Федеральное Агентство по образованию

Тольяттинский государственный университет

**Лабораторные работы**

По предмету «**Языки и методы программирования**»

Вариант заданий №2

Студент: Давыдов Антон

Группа: ФМОб-1002

Преподаватель: Зоркин В.А.

г. Тольятти

2013

1. Основы работы в среде языка программирования С#.

Данные в С# программах.

**Задание:** Написать программу вычисления площади поверхности параллелепипеда.

**Код программы:**

using System;

class L1V2

{

public static void Main(string[] args)

{

const byte LENGTH = 9;

const float WIDTH = 7.5F;

const byte HEIGTH = 5;

Console.WriteLine("Вычисление площади поверхности параллелепипеда.");

Console.WriteLine("Длина (см) -> " + LENGTH);

Console.WriteLine("Ширина (см) -> " + WIDTH);

Console.WriteLine("Высота (см) -> " + HEIGTH);

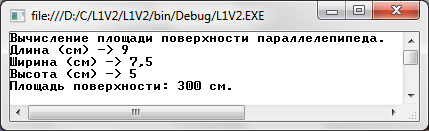
Console.WriteLine("Площадь поверхности: " + (2 \* (LENGTH \* WIDTH + LENGTH \* HEIGTH + WIDTH \* HEIGTH)) + " см.");

Console.ReadLine();

}

}

**Скриншот работы программы:**

****

2. Конструкции выбора.

**Задание:** Заданы координаты двух точек. Определить, лежат ли они на одной окружности с центром в начале координат. Результат вывести в окне сообщений. Задачу решить для точек с координатами (0;2), (2;0), (1;3), (2,2).

**Код программы:**

using System;

using System.Windows.Forms;

class L2V2

{

public static void Main(string[] args)

{

byte x1 = 0;

byte x2 = 0;

byte y1 = 0;

byte y2 = 0;

string res = "";

Console.Write("Х1 = ");

x1 = byte.Parse(Console.ReadLine());

Console.Write("Y1 = ");

y1 = byte.Parse(Console.ReadLine());

Console.Write("Х2 = ");

x2 = byte.Parse(Console.ReadLine());

Console.Write("Y2 = ");

y2 = byte.Parse(Console.ReadLine());

if (Math.Sqrt(x1 \* x1 + y1 \* y1) == Math.Sqrt(x2 \* x2 + y2 \* y2))

{

res = "Точки с заданными координатами лежат на одной окружности";

}

else

{

res = "Точки с заданными координатами не лежат на одной окружности";

}

MessageBox.Show(res,

"Результат",

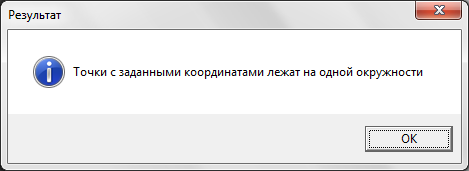
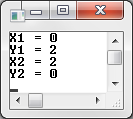
MessageBoxButtons.OK,

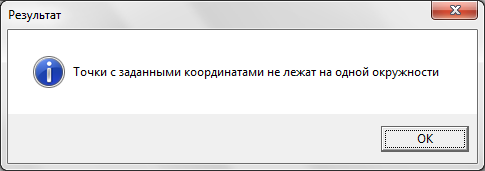
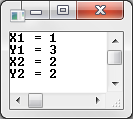
MessageBoxIcon.Information);

}

}

**Скриншоты работы программы:**





3.Циклы.

**Задание:** Вводя в цикле по 5 оценок каждого студента, подсчитать число студентов, не имеющих оценок 2 или 3. В группе учится **n** учеников.

