Лабораторная работа №1

Выполнил: Олишкевич Игорь Русланович.

Тема: решение квадратного уравнения.

Задание: требуется разработать приложение, которое по заданным значениям коэффициентов a, b и c квадратного уравнения (значения вводятся с клавиатуры пользователем) вычисляет и отображает на экране корни уравнения.

Листинг программы:

using System;

namespace laboratorn\_1

{

class Program

{

static void Main(string[] args)

{

while (true)

{

double a, b, c;

Console.WriteLine("Введите коэффициенты уравнения (a,b,c):");

a = double.Parse(Console.ReadLine());

b = double.Parse(Console.ReadLine());

c = double.Parse(Console.ReadLine());

Console.WriteLine("a = {0},b = {1},c = {2}", a, b, c);

double D = b \* b - 4 \* a \* c;

if (D > 0)

{

D = Math.Sqrt(D);

double k1 = (-b + D) / 2 \* a;

double k2 = (-b - D) / 2 \* a;

Console.WriteLine("(D = {0}) Первый корень {1}, второй корень {2}", D, k1, k2);

}

else if (D == 0)

{

double k1 = -b / 2 \* a;

Console.WriteLine("(D = {0}) Корень уравнения {1}", D, k1);

}

else

{

Console.WriteLine("кореней нет (D = {0})", D);

}

}

}

}

}

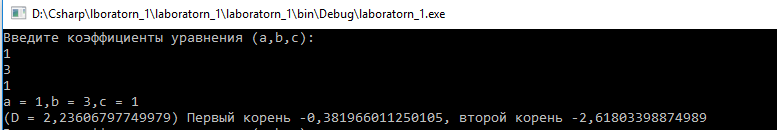
Скриншот выполнения:

Рисунок 1 – Выполнение программы

Алгоритм: ввод коэффициентов (a, b, c), вычисление дискриминанта. С помощью условного оператора проверяем чему равен дискриминант, если дискриминант меньше нуля, вычисляем корни уравнения, выводим результат. При дискриминанте равном нулю вычисляется корень уравнения и выводим результат вычислений. Если дискриминант меньше нуля выводится информация о отсутствии корней.

Контрольные вопросы:

1. C, C#, C++, F#, VB.Net
2. Целочисленные, с плавающей запятой, строковые переменные.
3. «Файл», «Создать проект», выбрать требуемую конфигурацию и путь для сохранения файлов проекта.
4. Оператор присваивания, составной оператор, операторы выбора, операторы цикла, оператор перехода.
5. С помощью кнопки «Debag» или «вручную», передав файл (листинг программы) компилятору .NET csc.exe.

Лабораторная работа №2

Выполнил: Олишкевич Игорь Русланович.

Тема: Разработка линейных программ

Цель:

Образовательная: сформировать умения создавать линейные программы в среде в Visual Studio,

Развивающая:

1. научить использовать основные операторы, типы данных и конструкции;
2. создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

1. создать условия для воспитания в обучающихся средствами урока уверенности в своих силах;
2. создать условия для воспитания сознательного и серьёзного отношения обучающихся к учебной дисциплине, убеждая их в том, что полученные знания пригодятся им в будущей деятельности.

Задания:

1. Запрашивает с клавиатуры три вещественных числа и выводит следующее сообщение (*a+(b+c))=(a+c+b)* (вещественные числа выводятся с точностью до 4 знаков после запятой);

Алгоритм: получение данных из консоли, преобразование в double, вывод с помощью форматирования статического метода Console.WriteLine.

1. Дано трехзначное число. Найти число, полученное при прочтении его цифр справа налево;  
   Алгоритм: получение данных из консоли, преобразование и иницализация строки в массив символов; вывод в консоли массива в обратном порядке.
2. Периметр треугольника, заданного координатами вершин x1, y1, x2, y2, x3, y3;  
   Алгоритм: ввод данных в консоль, исполнение уравнений, вывод результатов в консоль.
3. Определить силу притяжения *F* между телами массы *т*1 и *т*2, находящимися на расстоянии *r* друг от друга.  
   Алгоритм: ввод данных в консоль, исполнение уравнений, вывод результатов в консоль.
4. Пусть даны четыре целых числа (hour, min, sec, time). Первые три из них (hour, min, sec) - это время запуска ракеты в часах, минутах и секундах, четвертое (time) определяет время полета в секундах. Найдите и напечатайте время возвращения ракеты на землю.  
   Алгоритм: ввод данных в консоль, перевод всех единиц в секунды с подсчетом, преобразование обратно в «hh:mm:ss», вывод результатов в консоль.
5. ;  
   Алгоритм: ввод данных в консоль, исполнение уравнений, вывод результатов в консоль.
6. *а*64 за шесть операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.  
   Алгоритм: ввод данных в консоль, исполнение уравнений, вывод результатов в консоль.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Задание 1:");

double a, b, c;

Console.WriteLine("Введите a, b и c: ");

Console.WriteLine("a = {0}", a = Convert.ToDouble(Console.ReadLine()));

Console.WriteLine("b = {0}", b = Convert.ToDouble(Console.ReadLine()));

Console.WriteLine("c = {0}", c = Convert.ToDouble(Console.ReadLine()));

Console.WriteLine("Полученное выражение: ");

Console.WriteLine("({0:0.0000}+({1:0.0000}+{2:0.0000}))=({0:0.0000}+{2:0.0000}+{1:0.0000})", a, b, c);

Console.WriteLine("Задание 2:");

Console.WriteLine("Введите трёхзначное число: ");

string three = Console.ReadLine();

char[] arr = three.ToCharArray();

Array.Reverse(arr);

Console.WriteLine(arr);

Console.WriteLine("Задание 3:");

double x1, x2, x3, y1, y2, y3;

Console.WriteLine("Задайте координаты треугольника: x1, y1, x2, y2, x3, y3");

Console.WriteLine("x1 = {0}", x1 = double.Parse(Console.ReadLine()));

Console.WriteLine("y1 = {0}", y1 = double.Parse(Console.ReadLine()));

Console.WriteLine("x2 = {0}", x2 = double.Parse(Console.ReadLine()));

Console.WriteLine("y2 = {0}", y2 = double.Parse(Console.ReadLine()));

Console.WriteLine("x3 = {0}", x3 = double.Parse(Console.ReadLine()));

Console.WriteLine("y3 = {0}", y3 = double.Parse(Console.ReadLine()));

double x1y1x2y2 = Math.Sqrt(Math.Pow((x2 - x1), 2) + (Math.Pow((y2 - y1), 2)));

double x2y2x3y3 = Math.Sqrt(Math.Pow((x3 - x2), 2) + (Math.Pow((y3 - y2), 2)));

double x3y3x1y1 = Math.Sqrt(Math.Pow((x3 - x1), 2) + (Math.Pow((y3 - y1), 2)));

double Px1y1x2y2x3y3 = x1y1x2y2 + x2y2x3y3 + x3y3x1y1;

Console.WriteLine("Периметр треугольника = {0}", Px1y1x2y2x3y3);

Console.WriteLine("Задание 4:");

double G = (Math.Pow(6.7385, (10 ^ -11))), F, m1, m2, r;

Console.WriteLine("Задайте вес тел и расстояние их друг от друга: m1, m2, r");

Console.WriteLine("m1 = {0}", m1 = double.Parse(Console.ReadLine()));

Console.WriteLine("m2 = {0}", m2 = double.Parse(Console.ReadLine()));

Console.WriteLine("r = {0}", r = double.Parse(Console.ReadLine()));

Console.WriteLine("Сила притяжения = {0} кг\*м/с^2", F = G \* ((m1 \* m2)/(Math.Pow(r, 2))));

Console.WriteLine("Задание 5:");

int hour, min, sec, time, time1, hour1, min1, sec1;

Console.WriteLine("Введите часы, минуты и секунды запуска ракеты: hour, min, sec. После - время полета в секундах: time");

Console.WriteLine("hour = {0}", hour = Int32.Parse(Console.ReadLine()));

Console.WriteLine("min = {0}", min = Int32.Parse(Console.ReadLine()));

Console.WriteLine("sec = {0}", sec = Int32.Parse(Console.ReadLine()));

Console.WriteLine("time = {0}", time = Int32.Parse(Console.ReadLine()));

time1 = hour \* 360 + min \* 60 + sec + time;

Console.WriteLine("Общее время полета в секундах = {0}", time1);

hour1 = time1 / 360;

min1 = (time1 % 360) / 60;

sec1 = (time1 - hour \* 360 - min \* 60) % 60;

Console.WriteLine("Время прилета {0:00}:{1:00}:{2:00}", hour1, min1, sec1);

Console.WriteLine("Задание 6:");

double e, x, y;

e = 2.7;

x = 6.4;

y = (Math.Pow(e, x)) / (Math.Cos(Math.Sqrt(x - 1))) + (2 \* Math.Atan(Math.Pow(x, 2))) / (1 - x);

Console.WriteLine("Значение функции y = {0}", y);

Console.WriteLine("Задание 7:");

decimal a7;

Console.WriteLine("Введите а: ");

Console.WriteLine("a = {0}", a7 = decimal.Parse(Console.ReadLine()));

a7 = a7 \* a7;

a7 = a7 \* a7;

a7 = a7 \* a7;

a7 = a7 \* a7;

a7 = a7 \* a7;

a7 = a7 \* a7;

Console.WriteLine("a^64 = {0}", a7);

}

}

}

Скриншот выполнения:

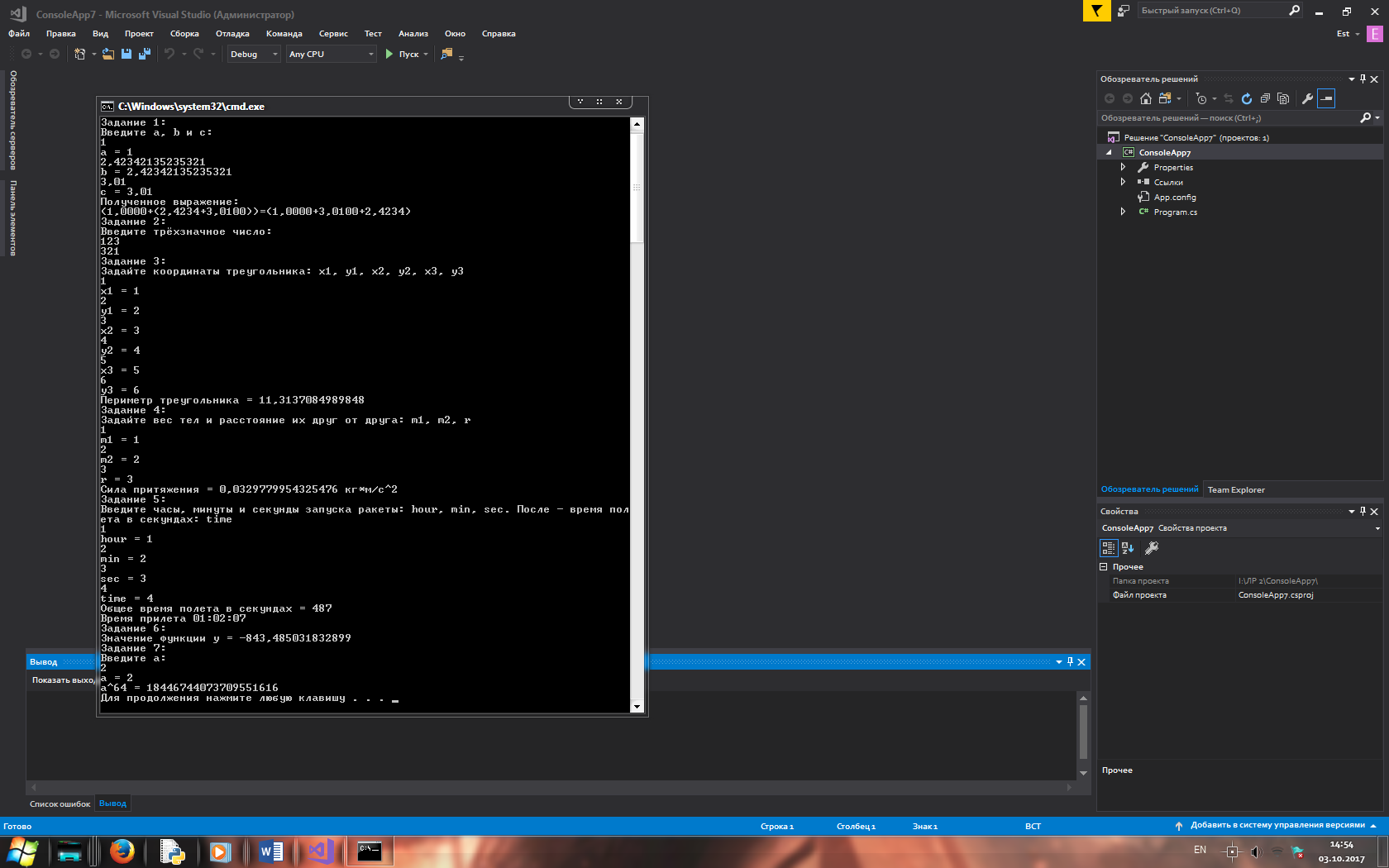


Рисунок 1 – Выполнение программы

Контрольные вопросы:

1. Какова общая структура программы на С#?

