(data.Date.Year, data.Date.Month))

{

return false;

}

else return true;

}

public static bool operator true(Data data)

{

if(data.Date.Day == 1 && data.Date.Month == 1) return true;

else return false;

}

public static bool operator false(Data data)

{

if (data.Date.Day == 1 && data.Date.Month == 1) return true;

else return false;

}

public static bool operator &(Data data1, Data data2)

{

if(data1.Date.ToShortDateString() == data2.Date.ToShortDateString()) return true;

else return false;

}

}

}

Код задание представлен ниже.

using Day10;

using System.Reflection.Metadata;

using System.Runtime.CompilerServices;

using System.Text.RegularExpressions;

while (true)

{

try

{

Console.Write("0 - Выход\n1 - Задание 1\n2 - Задание 2\n3 - Задание 3\n4 - Задание 4\nПункт: ");

int punct = Convert.ToInt32(Console.ReadLine());

switch (punct)

{

case 0:

Environment.Exit(0);

break;

case 1:

string text, reg;

Console.Write("Строка: ");

text = Console.ReadLine();

Console.Write("Регулярное выражение: ");

reg = Console.ReadLine();

try

{

Reg reg1 = new Reg(text, reg);

Console.WriteLine(reg1);

Console.Write("Новая строка: ");

text = Console.ReadLine();

Console.Write("Новоее регулярное выражение: ");

reg = Console.ReadLine();

reg1.T = text;

reg1.R = new Regex(reg);

Console.WriteLine(reg1);

reg1.is\_match();

reg1.match();

reg1.delete();

}

catch (InvalidOperationException)

{

Console.WriteLine("Путсая строка.");

}

break;

case 2:

try

{

int year, month, day;

Console.Write("Год: ");

year = Convert.ToInt32(Console.ReadLine());

Console.Write("Месяц: ");

month = Convert.ToInt32(Console.ReadLine());

Console.Write("День: ");

day = Convert.ToInt32(Console.ReadLine());

Data temp = new Data();

Data date = new Data(new DateTime(year, month, day));

Console.WriteLine(date);

date.Date = DateTime.Now;

Console.WriteLine($"Сейчас {date.Date.ToShortDateString()}");

date.next();

date.prev();

date.days();

Console.Write("Введите i-й день: ");

int i = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"I-я дата: {date[i].ToShortDateString()}");

Console.WriteLine($"Дата не является концом месяца: {!date}");

if(date)

{

Console.WriteLine("Начало года.");

}

else

{

Console.WriteLine("Не начало года.");

}

if(date & temp)

{

Console.WriteLine("Даты равны");

}

else

{

Console.WriteLine("Даты не равны");

}

}

catch (ArgumentException)

{

Console.WriteLine("Введите корректные данные.");

}

break;

case 3:

break;

case 4:

break;

default:

break;

}

}

catch (FormatException)

{

Console.WriteLine("Введите корректные данные.");

}

catch (OverflowException)

{

Console.WriteLine("Вы вышли за размерность.");

}

catch (Exception)

{

Console.WriteLine("Ошибка.");

}

}

Результаты работы программы представлен на рисунке 1.1 – 1.2.

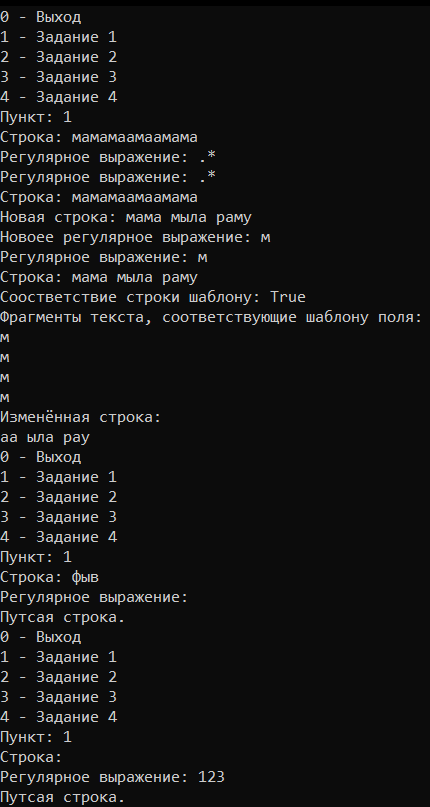


Рисунок 1.1 – Результат работы программы

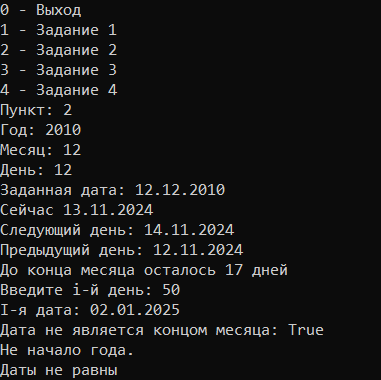


Рисунок 1.2 – Результат работы программы