**Код программы:**

using System;

using System.Windows.Forms;

class L3V2

{

public static void Main(string[] args)

{

byte j = 0;

byte x1 = 0;

byte x2 = 0;

byte x3 = 0;

byte x4 = 0;

byte x5 = 0;

byte k = 0;

Console.Write("Введите количество студентов: ");

j = byte.Parse(Console.ReadLine());

for (byte i = 1; i <= j; i++)

{

Console.Write("Введите первую оценку " + i + "'го студента :");

x1 = byte.Parse(Console.ReadLine());

Console.Write("Введите вторую оценку " + i + "'го студента :");

x2 = byte.Parse(Console.ReadLine());

Console.Write("Введите третью оценку " + i + "'го студента :");

x3 = byte.Parse(Console.ReadLine());

Console.Write("Введите четвёртую оценку " + i + "'го студента :");

x4 = byte.Parse(Console.ReadLine());

Console.Write("Введите пятую оценку " + i + "'го студента :");

x5 = byte.Parse(Console.ReadLine());

if (x1 > 3 && x2 > 3 && x3 > 3 && x4 > 3 && x5 > 3)

{

k++;

}

}

MessageBox.Show("Студентов, не имеющих оценок 3 и меньше = " + k,

"Результат",

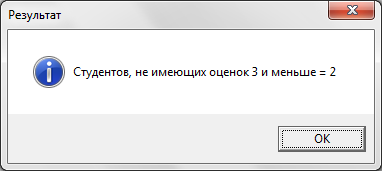
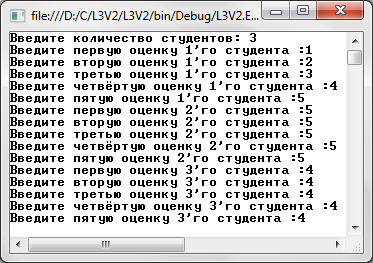
MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

}

**Скриншоты работы программы:**



4. Циклы.

**Задание:** В ЭВМ по очереди вводятся с клавиатуры фамилии спортсменов и их результаты в соревнованиях по прыжкам в длину. Число участников произвольно. Выдавать на экран лучший результат после выступления очередного спортсмена. После окончания соревнования вывести итоговое сообщение о победителе.

**Код программы:**

using System;

using System.Windows.Forms;

class L4V2

{

public static void Main(string[] args)

{

float jump\_1 = 0;

float jump\_2 = 0;

float jump\_3 = 0;

float jump\_max = 0;

float jump\_record = 0;

string name = "";

string new\_name = "";

Console.Write("Введите количество участников: ");

byte j = byte.Parse(Console.ReadLine());

for (byte i = 1; i <= j; i++)

{

Console.Write("Введите имя " + i + "'го участника :");

new\_name = Console.ReadLine();

Console.Write("Введите длинну первого прыжка " + i + "'го участника :");

jump\_1 = float.Parse(Console.ReadLine());

if (jump\_1 > jump\_max)

{

jump\_max = jump\_1;

}

Console.Write("Введите длинну второго прыжка " + i + "'го участника :");

jump\_2 = float.Parse(Console.ReadLine());

if (jump\_2 > jump\_max)

{

jump\_max = jump\_2;

}

Console.Write("Введите длинну третьего прыжка " + i + "'го участника :");

jump\_3 = float.Parse(Console.ReadLine());

if (jump\_3 > jump\_max)

{

jump\_max = jump\_3;

}

Console.WriteLine("Рекорд " + i + "'го участника по имени " + new\_name + ": " + jump\_max);

if (jump\_max > jump\_record)

{

jump\_record = jump\_max;

name = new\_name;

}

}

MessageBox.Show("Абсолютный рекорд установил участник по имени " + name + ": " + jump\_record,

"Результат",

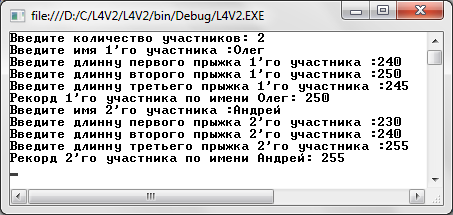
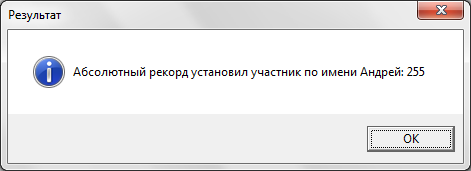
MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

}

**Скриншоты работы программы:**

5. Создание собственных методов.

**Задание:** Разработать класс «Поликлиника», который содержит информацию о трёх отделениях «Терапия», «Хирургия», «Кардиология» (стоимость общего обследования одного больного, стоимость специальных обследований одного больного, сумма на приобретение лекарственных препаратов для одного больного, средняя стоимость операции одного больного, средняя стоимость лечения одного больного). Программа должна рассчитывать общую сумму затрат на больного по каждому отделению.