Программа на языке C# может состоять из одного или нескольких файлов. Каждый файл может содержать нуль или несколько пространств имен. Пространство имен может содержать типы, такие как классы, структуры, интерфейсы, перечисления и делегаты, а также другие пространства имен.

1. Существуют ли ограничения на множество целых чисел, используемых в языке С#?

Да, существуют.

1. Какие формы записи вещественных чисел используются в языке С#?
2. Из каких символов может состоять идентификатор?

Идентификатор может состоять из любых символов латиницы и цифр, начинаясь только с буквы латиницы, а также запрещено использовать спецсимволы «@\*\|\^$#», например.

1. Чем вызвана необходимость использования комментариев в программе?

Необходимость использования комментариев вызвана криком о помощи программистов, дорабатывающих существующий исходный код какой-либо программы. Появление комментариев прямо в исходном коде позволило пояснять от себя, на «человеческом» языке, то, что происходит в программе. Благодаря комментариям можно сократить в разы время на понимание алгоритмов и структур данных используемых в программе.

1. Каково назначение методов write и writeline?

Методы write и writeline осуществляют вывод информации в консоль (write – без переноса каретки, writeline – с переносом).

7) Перечислите способы вывода данных на дисплей?

С помощью консоли, Windows Forms, Xamarin Form, UWP Forms, сторонних библиотек.

8) Перечислите основные управляющие последовательности языка С#?

1. \a - Звонок (предупреждение)
2. \b - Backspace
3. \f - Перевод страницы
4. \n - Новая строка
5. \r - Возврат каретки
6. \t - Горизонтальная табуляция
7. \v - Вертикальная табуляция
8. \' - Одиночная кавычка
9. \" - Двойная кавычка
10. \\ - Обратная косая черта

9) Назовите основные методы класса Math языка С#?

Math.Abs – Возвращает модуль числа;

Math.Cos – Возвращает косинус заданного числа;

Math.Log – Возвращает натуральны логарифм числа;

Math.Pow – Возвращает число в заданной степени.

10) Какой метод класса Console используется для того, чтобы вводимые данные читались с новой строки?

Console.ReadLine();

11) Что такое выражения в языке С#?

Выражение представляет собой последовательность из одного или нескольких операндов и от нуля до нескольких операторов, которую можно вычислить, получив в результате одно значение, объект, метод или пространство имен. Выражение может состоять из литерала, вызова метода, оператора или его операндов, а также из простого имени. Простые имена могут быть именами переменной, элемента типа, параметра метода, пространства имен или типа.

12) Понятие преобразования типов в языке С#?

Преобразование типов — это процесс изменения типа значения. Например, можно преобразовать строку «1234» в число. Кроме того, можно преобразовать данные любого типа в тип String. Преобразование некоторых типов невозможно.

13) Назовите формы записи операции инкремента и декремента языка С#?

Постфиксная и префиксная «i++;++i;i--; --i»

Лабораторная работа №3

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка программ с использованием операторов ветвления и цикла.

Цель:

Образовательная:

1. сформировать умения разрабатывать программы с использованием операторов ветвления в среде в Visual Studio,
2. сформировать умения разрабатывать программы с использованием операторов цикла в среде в Visual Studio,

Развивающая:

1. научить использовать основные операторы, типы данных и конструкции,
2. создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

1. создать условия для воспитания в обучающихся средствами урока уверенности в своих силах;
2. создать условия для воспитания сознательного и серьёзного отношения обучающихся к учебной дисциплине, убеждая их в том, что полученные знания пригодятся им в будущей деятельности;

Задания:

1. {(x - 2)^2 + 6, 1 < x < 2  
   y = {  
    {ln(x + 3 \* sqrt(x)), x => 2;  
   Алгоритм: ввод данных, преобразование в вещественный тип и вычисление в зависимости от введенного значения.
2. Написать программу, которая определяет: кратна ли трем сумма цифр двухзначного числа.  
   Алгоритм: ввод данных, преобразование в массив символов, подсчет суммы каждого символа в преобразованном числовом значении, проверка условия, вывод результата.
3. Дана точка на плоскости с координатами(х, у). Составить программу, которая выдает одно из сообщений "Да", "Нет", "На границе" в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе.  
   Алгоритм: ввод данных, преобразование в тип с плавающей запятой, вычисление вхождения в заданную область по формуле, вывод результата.
4. Составить программу(при решении данных задач использовать оператор switch или вложенные операторы if): с некоторой даты по настоящий день прошло n дней, найти неизвестную дату.  
   Алгоритм: ввод данных, исполнение соответстующего блока кода, в зависимости от введёных данных, вывод.
5. Составить программу решения задачи: в соревнованиях по фигурному катанию N судей независимо выставляют оценки спортсмену. Затем из объявленных оценок удаляют самую высокую (одну, если самую высокую оценку выставили несколько судей). Аналогично поступают с самой низкой оценкой. Для оставшихся оценок вычисляется среднее арифметическое, которое и становится зачетной оценкой. По заданным оценкам судей определить зачетную оценку спортсмена.  
   Алгоритм: ввод данных, задание массива, нахождение минимально и максимального элемента с индексами, вычисление значения, вывод.
6. Составить программу решения задачи: по дате (месяц и день) рождения определите знак Зодиака.  
   Алгоритм: ввод данных, исполнение соответстующего блока кода, в зависимости от введёных данных, вывод.
7. Даны цифры двух целых чисел: двузначного а2а1 и однозначного b, где а1 – число единиц, а2 – число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое – двузначное число и число-результат не определить; условный оператор не использовать.  
   Алгоритм: ввод данных, исполнение соответстующего блока кода, в зависимости от введёных данных, вывод.
8. Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for): только положительные целые числа из диапазона от А до В;  
   Алгоритм: ввод данных, исполнение соответстующего блока кода, в зависимости от введёных данных, вывод.
9. Вывести на экран числа следующим образом:  
   1  
   1 2  
   1 2 3  
   1 2 3 4  
   1 2 3 4 5  
   Алгоритм: исполнение циклов, вывод.
10. Решите задачу: дано натуральное число n. Получить все натуральные числа, меньшие n и взаимно простые с ним (два натуральных числа называются взаимно простыми, если их наибольший общий делитель равен 1).  
    Алгоритм: ввод данных, исполнение соответстующего блока кода, в зависимости от введёных данных, вывод.
11. Дано натуральное число. Определить: сколько раз в нем встречаются цифры х и у.  
    Алгоритм: ввод данных, нахождение количества, вывод.
12. Решите задачу: известны оценки каждого из студентов университета по физике. Посчитать количество пятерок, количество четверок, количество троек и количество двоек.

Алгоритм: ввод данных, задание массива, нахождение значений, вывод.

Листинг программы:  
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

class Program

{

public static int NOD(int a, int b)

{

if (a == b)

return a;

else

if (a > b)

return NOD(a - b, b);

else

return NOD(b - a, a);

}

static void Main(string[] args)

{

Console.WriteLine("Задание 1");

start1: Console.Write("Введите х для промежутка (1; +∞): ");

double y1 = 0, x1 = 0;

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

if (x1 >= 2)

{

y1 = Math.Log10(x1 + 3 \* Math.Sqrt(x1));

}

else if (x1 > 1 && x1 < 2)

{

y1 = Math.Pow((x1 - 2), 2) + 6;

}

else

{

goto start1;

}

Console.WriteLine("y1 = {0}", y1);

Console.WriteLine();

Console.WriteLine("Задание 2");

Console.WriteLine("Введите двухзначное число");

string two = Console.ReadLine();

double buf = two[0] + two[1] - 96;

Console.WriteLine("Сумма цифр = {0}", buf);

if (buf % 3 == 0)

{

Console.WriteLine("Сумма цифр кратна трем");

}

else

{

Console.WriteLine("Сумма цифр не кратна трем");

}

Console.WriteLine();

Console.WriteLine("Задание 3");

Console.Write("x=");

double x3, y;

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

Console.Write("y=");

y = double.Parse(Console.ReadLine());

if (x3 < 50 && x3 > -50 && y < 25 && y > -25)

Console.WriteLine("Да");

else if (((x3 == 50 || x3 == -50) && (y <= 25 && y >= -25)) || ((x3 <= 50 && x3 >= -50) && (y == 25 || y == -25)))

Console.WriteLine("На границе");

else Console.WriteLine("Нет");

Console.WriteLine();

int n, day4, month4, year4, all, day4new = 0, month4new = 0, year4new = 0;

Console.WriteLine("Задание 4");

Console.Write("Введите количество пройденных дней: n = ");

n = Int32.Parse(Console.ReadLine());

DateTime dateGregorian = DateTime.Today;

day4 = dateGregorian.Day;

month4 = dateGregorian.Month;

year4 = dateGregorian.Year;

all = day4 + month4 \* 30 + year4 \* 365 - n;

if (all > 365)

{

year4new = all / 365;

all = all - year4new \* 365;

}

if (all > 30)

{

month4new = all / 30;

day4new = all - month4new \* 30;

}

if (day4new == 0)

{

day4new = 30;

month4new = month4new - 1;

}

Console.WriteLine("Полученная дата: {0:00}.{1:00}.{2:0000}", day4new, month4new, year4new);

Console.WriteLine();

Console.WriteLine("Задание 5");

int bufMax, bufMin;

double srednee;

Console.Write("Введите кол-во судей: ");

int N = Int32.Parse(Console.ReadLine());

int[] Ocenki = new int[N];

Random rand = new Random();

srednee = 0;

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

{

Ocenki[i] = rand.Next(0, 6);

Console.Write("{0} ", Ocenki[i]);

srednee += Ocenki[i];

}

int maxValue = Ocenki.Max<int>();

int minValue = Ocenki.Min<int>();

Console.WriteLine();

bufMax = 0;

bufMin = 0;

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

{

if (maxValue == Ocenki[i])

bufMax += 1;

if (minValue == Ocenki[i])

bufMin += 1;

}

if (bufMax > 1)

{

srednee = srednee - maxValue \* bufMax;

N = N - bufMax;

}

if (bufMin > 1)

{

srednee = srednee - minValue \* bufMin;

N = N - bufMin;

}

srednee = srednee / N;

Console.WriteLine("Зачетная оценка: {0}", srednee);

Console.WriteLine();

string month, day, zodiak = "";

Console.WriteLine("Задание 6");

Console.Write("Введите месяц: ");

month = Console.ReadLine();

Console.Write("Введите день: ");

day = Console.ReadLine();

Console.WriteLine();

if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 3) | (Convert.ToInt32(day) < 21 && Convert.ToInt32(month) == 4))

{

zodiak = "Овен";

}

else if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 4) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 5))

{

zodiak = "Телец";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 5) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 6))

{

zodiak = "Близнецы";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 6) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 7))

{

zodiak = "Рак";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 7) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 8))

{

zodiak = "Лев";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 8) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 9))

{

zodiak = "Дева";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 9) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 10))

{

zodiak = "Весы";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 10) | (Convert.ToInt32(day) < 23 && Convert.ToInt32(month) == 11))

{

zodiak = "Скорпион";

}

else if ((Convert.ToInt32(day) > 22 && Convert.ToInt32(month) == 11) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 12))

{

zodiak = "Стрелец";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 12) | (Convert.ToInt32(day) < 21 && Convert.ToInt32(month) == 1))

{

zodiak = "Козерог";

}

else if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 1) | (Convert.ToInt32(day) < 20 && Convert.ToInt32(month) == 2))

{

zodiak = "Водолей";

}

else

{

zodiak = "Рыба";

}

Console.WriteLine("Ваш знак задиака: {0}", zodiak);

Console.WriteLine();

Console.WriteLine("Задание 7");

int a1, a2, b, c1, c2;

Console.Write("Введите a2: ");

a2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите a1: ");

a1 = Int32.Parse(Console.ReadLine());

Console.Write("Введите b: ");

b = Int32.Parse(Console.ReadLine());

if (a1 + b >= 10)

{

c1 = a1 + b - 10;

c2 = a2 + 1;

}

else

{

c1 = a1 + b;

c2 = a2;

}

Console.WriteLine("Цифры полученного числа: {0}, {1}", c2, c1);

Console.WriteLine();

Console.WriteLine("Задание 8");

Console.Write("Задайте диапазон от A: ");

int A = Int32.Parse(Console.ReadLine());

Console.Write(" до B: ");

int B = Int32.Parse(Console.ReadLine());

int temp = A;

for (int i = A; i < B; i++)

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

Console.WriteLine();

temp = A;

while (temp < B)

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

Console.WriteLine();

temp = A;

do

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

while (temp < B);

Console.WriteLine();

Console.WriteLine("Задание 9");

for (int i = 1; i < 7; i++)

{

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

{

Console.Write("{0} ", j);

}

Console.WriteLine();

}

Console.WriteLine();

Console.WriteLine("Задание 10");

Console.Write("Введите n: ");

int n10 = Int32.Parse(Console.ReadLine());

HashSet<int> primes = new HashSet<int>(Enumerable.Range(2, n10 - 1));

for (int i10 = 2; i10 < n10; i10++)

{

if (primes.Contains(i10))

{

primes.RemoveWhere(x => x > i10 && x % i10 == 0);

}

}

primes.Remove(2);

Console.Write("1, ");

Console.WriteLine(String.Join(", ", primes));

Console.WriteLine();

Console.WriteLine("Задание 11");

string nat, x11, y11;

Console.Write("Введите натуральное число: ");

nat = Console.ReadLine();

Console.Write("Введите x: ");

x11 = Console.ReadLine();

Console.Write("Введите y: ");

y11 = Console.ReadLine();

int count;

count = (nat.Length - nat.Replace(x11, "").Length) / x11.Length;

count = count + (nat.Length - nat.Replace(y11, "").Length) / y11.Length;

Console.WriteLine("Символы встречаются: {0}", count);

Console.WriteLine();

Console.WriteLine("Задание 12");

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

int students = Int32.Parse(Console.ReadLine());

int[] Students = new int[students];

Random rand12 = new Random();

int o5 = 0, o4 = 0, o3 = 0, o2 = 0;

for (int i12 = 0; i12 < students; i12++)

{

Students[i12] = rand12.Next(0, 6);

Console.Write("{0} ", Students[i12]);

if (Students[i12] == 5)

{

o5++;

}

if (Students[i12] == 4)

{

o4++;

}

if (Students[i12] == 3)

{

o3++;

}

if (Students[i12] == 2)

{

o2++;

}

}

Console.WriteLine();

Console.WriteLine("Количество 5 = {0}, 4 = {1}, 3 = {2}, 2 = {3}", o5, o4, o3, o2);

Console.ReadKey();

}

}

}

Тестирование:

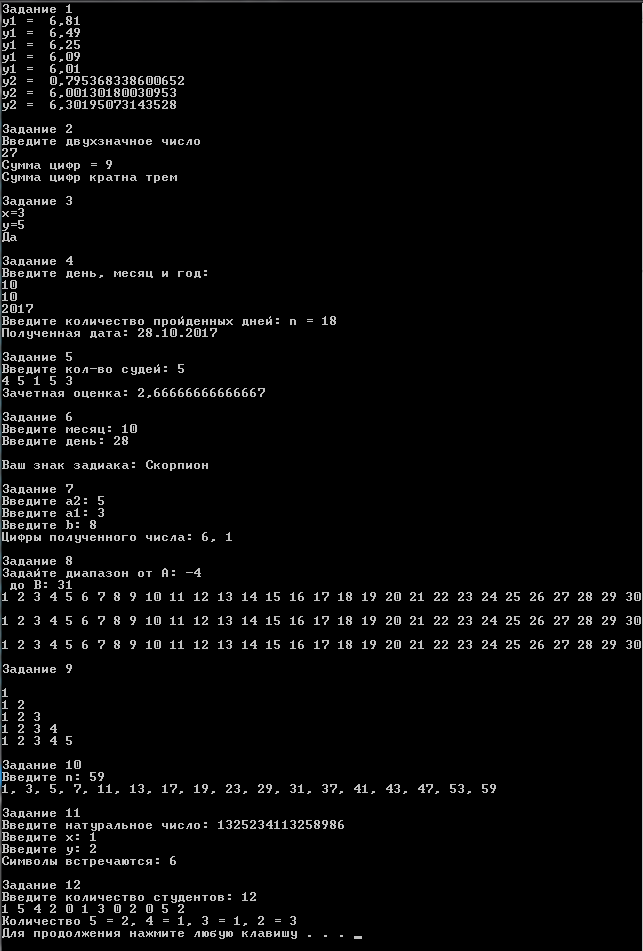


Рисунок 1 – Выполнение программы от первого до восьмого задания включительно

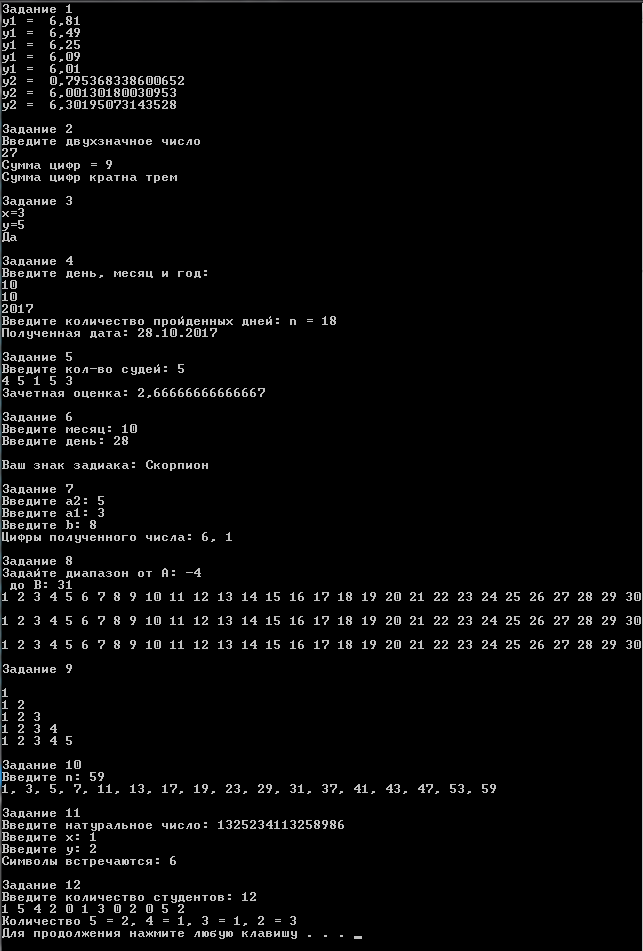


Рисунок 2 – Выполнение программы от девятого до двенадцатого задания включительно

Контрольные вопросы:

1. Какие существуют операторы передачи управления?

Switch, if, else if.

1. Что необходимо для выполнения ветви условного оператора более одного оператора?

Фигурные скобки.

1. Что такое «метка»?

Оператор безусловного перхода.

1. Когда следует использовать оператор безусловного перехода?

Когда обычные условные операторы непременимы.

1. Какова структура полного условного оператора?

If(<условие>)

{<Блок кода>}

Else if(<условие>)

{<блок кода>}

Else

{<блок кода>}

1. Какие действия выполняет оператор перехода?

Исполнение кода переносится у указанной метке.

1. Как работает switch?

Оператор switch принимает значение какой-либо переменной и в соответствии с ее значением исполняется соответствующий блок кода.

1. Что общего у условного оператора от оператора switch?

Они оба сверяют значение переменных.

1. В чем отличие условного оператора от switch?

Условный оператор обрабатывает только один вариант, а switch обрабатывает множество вариантов.

1. Может ли список констант в операторе case состоять из одной константы?

Да, может.

1. Как будет выполняться оператор switch, если ни одна из констант не равна текущему значению выражения case?

Будет исполняться блок default.

1. Когда следует использовать оператор перехода?

Когда никакие другие операторы не позволяют решить задачу.

1. В каких случаях используетя оператор break?

Когда необходимо выйти из текущего блока кода. К примеру в циклах.

1. Когда целесообразно использовать оператор цикла for (while, do while)?

При работе с массивами и многократном повторении одного и тогоже блока кода.

1. Как организовать принудительный выход из цикла?

С помощью оператора break.

1. Что является областью действия переменных, объявленных в части инициализации цикла?

Блок кода соответстующий циклу.

1. Верно ли, что в теле цикла с предусловием должен находиться один оператор?

Нет, не верно.

1. Что нужно сделать, чтобы циклически выполнялись несколько операторов?

Использовать цикл любого типа, в блоке кода данного цикла вписать необходимые операторы.

Лабораторная работа №4

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка методов, использование в программах встроенных функции

Цели:

Образовательная:

сформировать умения разработки методов и использования в программах встроенных функции.

Развивающая:

1. развивать у обучающихся исследовательскую культуру (развивать умения использовать научные методы познаний (наблюдение, гипотеза, эксперимент);
2. развивать способности четко формулировать свои мысли.

Воспитательная:

1. воспитывать в обучающихся средствами урока уверенность в своих силах;
2. воспитывать умение слушать других, культуру речи, общения.

Задания:

1. Написать метод, который вычисляет значения x=sin2(a) и y=cos2(a). Напечатать таблицу значений от –π до π с шагом π/4;
2. Используя функцию NOD2 из задания Proc10, описать процедуру Frac(a,b,p,q), преобразующую дробь a/b к несократимому виду p/q (все параметры процедуры — целого типа). Знак результирующей дроби p/q приписывается числителю (т.е. q > 0). С помощью этой процедуры найти несократимые дроби, равные a/b + c/d, a/b + e/f, a/b + g/h (числа a, b, c, d, e, f, g, h даны).  
   NOD2 из задания Proc 10 - Описать нерекурсивную функцию NOD2(A,B) целого типа, находящую наибольший общий делитель (НОД) двух натуральных чисел A и B, используя алгоритм Евклида: НОД(A,B) = НОД(B mod A,A), если A <> 0; НОД(0,B) = B.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp10

{

class Program

{

static void Method1(double a)

{

double x = Math.Pow(Math.Sin(a), 2);

double y = Math.Pow(Math.Cos(a), 2);

Console.WriteLine("Значение x = {0}, y = {1}", x, y);

double pi = 3.14, piStep = -pi;

while (piStep <= pi)

{

Console.Write("{0} ", piStep);

for (int i = 0; i < 3; i++)

{

piStep = piStep + pi / 4;

if (piStep <= pi + pi / 4)

Console.Write("{0} ", piStep);

}

Console.WriteLine();

}

}

static int NOD2(int A, int B)

{

int NOD;

while (A != B)

{

if (A > B)

A = A - B;

else

B = B - A;

}

NOD = A;

return NOD;

}

static void Frac(int a, int b)

{

int p, q;

p = Math.Abs(a) / NOD2(Math.Abs(a), Math.Abs(b));

q = Math.Abs(b) / NOD2(Math.Abs(a), Math.Abs(b));

if (a \* b < 0)

p = -p;

Console.WriteLine("{0}/{1}", p, q);

}

static void Main(string[] args)

{

Console.WriteLine("Задание 1.");

Console.Write("Введите a: ");

double a1 = double.Parse(Console.ReadLine());

Method1(a1);

Console.WriteLine("Задание 2.");

Console.Write("Введите a: ");

int a2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите b: ");

int b2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите c: ");

int c2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите d: ");

int d2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите e: ");

int e2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите f: ");

int f2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите g: ");

int g2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите h: ");

int h2 = Int32.Parse(Console.ReadLine());

Console.WriteLine("Результат: ");

Console.Write("{0}/{1} + {2}/{3} = ", a2, b2, c2, d2);

Frac(a2 \* d2 + b2 \* c2, b2 \* d2);

Console.Write("{0}/{1} + {2}/{3} = ", a2, b2, e2, f2);

Frac(a2 \* f2 + b2 \* e2, b2 \* f2);

Console.Write("{0}/{1} + {2}/{3} = ", a2, b2, g2, h2);

Frac(a2 \* h2 + b2 \* g2, b2 \* h2);

}

}

}

Тестирование:

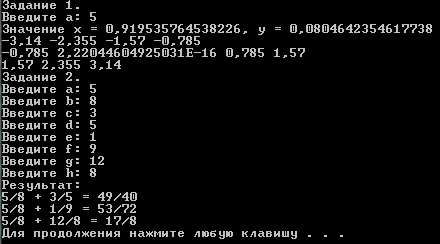


Рисунок 1 – Выполнение программы

Контрольные вопросы:

1. Существует ли подпрограммы без параметров?

Существуют, int f() { return ..}, например.

1. Чем отличается описание процедуры от описания функции?

Описание процедуры отличается от описания функции тем, что в функции обязательно выражение return, то есть обязательно возвратить какое-либо значение.

1. Какое количество значений возвращает функция?

Функция может возвращать только одно значение, если массивы и кортежи считать одним значением.

1. Какие переменные называются локальными?

Локальные переменные – это переменные участвующие в не основном блоке кода метода. Переменная цикла for, например.

Лабораторная работа №5

Выполнил: Олишкевич Игорь Русланович.

Тема: Разработка программ с использованием массивов.