**Код программы:**

using System;

using System.Windows.Forms;

class L5V2

{

public static void Main(string[] args)

{

Console.Write("Ведите название отделения (Терапия / Хирургия / Кардиология): ");

string name = Console.ReadLine().ToLower();

WhatKindOfOtdel(name);

}

public static void WhatKindOfOtdel(string name)

{

if (name == "терапия")

{

Console.Write("Ведите стоимость общего обследования: ");

int full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

int summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

CalculateTer(full\_s, summ\_lp, average);

}

if (name == "хирургия")

{

Console.Write("Ведите стоимость общего обследования: ");

int full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

int summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость операции: ");

int average\_o = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

CalculateXir(full\_s, special, summ\_lp, average\_o, average);

}

if (name == "кардиология")

{

Console.Write("Ведите стоимость общего обследования: ");

int full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

int summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

CalculateKard(full\_s, special, summ\_lp, average);

}

}

public static void CalculateTer(int full\_s, int summ\_lp, int average)

{

int rez = full\_s + summ\_lp + average;

DisplayRez(full\_s, summ\_lp, average, rez);

}

public static void CalculateXir(int full\_s, int special, int summ\_lp, int average\_o, int average)

{

int rez = full\_s + special + summ\_lp + average\_o + average;

DisplayRez(full\_s, special, summ\_lp, average\_o, average, rez);

}

public static void CalculateKard(int full\_s, int special, int summ\_lp, int average)

{

int rez = full\_s + special + summ\_lp + average;

DisplayRez(full\_s, special, summ\_lp, average, rez);

}

public static void DisplayRez(int full\_s, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Терапия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public static void DisplayRez(int full\_s, int special, int summ\_lp, int average\_o, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость операции: " + average\_o + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Хирургия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public static void DisplayRez(int full\_s,int special, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Кардиология",

MessageBoxButtons.OK,

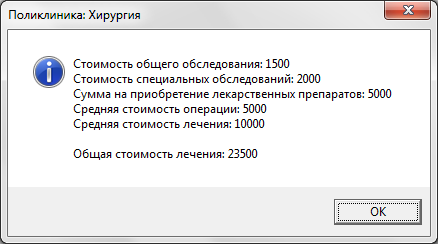
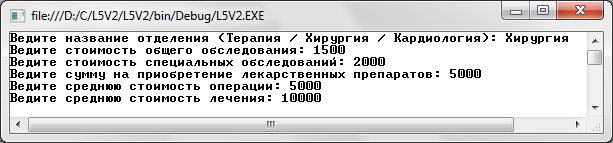
MessageBoxIcon.Information

);

}

}

**Скриншоты работы программы:**



6. Порождение объектов от порождающих классов.

**Задание:** В соответствии с вариантом практического задания №5:

1. Разработать три порождающих класса (с использованием конструкторов класса), с помощью которых клиентская программа сможет создавать объекты запрашивающие у пользователя информацию, необходимую для расчёта итоговых показателей.

2. Разработать порождающий класс «Вывод», с помощью которого клиентская программа сможет создавать объект, который будет выводить итоговую информацию. В классе определить три перегруженных конструктора.

3. Внести изменения в класс, содержащий метод Main(), разработанный в практическом задании №5, такие, чтобы можно было создавать объекты из определённых порождающих классов.

**Код программы:**

using System;

using System.Windows.Forms;

class L6V2

{

public static void Main(string[] args)

{

Console.Write("Ведите название отделения (Терапия / Хирургия / Кардиология): ");

string name = Console.ReadLine().ToLower();

if (name == "терапия")

{

CalculateTer ct = new CalculateTer();

DisplayRez dr = new DisplayRez(ct.full\_s, ct.summ\_lp, ct.average, ct.rez);

}

if (name == "хирургия")

{

CalculateXir cx = new CalculateXir();

DisplayRez dr = new DisplayRez(cx.full\_s, cx.special, cx.summ\_lp, cx.average\_o, cx.average, cx.rez);

}

if (name == "кардиология")

{

CalculateKard ck = new CalculateKard();

DisplayRez dr = new DisplayRez(ck.full\_s, ck.special, ck.summ\_lp, ck.average, ck.rez);