Цель:

Образовательная:

1. сформировать практические навыки работы с одномерными массивами, особенностями их ввода и вывода и обработке данных в них в среде в Visual Studio,
2. сформировать практические навыки работы с многомерными массивами, особенностями их ввода и вывода и обработке данных в них.

Развивающая:

1. научить использовать основные операторы, типы данных и конструкции,
2. создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

1. создать условия для воспитания в обучающихся средствами урока уверенности в своих силах;
2. создать условия для воспитания сознательного и серьёзного отношения обучающихся к учебной дисциплине, убеждая их в том, что полученные знания пригодятся им в будущей деятельности.

Здания:

1. вывести на экран дисплея положительные элементы массива и найти их количество;
2. в одномерном массиве, состоящем из n вещественных элементов, вычислить минимальный элемент массива;
3. в массиве содержатся результаты измерений температуры воздуха, которые проводились ежедневно в течение декабря месяца. Определите день, когда температура была ближе всего к средней температуре в декабре;
4. напишите программу для решения задачи. Пусть дана последовательность из 100 различных целых чисел. Найдите среднее арифметическое чисел этой последовательности, расположенных между максимальным и минимальным числами (в сумму включить и оба этих числа);
5. дан одномерный массив из действительных чисел. Определить сумму шести первых элементов массива;
6. решите задачу. В массиве хранится информация о максимальной скорости каждой из 40 марок легковых автомобилей. Определить порядковый номер самого быстрого автомобиля. Если таких автомобилей несколько, то должен быть найден номер последнего из них;
7. найти наибольший и наименьший элементы матрицы и поменять их местами;
8. дана целочисленная прямоугольная матрица. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент;
9. решите задачу. Фирма имеет 10 магазинов. Информация о доходе каждого магазина за каждый месяц года хранится в двумерном массиве (первого магазина – в первой строке, второго во второй и т.д.). Составить программу для расчета среднемесячного дохода любого магазина;
10. нахождение максимума и минимума. Дан двумерный массив. В каждой его строке найти координаты минимального элемента. Если элементов с минимальным значением в строке несколько, то должны быть найдены координаты самого правого из них;
11. решить задачу на проверку условий после выполнения расчетов. Информация о количестве жильцов в каждой из четырех квартир каждого этажа 12-этажного дома хранится в двумерном массиве (в первой строке – информация о квартирах первого этажа, во второй – второго и т.д.). На каком этаже проживает больше людей: на третьем или на пятом;
12. решить задачу по изменению исходного массива. Дан двумерный массив целых чисел. Все элементы массива, оканчивающиеся на 2, умножить на последний элемент соответствующего столбца;
13. работа с несколькими массивами. Дан двумерный массив размером n\*n. Сформировать одномерный массив из элементов заданного массива, расположенных под побочной диагональю.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp11

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Задание 1.");

int[] mas1 = new int[10];

Random rand1 = new Random();

int count1 = 0;

Console.WriteLine("Весь массив: ");

for (int i1 = 0; i1 < mas1.Length; i1++)

{

mas1[i1] = rand1.Next(-10, 10);

Console.Write("{0} ", mas1[i1]);

}

Console.WriteLine();

Console.WriteLine("Искомый массив: ");

for (int i1 = 0; i1 < mas1.Length; i1++)

{

mas1[i1] = rand1.Next(-10, 10);

if (mas1[i1] > 0)

{

Console.Write("{0} ", mas1[i1]);

count1 += 1;

}

}

Console.WriteLine();

Console.WriteLine("Количество положительных = {0}", count1);

Console.WriteLine();

Console.WriteLine("Задание 2.");

Console.Write("Введите количество элементов массива: ");

int n2 = Int32.Parse(Console.ReadLine());

int[] mas2 = new int[n2];

Random rand2 = new Random();

for (int i2 = 0; i2 < n2; i2++)

{

mas2[i2] = rand2.Next(0, 100);

Console.Write("{0} ", mas2[i2]);

}

Console.WriteLine();

Console.WriteLine("Min = {0}", mas2.Min());

Console.WriteLine();

Console.WriteLine("Задание 3.");

int[] mas3 = new int[31];

Random rand3 = new Random();

int sum3 = 0, min3, k3;

double srednee3 = 0;

for (int i3 = 0; i3 < mas3.Length; i3++)

{

mas3[i3] = rand3.Next(-30, 10);

Console.Write("День {0}: {1} ", i3 + 1, mas3[i3]);

sum3 += mas3[i3];

}

srednee3 = sum3 / 31;

Console.WriteLine();

Console.WriteLine("Средняя температура = {0}", srednee3);

min3 = Math.Abs(mas3[0] - Convert.ToInt32(srednee3));

k3 = 1;

for (int i03 = 1; i03 < mas3.Length; i03++)

if (Math.Abs(mas3[i03] - srednee3) < min3)

{

min3 = Math.Abs(mas3[i03] - Convert.ToInt32(srednee3));

k3 = i03;

}

Console.WriteLine("Наиболее приближенная температура - {0}, номер дня - {1}, разность со средней - {2}.", mas3[k3], k3, min3);

Console.WriteLine();

Console.WriteLine("Задание 4.");

int[] mas4 = new int[100];

Random rand4 = new Random();

int min4, max4, imin4, imax4, index4 = 0;

double srednee4 = 0, sum4 = 0;

Console.WriteLine("Изначальный массив:");

for (int i4 = 0; i4 < mas4.Length; i4++)

{

mas4[i4] = rand4.Next(0, 100);

Console.Write("{0} ", mas4[i4]);

}

Console.WriteLine();

min4 = mas4.Min();

imin4 = Array.IndexOf(mas4, min4);

max4 = mas4.Max();

imax4 = Array.IndexOf(mas4, max4);

Console.WriteLine("Полученный массив:");

if (imin4 > imax4)

{

mas4.Reverse();

min4 = mas4.Min();

imin4 = Array.IndexOf(mas4, min4);

max4 = mas4.Max();

imax4 = Array.IndexOf(mas4, max4);

}

for (int i4 = imin4; i4 < imax4 + 1; i4++)

{

index4 += 1;

sum4 += mas4[i4];

Console.Write("{0} ", mas4[i4]);

}

srednee4 = sum4 / index4;

Console.WriteLine();

Console.WriteLine("Среднее значение = {0}", srednee4);

Console.WriteLine();

Console.WriteLine("Задание 5.");

int[] mas5 = new int[10];

Random rand5 = new Random();

int sum5 = 0;

for (int i5 = 0; i5 < mas5.Length; i5++)

{

mas5[i5] = rand5.Next(-10, 10);

Console.Write("{0} ", mas5[i5]);

if (i5 < 6)

sum5 += mas5[i5];

}

Console.WriteLine();

Console.WriteLine("Сумма первых шести элементов = {0}", sum5);

Console.WriteLine();

Console.WriteLine("Задание 6.");

int[] mas6 = new int[40];

Random rand6 = new Random();

int max6 = 0, index6 = 0;

for (int i6 = 0; i6 < mas6.Length; i6++)

{

mas6[i6] = rand6.Next(150, 155);

Console.Write("Номер {0} - {1}. ", i6 + 1, mas6[i6]);

if (max6 <= mas6[i6])

{

max6 = mas6[i6];

index6 = i6;

}

}

Console.WriteLine();

Console.WriteLine("Последний номер самого скоростного автомобиля = {0}", index6 + 1);

Console.WriteLine();

Console.WriteLine("Задание 7.");

int[,] mas7 = new int[3, 3];

Random rand7 = new Random();

int max7, min7, imax7 = 0, jmax7 = 0, imin7 = 0, jmin7 = 0, buf7;

Console.WriteLine("Изначальная матрица:");

for (int i7 = 0; i7 < 3; i7++)

{

for (int j7 = 0; j7 < 3; j7++)

{

mas7[i7, j7] = rand7.Next(0, 10);

Console.Write("{0} ", mas7[i7, j7]);

}

Console.WriteLine();

}

max7 = mas7[1, 1];

min7 = mas7[1, 1];

for (int i7 = 0; i7 < 3; i7++)

for (int j7 = 0; j7 < 3; j7++)

{

if (min7 > mas7[i7, j7])

{

min7 = mas7[i7, j7];

imin7 = i7;

jmin7 = j7;

}

if (max7 < mas7[i7, j7])

{

max7 = mas7[i7, j7];

imax7 = i7;

jmax7 = j7;

}

}

buf7 = mas7[imin7, jmin7];

mas7[imin7, jmin7] = mas7[imax7, jmax7];

mas7[imax7, jmax7] = buf7;

Console.WriteLine("Полученная матрица:");

for (int i7 = 0; i7 < 3; i7++)

{

for (int j7 = 0; j7 < 3; j7++)

{

Console.Write("{0} ", mas7[i7, j7]);

}

Console.WriteLine();

}

Console.WriteLine();

Console.WriteLine("Задание 8.");

int[,] mas8 = new int[5, 4];

int[] imas8 = new int[5];

Random rand8 = new Random();

int ii = 0, sum8 = 0;

Console.WriteLine("Матрица:");

for (int i8 = 0; i8 < 5; i8++)

{

for (int j8 = 0; j8 < 4; j8++)

{

mas8[i8, j8] = rand8.Next(-1, 6);

Console.Write("{0} ", mas8[i8, j8]);

}

Console.WriteLine();

}

for (int i8 = 0; i8 < 5; i8++)

{

for (int j8 = 0; j8 < 4; j8++)

if (mas8[i8, j8] < 0)

{

imas8[ii] = i8;

ii += 1;

break;

}

}

for (int i8 = 0; i8 < ii; i8

{

sum8 = 0;

for (int j8 = 0; j8 < 4; j8++)

sum8 += mas8[imas8[i8], j8];

Console.WriteLine("Сумма элементов в строке {0} = {1}", imas8[i8], sum8);

}

Console.WriteLine();

Console.WriteLine("Задание 9.");

int[,] mas9 = new int[10, 12];

Random rand9 = new Random();

double srednee = 0, sum9 = 0;

Console.WriteLine("Матрица:");

for (int i9 = 0; i9 < 10; i9++)

{

for (int j9 = 0; j9 < 12; j9++)

{

mas9[i9, j9] = rand9.Next(0, 5);

Console.Write("{0} ", mas9[i9, j9]);

}

Console.WriteLine();

}

Console.Write("Введите номер магазина для подсчета его среднемесячного дохода: ");

int n10 = Int32.Parse(Console.ReadLine());

for (int j9 = 0; j9 < 12; j9++)

sum9 += mas9[n10 - 1, j9];

srednee = sum9 / 12;

Console.Write("Среднемесячный доход = {0}", srednee);

Console.WriteLine();

Console.WriteLine("Задание 10.");

int[,] mas10 = new int[5, 5];

Random rand10 = new Random();

int min10 = 0, imin10 = 0, jmin10 = 0;

Console.WriteLine("Матрица:");

for (int i10 = 0; i10 < 5; i10++)

{

for (int j10 = 0; j10 < 5; j10++)

{

mas10[i10, j10] = rand10.Next(0, 5);

Console.Write("{0} ", mas10[i10, j10]);

}

Console.WriteLine();

}

for (int i10 = 0; i10 < 5; i10++)

{

min10 = mas10[i10, 0];

imin10 = i10;

jmin10 = 0;

for (int j10 = 0; j10 < 5; j10++)

if (min10 >= mas10[i10, j10])

{

min10 = mas10[i10, j10];

imin10 = i10;

jmin10 = j10;

}

Console.WriteLine("Минимум в строке {0} = {1}. Координата y = {2}.", i10 + 1, min10, jmin10 + 1);

}

Console.WriteLine();

Console.WriteLine("Задание 11.");

int[,] mas11 = new int[12, 4];

Random rand11 = new Random();

int sum11\_3 = 0, sum11\_5 = 0;

Console.WriteLine("Матрица:");

for (int i11 = 0; i11 < 12; i11++)

{

for (int j11 = 0; j11 < 4; j11++)

{

mas11[i11, j11] = rand11.Next(1, 6);

Console.Write("{0} ", mas11[i11, j11]);

}

Console.WriteLine();

}

for (int j11 = 0; j11 < 4; j11++)

{

sum11\_3 += mas11[2, j11];

sum11\_5 += mas11[4, j11];

}

Console.Write("На 3 этаже проживает {0}, а на 5 - {1}. ", sum11\_3, sum11\_5);

if (sum11\_3 > sum11\_5)

Console.WriteLine("На 3 проживает больше человек.");

else if (sum11\_3 < sum11\_5)

Console.WriteLine("На 5 проживает больше человек.");

else

Console.WriteLine("Проживает одинаковое количество человек.");

Console.WriteLine();

Console.WriteLine("Задание 12.");

int[,] mas12 = new int[5, 5];

Random rand12 = new Random();

Console.WriteLine("Матрица:");

for (int i12 = 0; i12 < 5; i12++)

{

for (int j12 = 0; j12 < 5; j12++)

{

mas12[i12, j12] = rand12.Next(-2, 13);

Console.Write("{0} ", mas12[i12, j12]);

}

Console.WriteLine();

}

for (int i12 = 0; i12 < 5; i12++)

for (int j12 = 0; j12 < 5; j12++)

{

string str12 = Convert.ToString(mas12[i12, j12]).Substring(Convert.ToString(mas12[i12, j12]).Length - 1);

if (str12 == "2")

{

mas12[i12, j12] \*= mas12[4, j12];

}

}

Console.WriteLine("Полученная матрица:");

for (int i12 = 0; i12 < 5; i12++)

{

for (int j12 = 0; j12 < 5; j12++)

{

Console.Write("{0} ", mas12[i12, j12]);

}

Console.WriteLine();

}

Console.WriteLine();

Console.WriteLine("Задание 13.");

Console.Write("Введите размерность массива: ");

int n13 = Int32.Parse(Console.ReadLine());

int[,] mas13 = new int[n13, n13];

Random rand13 = new Random();

int n13\_new = 0;

for (int i13 = 0; i13 < n13; i13++)

n13\_new += i13;

int[] mas13\_new = new int[n13\_new];

int ii13 = 0;

Console.WriteLine("Матрица:");

for (int i13 = 0; i13 < n13; i13++)

{

for (int j13 = 0; j13 < n13; j13++)

{

mas13[i13, j13] = rand13.Next(0, 10);

Console.Write("{0} ", mas13[i13, j13]);

}

Console.WriteLine();

}

for (int i13 = 0; i13 < n13; i13++)

for (int j13 = 0; j13 < n13; j13++)

if (i13 + j13 > n13 - 1)

{

mas13\_new[ii13] = mas13[i13, j13];

ii13 += 1;

}

Console.WriteLine("Новый массив:");

for (int i13 = 0; i13 < n13\_new; i13++)

Console.Write("{0} ", mas13\_new[i13]);

Console.ReadKey();

}

}

}

Тестирование:

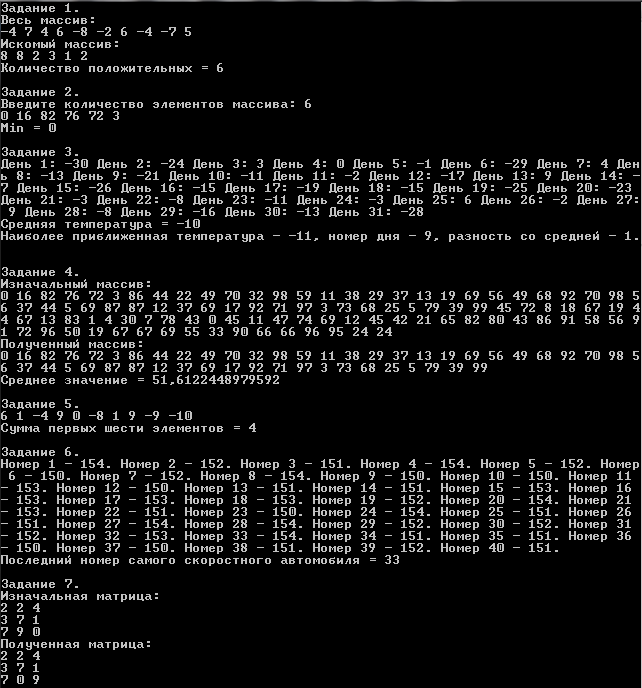


Рисунок 1 – Выполнение программы с первого по четвертое задание включительно



Рисунок 2 – Выполнение программы с пятого по десятое задание включительно

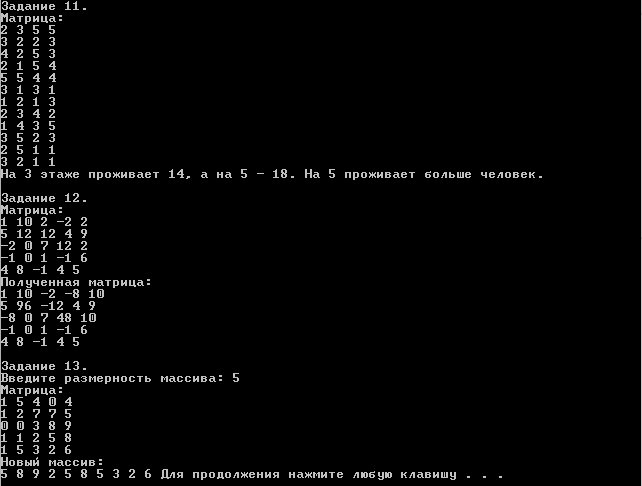


Рисунок 3 – Выполнение программы с одиннадцатого по тринадцатое задание включительно

Контрольные вопросы:

1) Каким образом описывают массивы?

Тип\_данных[,] название\_массива. Запятая или запятые в квадратных скобках обозначает размерность массива (0 запятых – одномерный массив);

2) Укажите особенности ввода и вывода одномерных массивов?

Осуществляется единственный цикл по перебору каждого элемента при помощи идентификации. В момент перехода к элементу можно производить ввод и вывод.

3) В чем состоит особенность использования приемов программирования при обработке одномерных массивов?

Массив начинается с [0] элемента и может иметь различные типы, а также являться динамизированным.

4) Существуют ли ограничения на размерность массива?

Ограничения лежат в пределах возможностей типа массива.

5) Где и как определяется общее число элементов массива?

Общее число элементов объявляется единожды при инициализации, после изменения размерности невозможны.

6) Когда индекс элемента в массиве совпадает с порядковым номером этого элемента?

Можно произвести операции над этим элементом.

7) Что называется базовым типом?

Стандартный набор типов данных.

8) Верно ли, что элементами массива могут быть данные любого типа, включая структурированные?

Нет.

9) Какой массив называется одномерным?

Массив, включающий в себя 1 – мерное пространство. Индексация происходит только по одному значению (например int[] mas = new int[5] – одномерный массив из 5 элементов) .

10) Можно ли с помощью стандартной процедуры write(x) вывести весь массив х целиком?

Нет, необходимо использовать цикл.

11) Каким образом описывают массивы?

type[] arrayName;

12) Как происходит обращение к элементам массива?

По идентификатору элемента.

13) Укажите особенности ввода и вывода массивов.

Элементы массива можно добавить при инициализации, также во время работы программы путем кодинга или ручного ввода. Вывод осуществляется программно при помощи цикла и встроенных модулей вывода.

14) В чем состоит особенность использования приемов программирования при обработке массивов?

Возможность обращаться как напрямую к непосредственно массиву или его элементам.

15) Дайте определение массива.

Структура данных в виде набора компонентов (элементов массива), расположенных в памяти непосредственно друг за другом. При этом доступ к отдельным элементам массива осуществляется с помощью индексации, то есть через ссылку на массив с указанием номера (индекса) нужного элемента. За счёт этого, в отличие от списка, массив является структурой данных, пригодной для осуществления произвольного доступа к её ячейкам.

16) Какие типы данных не допустимы для компонентов массива? Почему?

Структурированные типы данных недопустимы.

17) Когда индекс элемента в массиве совпадает с порядковым номером этого элемента?

Никогда, индекс массива всегда будет отличаться на единицу от порядкового номера.

18) Может ли индекс быть выражением любого порядкового типа?

Да, может.

19) Можно ли в описании массива использовать предварительно определенные константы?

Да, можно.

20) Как можно сымитировать работу с массивом переменной длины?

Задать дополнительное ограничение по индексам.

Лабораторная работа №6

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка программ с использованием символов и строк.

Цели:

Образовательная:

1. сформировать практические навыки работы с символами и строками, особенностями их ввода и вывода и обработке данных в них в среде в Visual Studio,
2. сформировать практические навыки работы с символами и строками, особенностями их ввода и вывода и обработке данных в них.

Развивающая:

1. научить использовать основные операторы, методы и свойства классов char и string,
2. создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

1. создать условия для воспитания в обучающихся средствами урока уверенности в своих силах;
2. создать условия для воспитания сознательного и серьёзного отношения обучающихся к учебной дисциплине, убеждая их в том, что полученные знания пригодятся им в будущей деятельности.

Задания:

1. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Поменять местами i- и j-ю буквы. Для ввода i и j на форме добавить свои поля вода;
2. Дана строка. Подсчитать, сколько различных символов встречается в ней. Вывести их на экран.
3. В записке слова зашифрованы – каждое из них записано наоборот. Расшифровать сообщение.
4. Составить программу, которая будет вводить строку в переменную string. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.
5. Выберите 10 произвольных букв русского алфавита. Введите произвольное слово. С помощью ключа длиной от 3 до 8 символов произведите шифровку слова в числовую комбинацию.

В Е Ж М Н О П Р С Т КЛЮЧ - 1234

0 1 2 3 4 5 6 7 8 9

МНОЖЕСТВО

3 4 5 2 1 8 9 0 5

1 2 3 4 1 2 3 4 1

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

4 6 8 6 3 1 2 4 6 - РЕЗУЛЬТАТ

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Задание 1.");

string str1new = "", str1 = "";

back1begin: Console.WriteLine("Введите текст: ");

str1 = Console.ReadLine();

if (str1.Length < 2)

goto back1begin;

back1begintext: Console.Write("Введите i-позицию: ");

int i1 = Int32.Parse(Console.ReadLine());

Console.Write("Введите j-позицию, большую чем i: ");

int j1 = Int32.Parse(Console.ReadLine());

if (i1 > j1 || i1 > str1.Length || j1 > str1.Length)

{

goto back1begintext;

}

for (int i = 0; i < i1; i++)

{

str1new += str1.Substring(i, 1);

}

str1new += str1.Substring(j1, 1);

for (int i = i1 + 1; i < j1; i++)

{

str1new += str1.Substring(i, 1);

}

str1new += str1.Substring(i1, 1);

for (int i = j1 + 1; i < str1.Length; i++)

{

str1new += str1.Substring(i, 1);

}

Console.WriteLine(str1new);

Console.WriteLine();

Console.WriteLine("Задание 2-3.");

Console.Write("Введите слово: ");

string text2 = Console.ReadLine();

Dictionary<char, int> dictionarys = text2.GroupBy(x => x)

.ToDictionary(x => x.Key, x => x.Count());

foreach (KeyValuePair<char, int> keyValuePair in dictionarys)

{

Console.WriteLine("{0} : {1}", keyValuePair.Key, keyValuePair.Value);

}

char[] arr2 = text2.ToCharArray();

Array.Reverse(arr2);

Console.WriteLine(arr2);

Console.WriteLine();

Console.WriteLine("Задание 4.");

Console.Write("Введите текст: ");

string[] slova = Console.ReadLine().Split(' ');

List<string> list = new List<string>();

Console.WriteLine("Повторяющиеся слова:");

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

{

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

{

if (i != j)

{

if (slova[i] == slova[j] && !list.Contains(slova[i]))

list.Add(slova[i]);

}

}

}

foreach (string slovo in list.ToArray())

Console.WriteLine(slovo);

Console.WriteLine();

Console.WriteLine("Задание 5.");

string a5 = "АКОЛИНСБУЗ", key5 = "", result5str = "";

int ii5 = 0;

Console.Write("Введите слово, содержащее буквы А К О Л И Н С Б У З: ");

string b5 = Console.ReadLine();

Console.Write("Введите длину ключа от 3 до 8: ");

back5begin: int key5in = Int32.Parse(Console.ReadLine());

if (key5in < 3 || key5in > 8)

goto back5begin;

for (int i = 0; i < b5.Length; i++) //Состовляем ключ в строку по длине введенного слова

{

ii5++;

if (ii5 <= key5in)

{

key5 += ii5;

}

else

{

ii5 = 1;

key5 += ii5;

}

}

Console.WriteLine("Полученный ключ: {0}.", key5);

int n5 = b5.Length + 1;

int [] mas5 = new int [n5];

int[] mas5new = new int[n5];

Console.Write("Полученные коды букв: ");

for (int i = 0; i < b5.Length; i++)

{

mas5[i] = (a5.IndexOf(b5[i])); //Получаем индекс буквы

mas5new[i] = mas5[i] + Int32.Parse(key5.Substring(i, 1)); //Получаем шифрованную букву

Console.Write("{0} ", mas5new[i]);

}

Console.WriteLine();

for (int i = b5.Length - 1; i > -1; i--) //Посимвольно складываем шифрованные буквы в строку задом наперед

{

if (mas5new[i] > 9 && i != 0)

{

result5str += Convert.ToString(mas5new[i]).Substring(1, 1);

mas5new[i - 1] += 1;

}

else if (mas5new[i] < 10)

{

result5str += Convert.ToString(mas5new[i]);

}

if (i == 0 && mas5new[i] > 9)

{

result5str += Convert.ToString(mas5new[i]).Substring(1, 1);

result5str += Convert.ToString(mas5new[i]).Substring(0, 1);

}

}

char[] result5strnew = result5str.ToCharArray();