}

}

}

class CalculateTer

{

public int full\_s;

public int summ\_lp;

public int average;

public int rez;

public CalculateTer()

{

Console.Write("Ведите стоимость общего обследования: ");

full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + summ\_lp + average; // результат

}

}

class CalculateXir

{

public int full\_s;

public int summ\_lp;

public int average;

public int rez;

public int special;

public int average\_o;

public CalculateXir()

{

Console.Write("Ведите стоимость общего обследования: ");

full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость операции: ");

average\_o = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average\_o + average;

}

}

class CalculateKard

{

public int full\_s;

public int summ\_lp;

public int average;

public int rez;

public int special;

public CalculateKard()

{

Console.Write("Ведите стоимость общего обследования: ");

full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average;

}

}

class DisplayRez

{

public DisplayRez(int full\_s, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Терапия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average\_o, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость операции: " + average\_o + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Хирургия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Кардиология",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

}

7. Управление доступом к данным объекта.

**Задание:** По варианту практических заданий №5 и №6:

1. Модифицировать каждый из трёх порождающих классов, разработанных в практическом задании №6, в соответствии со следующими требованиями:

* Установить для всех переменных экземпляра спецификатор доступности **private**;
* Разработать свойства для каждой переменной экземпляра, включив в них методы аксессора **get()** и **set()**;
* Включить в методы аксессора **set()** процедуры проверки значений переменных экземпляра (исключить возможность ввода недопустимых исходных данных);
* В методе расчёта результирующих показателей каждого класса внести коррективы по использованию имён свойств вместо имён закрытых переменных экземпляра.

2. Заменить в программе, содержащей начальный класс, использование закрытых переменных экземпляра на использование свойств.

**Код программы:**

using System;

using System.Windows.Forms;

class L7V2

{

public static void Main(string[] args)

{

Console.Write("Ведите название отделения (Терапия / Хирургия / Кардиология): ");

string name = Console.ReadLine().ToLower();

if (name == "терапия")

{

CalculateTer ct = new CalculateTer();

DisplayRez dr = new DisplayRez(ct.Full\_s, ct.Summ\_lp, ct.Average, ct.Rez);

}

if (name == "хирургия")

{

CalculateXir cx = new CalculateXir();

DisplayRez dr = new DisplayRez(cx.Full\_s, cx.Special, cx.Summ\_lp, cx.Average\_o, cx.Average, cx.Rez);

}

if (name == "кардиология")

{

CalculateKard ck = new CalculateKard();

DisplayRez dr = new DisplayRez(ck.Full\_s, ck.Special, ck.Summ\_lp, ck.Average, ck.Rez);

}

}

}

class CalculateTer

{

private int full\_s = 0;

private int summ\_lp = 0;

private int average = 0;

private int rez = 0;

public int Full\_s

{

get { return full\_s; }

set {

if (value < 0)

{ MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0); }

else

full\_s = value; }

}

public int Summ\_lp

{

get { return summ\_lp; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

summ\_lp = value;

}

}

public int Average

{

get { return average; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average = value;

}

}

public int Rez

{

get { return rez; }

}

public CalculateTer()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + summ\_lp + average; // результат

}

}

class CalculateXir

{

private int full\_s;

private int summ\_lp;

private int average;

private int rez;

private int special;

private int average\_o;

public int Full\_s

{

get { return full\_s; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

full\_s = value;

}

}

public int Summ\_lp

{

get { return summ\_lp; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

summ\_lp = value;

}

}

public int Average

{

get { return average; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average = value;

}

}

public int Rez

{

get { return rez; }

}

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public int Average\_o

{

get { return average\_o; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average\_o = value;

}

}

public CalculateXir()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость операции: ");

Average\_o = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average\_o + average;

}

}

class CalculateKard

{

private int full\_s;

private int summ\_lp;

private int average;

private int rez;

private int special;

public int Full\_s

{

get { return full\_s; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

full\_s = value;

}

}

public int Summ\_lp

{

get { return summ\_lp; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

summ\_lp = value;

}

}

public int Average

{

get { return average; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average = value;

}

}

public int Rez

{

get { return rez; }

}

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public CalculateKard()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average;

}

}

class DisplayRez

{

public DisplayRez(int full\_s, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Терапия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average\_o, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость операции: " + average\_o + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Хирургия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Кардиология",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

}

8. Наследование.

**Задание:** По варианту практических заданий №5, 6, 7:

1. Разработать базовый класс.

2. Модифицировать порождающие классы (по заданию №7) в производные от базового.

**Код программы:**

**Calculate.cs (базовый класс)**

using System;

using System.Windows.Forms;

abstract class Calculate

{

protected int full\_s = 0;

protected int summ\_lp = 0;

protected int average = 0;

protected int rez = 0;

public int Full\_s

{

get { return full\_s; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

full\_s = value;

}

}

public int Summ\_lp

{

get { return summ\_lp; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

summ\_lp = value;

}

}

public int Average

{

get { return average; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average = value;

}

}

public int Rez

{

get { return rez; }

}

}

**Code File1.cs**

using System;

using System.Windows.Forms;

class L8V2

{

public static void Main(string[] args)

{

Console.Write("Ведите название отделения (Терапия / Хирургия / Кардиология): ");

string name = Console.ReadLine().ToLower();

if (name == "терапия")

{

CalculateTer ct = new CalculateTer();

DisplayRez dr = new DisplayRez(ct.Full\_s, ct.Summ\_lp, ct.Average, ct.Rez);

}

if (name == "хирургия")

{

CalculateXir cx = new CalculateXir();

DisplayRez dr = new DisplayRez(cx.Full\_s, cx.Special, cx.Summ\_lp, cx.Average\_o, cx.Average, cx.Rez);

}

if (name == "кардиология")

{

CalculateKard ck = new CalculateKard();

DisplayRez dr = new DisplayRez(ck.Full\_s, ck.Special, ck.Summ\_lp, ck.Average, ck.Rez);

}

}

}

class CalculateTer : Calculate

{

public CalculateTer()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + summ\_lp + average; // результат

}

}

class CalculateXir : Calculate

{

private int special;

private int average\_o;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public int Average\_o

{

get { return average\_o; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average\_o = value;

}

}

public CalculateXir()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость операции: ");

Average\_o = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average\_o + average;

}

}

class CalculateKard : Calculate

{

private int special;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public CalculateKard()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average;

}

}

class DisplayRez

{

public DisplayRez(int full\_s, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Терапия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average\_o, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость операции: " + average\_o + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Хирургия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Кардиология",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

}

9. Массивы.

**Задание:** По варианту практических заданий №5, 6, 7, 8:

Модифицировать клиентскую программу, содержащую метод Main(), с целью использования месива для расчёта общей суммы затрат на больных по поликлинике.

**Код программы:**

using System;

using System.Windows.Forms;

class L9V2

{

public static void Main(string[] args)

{

string confirmation = "";

int i = 0;

int[] all\_cost = new int[10];

int full\_cost = 0;

do

{

Console.Write("Ведите название отделения (Терапия / Хирургия / Кардиология): ");

string name = Console.ReadLine().ToLower();

if (name == "терапия")

{

CalculateTer ct = new CalculateTer();

all\_cost[i] = ct.Rez;

DisplayRez dr = new DisplayRez(ct.Full\_s, ct.Summ\_lp, ct.Average, ct.Rez);

}

if (name == "хирургия")

{

CalculateXir cx = new CalculateXir();

all\_cost[i] = cx.Rez;

DisplayRez dr = new DisplayRez(cx.Full\_s, cx.Special, cx.Summ\_lp, cx.Average\_o, cx.Average, cx.Rez);

}

if (name == "кардиология")

{

CalculateKard ck = new CalculateKard();

all\_cost[i] = ck.Rez;

DisplayRez dr = new DisplayRez(ck.Full\_s, ck.Special, ck.Summ\_lp, ck.Average, ck.Rez);

}

i++;

Console.Write("Продолжить расчёт для следующего пациента? (ДА / НЕТ): ");

confirmation = Console.ReadLine().ToLower();

} while (confirmation == "да");

for (int j = 0; j < i; j++)

{

full\_cost += all\_cost[j];

}

MessageBox.Show

(

"Общая стоимость лечения всех пациентов больницы: " + full\_cost,

"Поликлиника",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

}

class CalculateTer : Calculate

{

public CalculateTer()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + summ\_lp + average; // результат