Array.Reverse(result5strnew);

Console.WriteLine(result5strnew);

Console.ReadKey();

}

}

}

Тестирование:

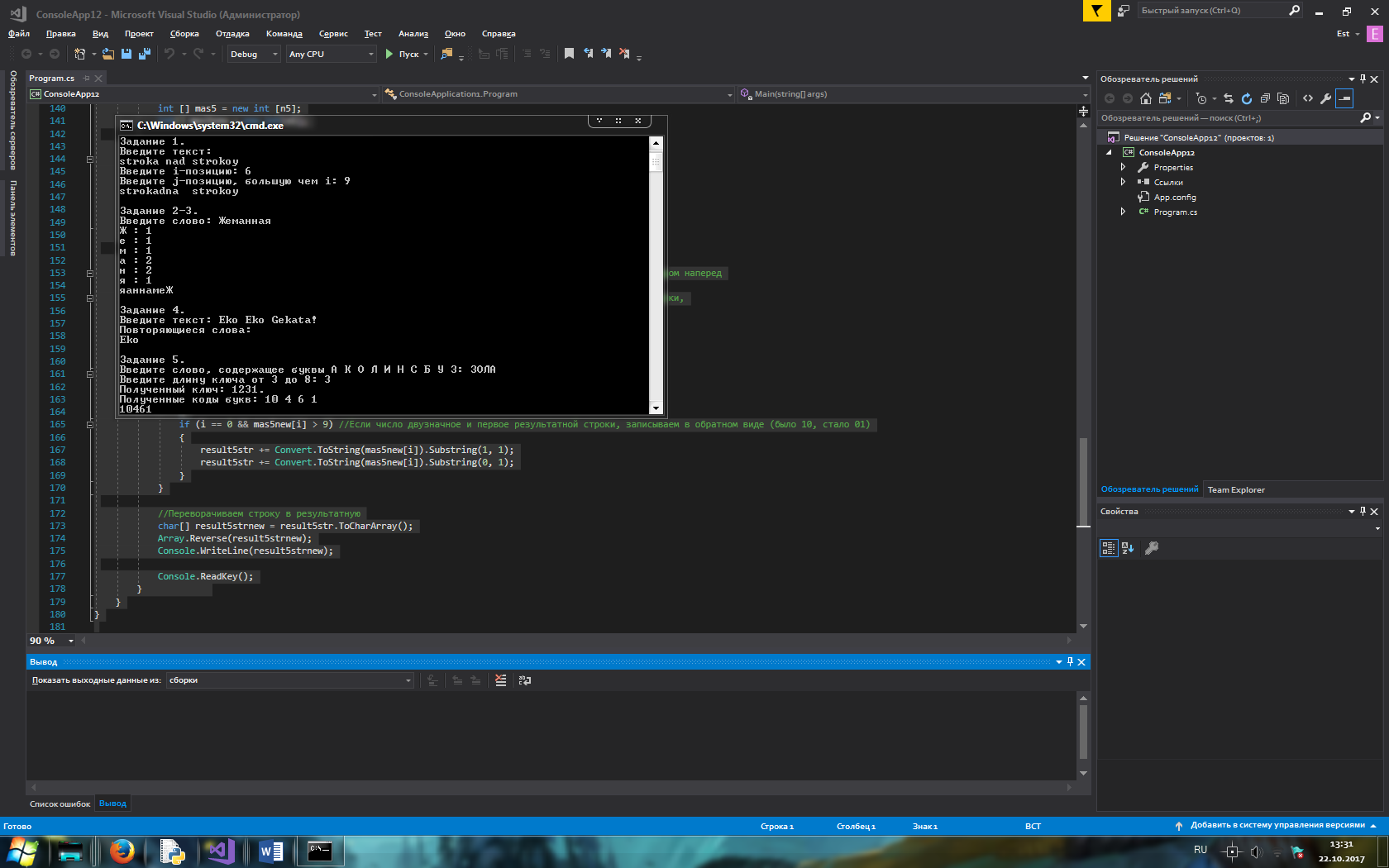


Рисунок 1 – Выполнение программы

Контрольные вопросы:

1. Способы создания строкового типа.
2. string s;
3. string s=''кол около колокола'';
4. string s=@'Привет!
5. Сегодня хорошая погода!!! ''
6. string s=new string (' ', 20);
7. int x = 12344556;
8. string s = x.ToString();
9. char [] a={'a', 'b', 'c', 'd', 'e'};
10. string v=new string (a);
11. char [] a={'a', 'b', 'c', 'd', 'e'};
12. string v=new string (a, 0, 2).

2) Символьный тип char предназначен для хранения символа в кодировке Unicode. Символьный тип относится к встроенным типам данных С# и соответствует стандартному классу Сhar библиотеки .Net из пространства имен System.

Способы инициализации:

1. char ch1 = 'A';
2. обычная ESC-последовательность – код символа Z
3. char ch2 = '\x5A';
4. Unicode-последовательность – код символа X
5. char ch3 = '\u0058';
6. объявление в объектном стиле в динамической памяти
7. char ch = new Char();
8. присваивание по правилам присваивания
9. значения ссылочному типу
10. ch = ch1; значение ‘A’
11. неявное преобразование символьного типа в тип int
12. int code;
13. code = ch; в code – значение 65
14. явное преобразование – в ch1 будет значение ‘B’
15. ch1 = (char)(code +1);

преобразование символьного типа в строку

string s;

s = ch; это некорректно – разные типы

преобразование типов – в строке будет значение BZX

s = ch1.ToString() + ch2.ToString() + ch3.ToString();

3 - 4) Методы и свойства класса string.

1. Clone() Возвращает ссылку на данный экземпляр класса String.
2. System\_CAPS\_pubmethodSystem\_CAPS\_static Compare(String, Int32, String, Int32, Int32) Сравнивает подстроки двух указанных объектов String и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
3. System\_CAPS\_pubmethodSystem\_CAPS\_static Compare(String, Int32, String, Int32, Int32, Boolean) Сравнивает подстроки двух заданных объектов String (с учетом или без учета регистра) и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
4. System\_CAPS\_pubmethodSystem\_CAPS\_static Compare(String, Int32, String, Int32, Int32, Boolean,CultureInfo) Сравнивает подстроки двух заданных объектов String (с учетом или без учета регистра), используя сведения о языке и региональных параметрах, и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
5. System\_CAPS\_pubmethodSystem\_CAPS\_static Compare(String, Int32, String, Int32, Int32, CultureInfo, CompareOptions) Сравнивает подстроки двух заданных объектов String, используя указанные параметры сравнения и сведения о языке и региональных параметрах, которые влияют на сравнение, и возвращает целое число, показывающее связь между двумя подстроками в порядке сортировки.
6. System\_CAPS\_pubmethodSystem\_CAPS\_static Compare(String, Int32, String, Int32, Int32, StringComparison) Сравнивает подстроки двух указанных объектов String с использованием заданных правил и возвращает целое число, которое показывает их относительное положение в порядке сортировки.

Лабораторная работа №7

Выполнил: Олишкевич Игорь Русланович.

Тема: создание и применение регулярных выражений.

Цель: сформировать практические навыки работы с символами и строками, особенностями их ввода и вывода и обработке данных в них в среде в Visual Studio.

Задания:

1. В сообщении могут встречаться номера телефонов, записанные в формате xx-xx-xx, xxx-xxx или xxx-xx-xx. Вывести все номера телефонов, которые содержатся в сообщении;
2. Шаблоны регулярных выражений для групп time, ip и site записаны в упрощенном виде. Преобразуйте их к такому виду, чтобы они соответствовали ограничениям, накладываемым на формат времени, ip-адреса и адреса web-сайта.  
   Алгоритм: ввод данных, проверка на соответствие введенных данных с помощью регулярных выражений, вывод соответствующего результата.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Задание 1.");

Boolean Go = true;

Random rand = new Random();

while (Go)

{

back: string str1 = "";

str1 += rand.Next(1, 10);

int buf1 = rand.Next(2);

if (buf1 == 0)

{

for (int i = 0; i < 6; i++)

str1 += rand.Next(10);

Console.WriteLine("{0:000-00-00}", Convert.ToInt32(str1));

goto back;

}

for (int i = 0; i < 5; i++)

str1 += rand.Next(10);

int buf2 = rand.Next(2);

if (buf2 == 0)

Console.WriteLine("{0:00-00-00}", Convert.ToInt32(str1));

else

Console.WriteLine("{0:000-000}", Convert.ToInt32(str1));

int buf3 = rand.Next(40);

if (buf3 == 33)

Go = false;

}

Console.WriteLine();

Console.WriteLine("Задание 2.");

string text = @"04:55:34 223.345.12.158 www.aaa.ru

04:59:55 213.134.112.56 www.bbb.net

05:05:01 223.34.12.156 www.aaa.by";

Regex theReg = new Regex(@"(?<time>(\d|\:)+)\s" +

@"(?<ip>(\d|\.)+)\s" +

@"(?<site>\S+)");

MatchCollection theMatches = theReg.Matches(text);

foreach (Match theMatch in theMatches)

{

if (theMatch.Length != 0)

{

Console.WriteLine("\ntheMatch: {0}", theMatch.ToString()); //1

Console.WriteLine("time: {0}", theMatch.Groups["time"]); //2

Console.WriteLine("ip: {0}", theMatch.Groups["ip"]); //3

Console.WriteLine("site: {0}", theMatch.Groups["site"]); //4

}

}

Console.WriteLine();

Console.WriteLine("Правильные ограничения:");

Console.WriteLine();

Regex theReg1 = new Regex(@"(?<time1>\d{1,2}:\d{1,2}:\d{1,2})\s" +

@"(?<ip1>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s" +

@"(?<site1>\S{1,3}\.\S\*\.by|\S{1,3}\.\S\*\.net)");

MatchCollection thematches1 = theReg1.Matches(text);

foreach (Match sss in thematches1)

{

Console.WriteLine("\ntheMatch: {0}", sss.ToString()); //1

Console.WriteLine("time: {0}", sss.Groups["time1"]); //2

Console.WriteLine("ip: {0}", sss.Groups["ip1"]); //3

Console.WriteLine("site: {0}", sss.Groups["site1"]); //4

}

Console.ReadKey();

}

}

}

Тестирование:

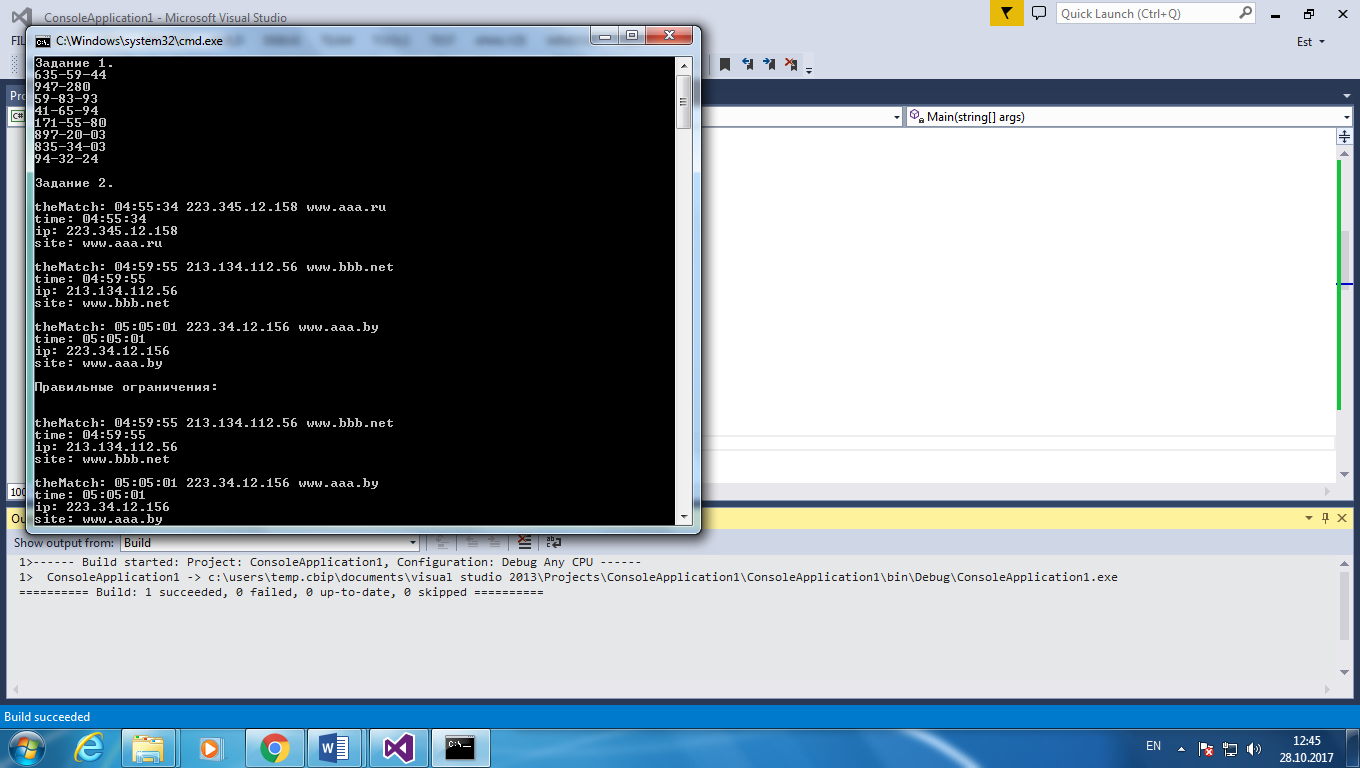


Рисунок 1 – Выполнение программы

Ответы на контрольные вопросы:

1. Используя регулярные выражения, вы указывается правила для множества возможных строк, которые вы хотите проверить; это множество может содержать английские фразы, или адреса электронной почты, или TeX команды, или все что угодно. С помощью РВ вы можете задавать вопросы, такие как «Соответствует ли эта строка шаблону?», или «Совпадает ли шаблон где-нибудь с этой строкой?». Вы можете также использовать регулярные выражения, чтобы изменить строку или разбить ее на части различными способами.
2. Обратные ссылки предоставляют удобный способ идентификации повторяющегося символа или подстроки в строке. Например, если входная строка содержит несколько экземпляров произвольной подстроки, то можно найти первое вхождение с помощью группы записи, а затем использовать обратную ссылку для поиска последующих вхождений подстроки.