}

}

class CalculateXir : Calculate

{

private int special;

private int average\_o;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public int Average\_o

{

get { return average\_o; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average\_o = value;

}

}

public CalculateXir()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость операции: ");

Average\_o = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average\_o + average;

}

}

class CalculateKard : Calculate

{

private int special;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public CalculateKard()

{

Console.Write("Ведите стоимость общего обследования: ");

Full\_s = int.Parse(Console.ReadLine());

Console.Write("Ведите стоимость специальных обследований: ");

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

Console.Write("Ведите сумму на приобретение лекарственных препаратов: ");

Summ\_lp = int.Parse(Console.ReadLine());

Console.Write("Ведите среднюю стоимость лечения: ");

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

rez = full\_s + special + summ\_lp + average;

}

}

class DisplayRez

{

public DisplayRez(int full\_s, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Терапия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average\_o, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость операции: " + average\_o + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Хирургия",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

public DisplayRez(int full\_s, int special, int summ\_lp, int average, int rez)

{

MessageBox.Show

(

"Cтоимость общего обследования: " + full\_s + "\n" +

"Cтоимость специальных обследований: " + special + "\n" +

"Cумма на приобретение лекарственных препаратов: " + summ\_lp + "\n" +

"Средняя стоимость лечения: " + average + "\n" + "\n" +

"Общая стоимость лечения: " + rez,

"Поликлиника: Кардиология",

MessageBoxButtons.OK,

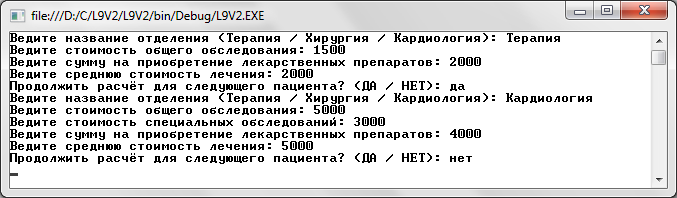
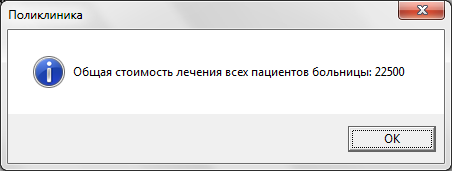
MessageBoxIcon.Information

);

}

}

**Скриншоты работы программы:**

10. Разработка графического интерфейса пользователя.

**Задание:** По варианту практических заданий №5, 6, 7, 8:

1. Спроектировать графический интерфейс пользователя.

2. Модифицировать производные классы, перегрузив расчётные методы таким образом, чтобы пользователь мог взаимодействовать с графическим интерфейсом (программа должна воспринимать исходные данные из текстовых полей, а не с консоли).

3. Написать необходимые обработчики событий для всех управляющих элементов графического интерфейса.

**Код программы:**

**GUI.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace L10V2

{

public partial class GUI : Form

{

public GUI()

{

InitializeComponent();

}

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

{

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = false;

label7.Visible = false;

textBox\_full\_s.Visible = true;

textBox\_summ\_lp.Visible = true;

textBox\_average.Visible = true;

textBox1.Visible = false;

textBox2.Visible = false;

Category.Text = "Терапия:";

}

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

{

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = true;

label7.Visible = true;

textBox\_full\_s.Visible = true;

textBox\_summ\_lp.Visible = true;

textBox\_average.Visible = true;

textBox1.Visible = true;

textBox2.Visible = true;

Category.Text = "Хирургия:";

}

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

{

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = true;

label7.Visible = false;

textBox\_full\_s.Visible = true;

textBox\_summ\_lp.Visible = true;

textBox\_average.Visible = true;

textBox1.Visible = true;

textBox2.Visible = false;

Category.Text = "Кардиология:";

}

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

{

if (Category.Text == "Терапия:")

{

if (textBox\_full\_s.Text == "" || textBox\_summ\_lp.Text == "" || textBox\_average.Text == "")

{

MessageBox.Show("Необходимо ввести сумму", "Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

}

else

{

CalculateTer ct = new CalculateTer(int.Parse(textBox\_full\_s.Text), int.Parse(textBox\_summ\_lp.Text), int.Parse(textBox\_average.Text));

label8.Text = Convert.ToString(ct.Rez);

}

}

if (Category.Text == "Хирургия:")

{

if (textBox\_full\_s.Text == "" || textBox\_summ\_lp.Text == "" || textBox\_average.Text == "" || textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("Необходимо ввести сумму", "Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

}

else

{

CalculateXir cx = new CalculateXir(int.Parse(textBox\_full\_s.Text), int.Parse(textBox1.Text), int.Parse(textBox\_summ\_lp.Text), int.Parse(textBox2.Text), int.Parse(textBox\_average.Text));

label8.Text = Convert.ToString(cx.Rez);

}

}

if (Category.Text == "Кардиология:")

{

if (textBox\_full\_s.Text == "" || textBox\_summ\_lp.Text == "" || textBox\_average.Text == "" || textBox1.Text == "")

{

MessageBox.Show("Необходимо ввести сумму", "Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

}

else

{

CalculateKard ck = new CalculateKard(int.Parse(textBox\_full\_s.Text), int.Parse(textBox1.Text), int.Parse(textBox\_summ\_lp.Text), int.Parse(textBox\_average.Text));

label8.Text = Convert.ToString(ck.Rez);

}

}

}

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

{

radioButton1.Checked = true;

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = false;

label7.Visible = false;

textBox\_full\_s.Visible = true;

textBox\_summ\_lp.Visible = true;

textBox\_average.Visible = true;

textBox1.Visible = false;

textBox2.Visible = false;

Category.Text = "Терапия:";

textBox\_full\_s.Text = "";

textBox\_summ\_lp.Text = "";

textBox\_average.Text = "";

label8.Text = "0";

}

}

}

**DrawGUI.cs**

using System;

using System.Windows.Forms;

class DrawGUI

{

public static void Main(string[] args)

{

L10V2.GUI x = new L10V2.GUI();

Application.Run(x);

}

}

**DrawGUI.cs**

using System;

using System.Windows.Forms;

class L101V2

{

}

class CalculateTer : Calculate

{

public CalculateTer(int full\_s, int summ\_lp, int average)

{

Full\_s = full\_s;

Summ\_lp = summ\_lp;

Average = average;

rez = Full\_s + Summ\_lp + Average;

}

}

class CalculateXir : Calculate

{

private int special;

private int average\_o;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public int Average\_o

{

get { return average\_o; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

average\_o = value;

}

}

public CalculateXir(int full\_s, int special, int summ\_lp, int average\_o, int average)

{

Full\_s = full\_s;

Summ\_lp = summ\_lp;

Average = average;

Special = special;

Average\_o = average\_o;

rez = Full\_s + Special + Summ\_lp + Average\_o + Average;

}

}

class CalculateKard : Calculate

{

private int special;

public int Special

{

get { return special; }

set

{

if (value < 0)

{

MessageBox.Show("Неверное значение - " + value,

"Расчёт стоимости",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

Environment.Exit(0);

}

else

special = value;

}

}

public CalculateKard(int full\_s, int special, int summ\_lp, int average)

{

Full\_s = full\_s;

Summ\_lp = summ\_lp;

Average = average;

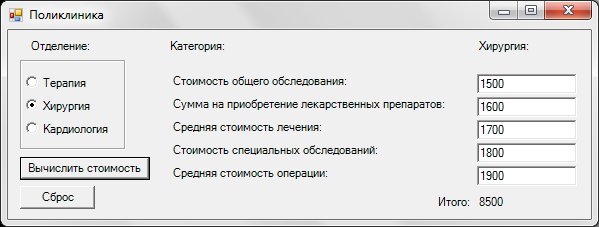
Special = special;

rez = Full\_s + Special + Summ\_lp + Average;

}

}

**Скриншот работы программы:**



11. Windows-приложения на C#.

**Задание:**

1. Разработать проект Калькулятор, выполняющий арифметические операции.
2. Разработать эффект анимации в виде мигающего заголовка формы и хотя бы одного из элементов.
3. Изменить код обработчика события Tick, чтобы анимационный объект изменялся через заданное число интервалов.
4. Добавить в проект кнопку «Старт / Стоп анимация».
5. Добавить в проект звуковой сигнал.
6. Создать анимацию цвета фона формы и текста элементов управления в проекте.
7. Для задачи лаб. работы №1 по своему варианту разработать графический интерфейс с эффектами анимации и звука.
8. Заголовок формы выполнить в виде бегущей строки.