Лабораторная работа №8

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка программ с использованием структур и перечислений.

Цель: сформировать понятия структур и перечислений и их реализации в С#.

Упражнение 1: Создание перечисления.

В этом упражнении Вы создадите перечисление для представления различных типов банковских счетов. Затем Вы используете это перечисление для создания двух переменных, которым Вы присвоите значения Checking и Deposit. Далее Вы выведете на экран значения этих переменных, используя функцию System.Console.WriteLine.

Создайте перечисление

Создайте проект BankAccount.sln в папке install folder\Labs\Lab03\Starter\BankAccount.

Переменуйте файл Program.cs на файл Enum.cs, согласитесь с предложением изменить ссылки на новое имя.

Перед описанием класса добавьте перечисление AccountType:

public enum AccountType { Checking, Deposit }

Данное перечисление содержит типы Checking и Deposit.

В методе Main объявите две переменные типа AccountType:

AccountType goldAccount;

AccountType platinumAccount;

Присвойте первой переменной значение Checking, а второй - Deposit: goldAccount = AccountType.Checking; platinumAccount = AccountType.Deposit;

Выведите на консоль значения обеих переменных, два раза используя метод Console.WriteLine:

Console.WriteLine("The Customer Account Type is {0}", goldAccount);

Console.WriteLine("The Customer Account Type is {0}", platinumAccount);

Откомпилируйте и запустите программу.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

class Enum

{

static void Main(string[] args)

{

AccountType goldAccount;

AccountType platinumAccount;

goldAccount = AccountType.Checking;

platinumAccount = AccountType.Deposit;

Console.WriteLine("The Customer Account Type is {0}", goldAccount);

Console.WriteLine("The Customer Account Type is {0}", platinumAccount);

Console.ReadKey();

}

}

}

Скриншоты выполнения:

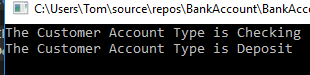


Рисунок 1 – Выполнение программы

Упражнение 2: Создание и использование структуры

В этом упражнении Вы создадите структуру, которую можно использовать для представления банковских счетов. Для хранения номеров счетов (тип данных long), балансов счетов (тип данных decimal) и типов счетов (перечисление, созданное в упражнении 1) будете использовать переменные. Затем создадите переменную типа структуры, заполните ее данными и выведете результаты на консоль.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

public struct BankAccount

{

public long accNo;

public decimal accBal;

public AccountType accType;

}

class Enum

{

static void Main(string[] args)

{

BankAccount goldAccount;

BankAccount platinumAccount;

goldAccount.accType = AccountType.Checking;

goldAccount.accBal = (decimal)3200.00;

goldAccount.accNo = 123;

Console.WriteLine("\*\*\* Account Summary \*\*\*");

Console.WriteLine("Acct Number {0}", goldAccount.accNo);

Console.WriteLine("Acct Type {0}", goldAccount.accType);

Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);

Console.ReadKey();

}

}

}

Скриншоты выполнения:

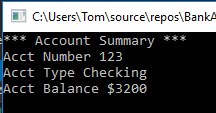


Рисунок 2 – Выполнение программы

Упражнение 3: Добавление возможности ввода/вывода

В этом упражнении Вы измените код, написанный в упражнении 2. Вместо использования счета номер 123, Вы будете запрашивать номер счета у пользователя, а потом использовать его при выводе информации о банковском счете на консоль.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

public struct BankAccount

{

public long accNo;

public decimal accBal;

public AccountType accType;

}

class Enum

{

static void Main(string[] args)

{

BankAccount goldAccount;

BankAccount platinumAccount;

goldAccount.accType = AccountType.Checking;

goldAccount.accBal = (decimal)3200.00;

Console.Write("Enter account number: ");

goldAccount.accNo = long.Parse(Console.ReadLine());

Console.WriteLine("\*\*\* Account Summary \*\*\*");

Console.WriteLine("Acct Number {0}", goldAccount.accNo);

Console.WriteLine("Acct Type {0}", goldAccount.accType);

Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);

Console.ReadKey();

}

}

}

Скриншоты выполнения:

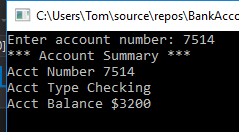


Рисунок 3 – Выполнение программы

Задание: написать программу согласно заданию с использованием структуры.

|  |  |
| --- | --- |
| Наименование, Цена, Дата производства, Срок годности, Количество, Производитель | Вывести все сведения о просроченных товарах. |

Алгоритм: создание массива структур с необходимыми полями данных, проверка на текущую дату от введенной, если структура удовлетворяет условию, то выводятся значения все полей экземпляра.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace ConsoleApp3

{

struct Product

{

public string Name;

public int Price;

public DateTime Manufactured;

public DateTime Expires;

public int Quantity;

public string Company;

public Product(string Name, int Price, DateTime Manufactured,

DateTime Expires, int Quantity, string Company)

{

this.Name = Name;

this.Price = Price;

this.Manufactured = Manufactured;

this.Expires = Expires;

this.Quantity = Quantity;

this.Company = Company;

}

}

class Program

{

static void Main()

{

try

{

Console.Write("Кол-во товаров: ");

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

Product[] Products = new Product[n];

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

{

Console.WriteLine("Заполнение данных {0} товара.", i + 1);

Console.Write("Наименование: ");

string Name = Console.ReadLine();

Console.Write("Цена: ");

int Price = Int32.Parse(Console.ReadLine());

Console.Write("Дата изготовления: ");

DateTime Manufactured = DateTime.Parse(Console.ReadLine());

Console.Write("Срок годности: ");

DateTime Expires = DateTime.Parse(Console.ReadLine());

Console.Write("Количество: ");

int Quantity = Int32.Parse(Console.ReadLine());

Console.Write("Производитель: ");

string Company = Console.ReadLine();

Console.WriteLine();

Products[i] = new Product(Name, Price, Manufactured, Expires, Quantity, Company);

}

Console.WriteLine("Информация о просроченных товарах:");

foreach (Product a in Products)

{

if (a.Expires < DateTime.Today)

{

Console.WriteLine("Наименование: {0}", a.Name);

Console.WriteLine("Цена:{0}", a.Price);

Console.WriteLine("Дата изготовления: {0}", a.Manufactured);

Console.WriteLine("Срок годности: {0}", a.Expires);

Console.WriteLine("Количество: {0}", a.Quantity);

Console.WriteLine("Производитель: {0}", a.Company);

Console.WriteLine();

}

}

}

catch (Exception e)

{

Console.WriteLine(e.ToString());

}

finally

{

Console.ReadKey();

}

}

}

}

Тестирование:

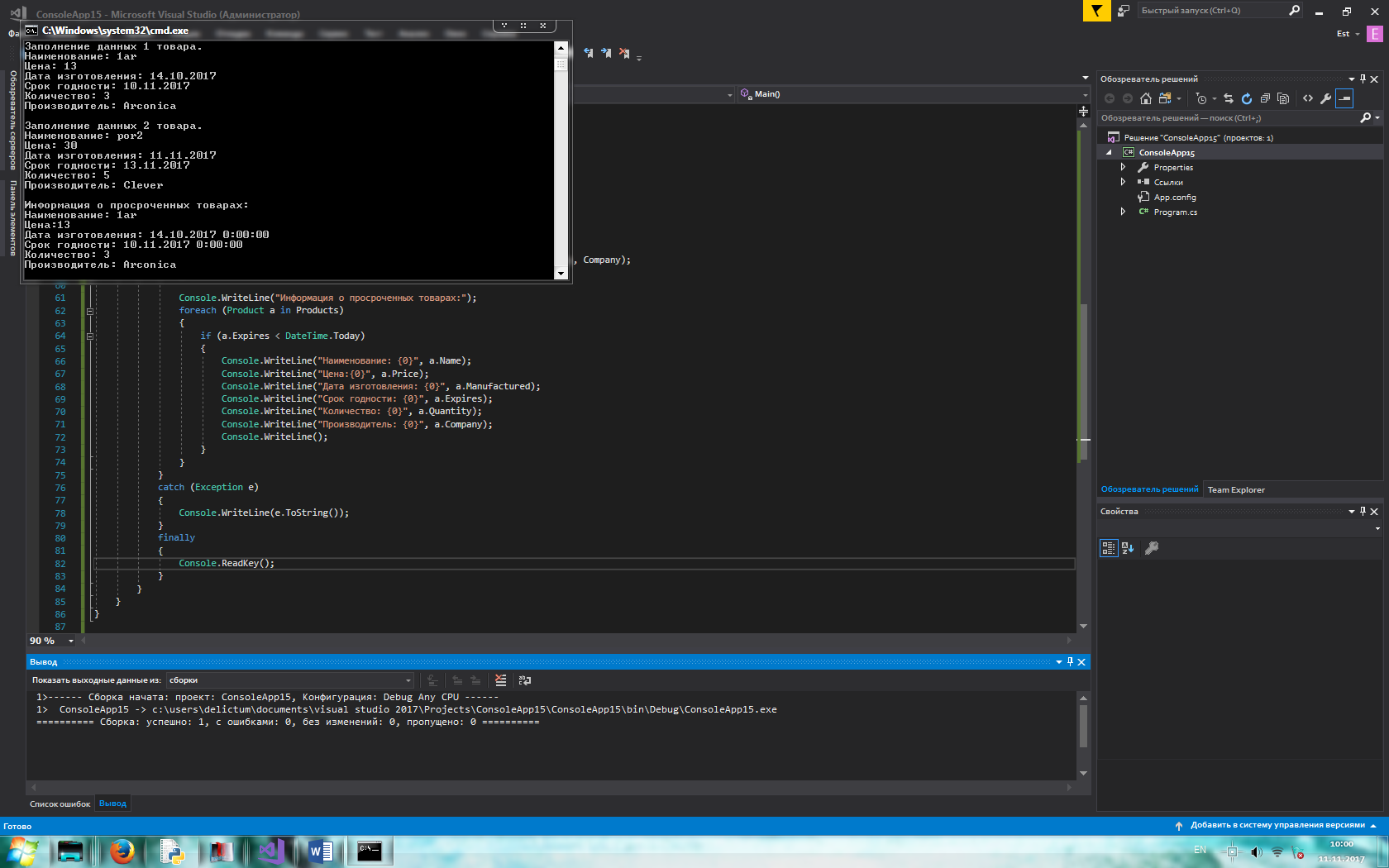


Рисунок 4 – Выполнение программы

Лабораторная работа №9

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка классов и использование их в программах.

Цель: сформировать понятия класса, конструктора класса, деструктора класса и их реализации в С#.

Задания:

1. Заданы координаты четырёх точек A, B, C, D на плоскости. Определить наибольший из периметров треугольников ABC, ABD, ACD.