**Код программы:**

**FormCalc.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace L11V2

{

public partial class FormCalc : Form

{

public double store1 = 0;

public double store2 = 0;

public double rez = 0;

public FormCalc()

{

InitializeComponent();

}

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

{

store1 = double.Parse(textBox1.Text);

textBox1.Text = "";

label1.Text = "+";

}

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

{

store1 = double.Parse(textBox1.Text);

textBox1.Text = "";

label1.Text = "-";

}

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

{

store1 = double.Parse(textBox1.Text);

textBox1.Text = "";

label1.Text = "\*";

}

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

{

store1 = double.Parse(textBox1.Text);

textBox1.Text = "";

label1.Text = "/";

}

private void button5\_Click(object sender, EventArgs e)

{

store2 = double.Parse(textBox1.Text);

if (label1.Text == "+")

{

rez = store1 + store2;

textBox1.Text = Convert.ToString(rez);

label1.Text = "";

}

if (label1.Text == "-")

{

rez = store1 - store2;

textBox1.Text = Convert.ToString(rez);

label1.Text = "";

}

if (label1.Text == "\*")

{

rez = store1 \* store2;

textBox1.Text = Convert.ToString(rez);

label1.Text = "";

}

if (label1.Text == "/")

{

rez = store1 / store2;

textBox1.Text = Convert.ToString(rez);

label1.Text = "";

}

}

private void button7\_Click(object sender, EventArgs e)

{

store1 = 0;

store2 = 0;

rez = 0;

label1.Text = "";

textBox1.Text = "";

}

bool myText = false;

private void timer1\_Tick(object sender, EventArgs e)

{

if (myText == false)

{

Microsoft.VisualBasic.Interaction.Beep();

this.BackColor = System.Drawing.Color.White;

this.ForeColor = System.Drawing.Color.Blue;

this.Text = "Калькулятор";

label2.Text = "Текст";

myText = true;

}

else

{

this.BackColor = System.Drawing.Color.Red;

this.ForeColor = System.Drawing.Color.Yellow;

this.Text = "Calculator";

label2.Text = "Text";

myText = false;

}

}

bool OffOn = false;

private void button6\_Click(object sender, EventArgs e)

{

if (OffOn == false)

{

timer1.Enabled = false;

OffOn = true;

}

else

{

timer1.Enabled = true;

OffOn = false;

}

}

private void button8\_Click(object sender, EventArgs e)

{

L11V2.L1\_V2 y = new L11V2.L1\_V2();

y.Show();

}

}

}

**L1\_V2.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace L11V2

{

public partial class L1\_V2 : Form

{

public L1\_V2()

{

InitializeComponent();

}

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

{

Microsoft.VisualBasic.Interaction.Beep();

label5.Text = Convert.ToString(2 \* (double.Parse(textBox1.Text) \* double.Parse(textBox2.Text) + double.Parse(textBox1.Text) \* double.Parse(textBox3.Text) + double.Parse(textBox2.Text) \* double.Parse(textBox3.Text)));

}

public string temp = "";

public string ReTemp = "";

private void timer1\_Tick(object sender, EventArgs e)

{

temp = this.Text.Substring(0, 1);

ReTemp = this.Text.Substring(1);

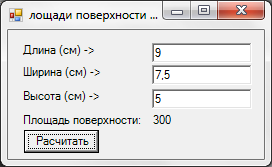
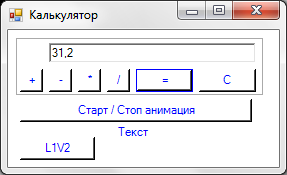
this.Text = ReTemp + temp;

}

}

}

**Скриншоты работы программы:**



12. Анимационные программы.

**Задание:**

1. Проверить 12.1 на компьютере.
2. Доработать проект, усложнив рисунок таким образом, чтобы на экране отображался работающий «секундомер». Линии придать вид стрелки, на циеферблате вывести шкалу и значения секунд. Подобрать цветовую гамму.
3. Начало движения должно осуществляться из нового положения (по сравнению с примером 12.1).

**Код программы:**

**FormCalc.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;