Листинг программы:  
using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ConsoleApp39

{

class Trungle

{

double[] xABCD, yABCD;

int n;

double AB, BC, CA, BD, DA, AC, CD, ABC, ABD, ACD;

public Trungle(int n)

{

this.xABCD = new double[n];

this.yABCD = new double[n];

this.n = n;

}

public double TrIn()

{

Console.WriteLine("Задайте координаты точек A, B, C и D по x: ");

for (int i = 0; i < 4; i++)

xABCD[i] = double.Parse(Console.ReadLine());

Console.WriteLine("Задайте координаты точек A, B, C и D по y: ");

for (int i = 0; i < 4; i++)

yABCD[i] = double.Parse(Console.ReadLine());

return (0);

}

public double TrOut()

{

Console.Write("Вершины AB, BC, CA: {0}, {1} и {2}. Периметр треугольника ABC: {3}", AB, BC, CA, ABC);

Console.WriteLine();

Console.Write("Вершины AB, BC, CA: {0}, {1} и {2}. Периметр треугольника ABD: {3}", AB, BD, DA, ABD);

Console.WriteLine();

Console.Write("Вершины AB, BC, CA: {0}, {1} и {2}. Периметр треугольника ACD: {3}", AC, CD, DA, ACD);

Console.WriteLine();

return (0);

}

private double Vershina(double x1, double y1, double x2, double y2)

{

return Math.Sqrt((Math.Pow((x1 - x2), 2)) + (Math.Pow((y1 - y2), 2)));

}

private double Perimetr(double AB, double BC, double CA)

{

return AB + BC + CA;

}

public double Strata()

{

AB = Vershina(xABCD[0], yABCD[0], xABCD[1], yABCD[1]);

BC = Vershina(xABCD[1], yABCD[1], xABCD[2], yABCD[2]);

CA = Vershina(xABCD[2], yABCD[2], xABCD[0], yABCD[0]);

BD = Vershina(xABCD[1], yABCD[1], xABCD[3], yABCD[3]);

DA = Vershina(xABCD[3], yABCD[3], xABCD[0], yABCD[0]);

CD = Vershina(xABCD[2], yABCD[2], xABCD[3], yABCD[3]);

ABC = Perimetr(AB, BC, CA);

ABD = Perimetr(AB, BD, DA);

ACD = Perimetr(AC, CD, DA);

return (0);

}

}

public class Program

{

static void Main(string[] args)

{

Trungle tr = new Trungle(5);

tr.TrIn();

tr.Strata();

tr.TrOut();

Console.ReadKey();

}

}

}

Тестирование:

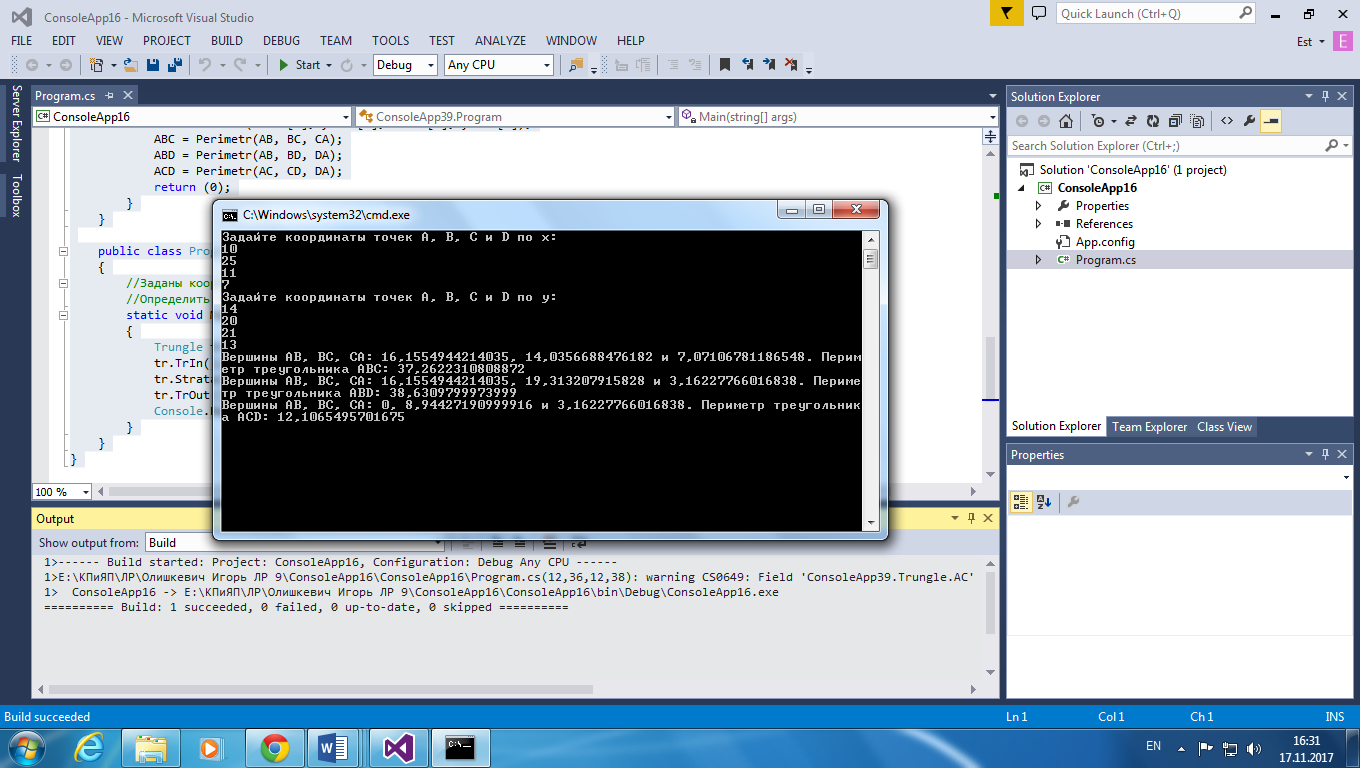


Рисунок 1 – Выполнени программы

1. Реализовать класс Goods (товар). В классе должны быть представлены поля:

* наименование товара (категория);
* дата оформления;
* цена товара;
* количество единиц товара;
* номер накладной, по которой товар поступил на склад.

Реализовать методы:

* изменения цены товара;
* изменения количества товара (увеличение/уменьшение);
* вычисления общей стоимости товара данной категории;
* toString() должен выдавать в виде строки стоимость товара.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Goods

{

string Name;

DateTime DateO;

int Price, Count, NumerN;

public Goods(string Name, DateTime DateO, int Price, int Count, int NumerN)

{

this.Name = Name;

this.DateO = DateO;

this.Price = Price;

this.Count = Count;

this.NumerN = NumerN;

}

public double DisPrice(int n)

{

return Price += n;

}

public double DisCount(int n)

{

return Count += n;

}

public void ResPrice()

{

double n = Price \* Count;

Console.WriteLine("Общая стоимость товара: {0}", n);

}

public void ToString()

{

Console.WriteLine("Стоимость товара: {0}", Price);

}

}

class Program

{

static void Main(string[] args)

{

Goods g = new Goods("Йогурт", DateTime.Today, 89, 73, 124235);

g.ToString();

Console.Write("Введите на какую сумму изменить цену. Для уменьшения ввод отрицательного числа, иначе - положительного: ");

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

g.DisPrice(buf);

g.ToString();

Console.Write("Введите на какое количество изменить товары. Для уменьшения ввод отрицательного числа, иначе - положительного: ");

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

g.DisCount(buf);

g.ResPrice();

Console.ReadKey();

}

}

}

Скриншоты выполнения:

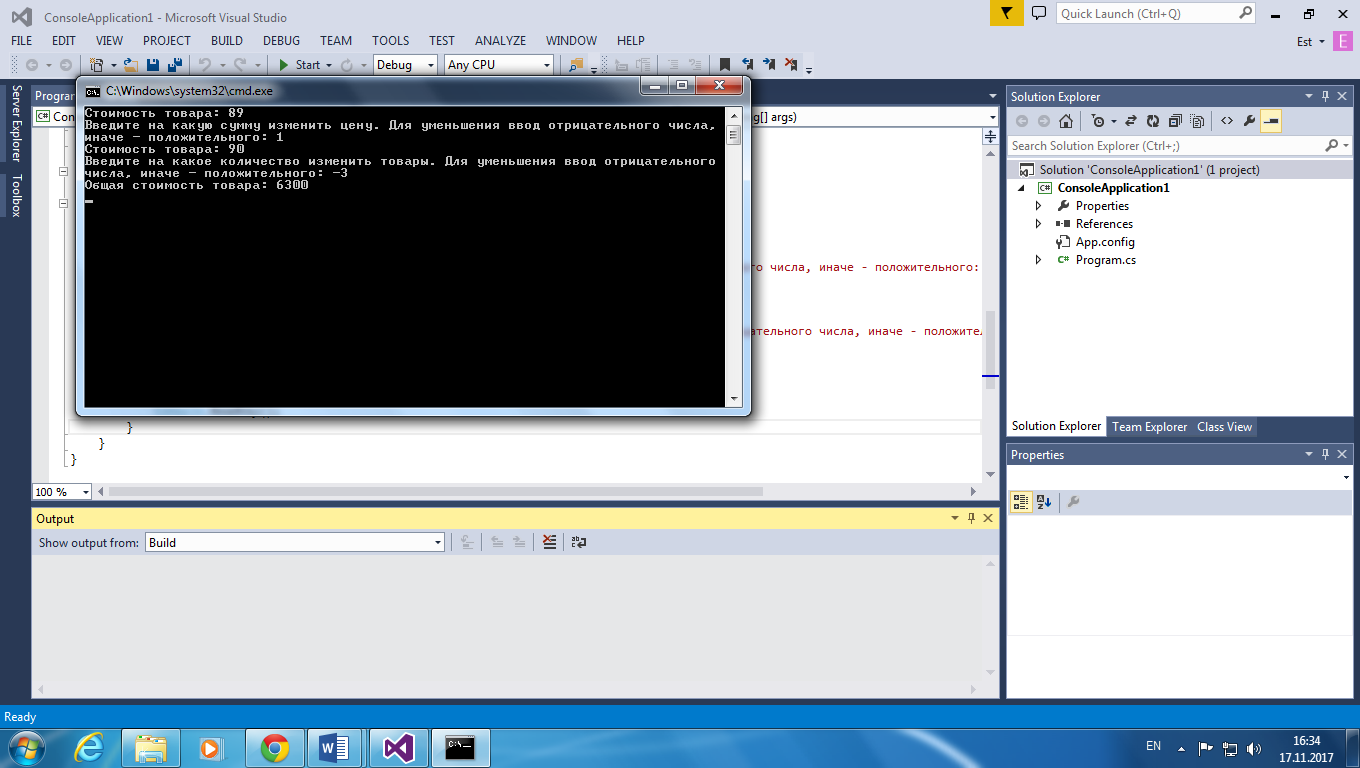


Рисунок 2 – Выполнение программы

Лабораторная работа №10

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка классов и наследование.

Цель: научиться пользоваться возможность наследования в C#.

Задание: построить иерархию классов в соответствии с вариантом задания: транспорт, экспресс, автомобиль, поезд. Расширить иерархию классов из с использованием виртуального класса в качестве основы иерархии. Показать пример использования полиморфизма методов.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace StructApp

{

class Transport

{

public string toplivo;

public virtual void Vyvod()

{

Console.WriteLine("Источник энергии - " + toplivo);

}

}

class Express : Transport

{

public string nomer;

public virtual void Number()

{

Console.WriteLine("Номер маршрута - " + nomer + "\"Э\"");

}

}

class Auto : Express

{

public string mesto;

public virtual void Place()

{

Console.WriteLine("Место - " + mesto);

}

}

class Train : Express

{

public string vagon;

public virtual void Wagon()

{

Console.WriteLine("Вагон - " + vagon);

}

}

class Program

{

static void Main(string[] args)

{

Auto transport1 = new Auto();

transport1.toplivo = "бензин";

transport1.nomer = "134";

transport1.mesto = "19";

transport1.Vyvod();

transport1.Number();

transport1.Place();

Console.WriteLine();

Train transport2 = new Train();

transport2.toplivo = "электричество";

transport2.nomer = "563";

transport2.vagon = "07С";

transport2.Vyvod();

transport2.Number();

transport2.Wagon();

Console.ReadKey();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace StructApp

{

abstract public class Transport

{

public string toplivo;

public virtual void Vyvod()

{

Console.WriteLine("Источник энергии - " + toplivo);

}

}

class Express : Transport

{

public string nomer;

public virtual void Number()

{

Console.WriteLine("Номер маршрута - " + nomer + "\"Э\"");

}

}

class Auto : Express

{

public string mesto;

public virtual void Place()

{

Console.WriteLine("Место - " + mesto);

}

}

class Train : Express

{

public string vagon;

public virtual void Wagon()

{

Console.WriteLine("Вагон - " + vagon);

}

}

class Program

{

static void Main(string[] args)

{

Auto transport1 = new Auto();

transport1.toplivo = "бензин";

transport1.nomer = "134";

transport1.mesto = "19";

transport1.Vyvod();

transport1.Number();

transport1.Place();

Console.WriteLine();

Train transport2 = new Train();

transport2.toplivo = "электричество";

transport2.nomer = "563";

transport2.vagon = "07С";

transport2.Vyvod();

transport2.Number();

transport2.Wagon();

Console.ReadKey();

}

}

Скриншоты выполнения:

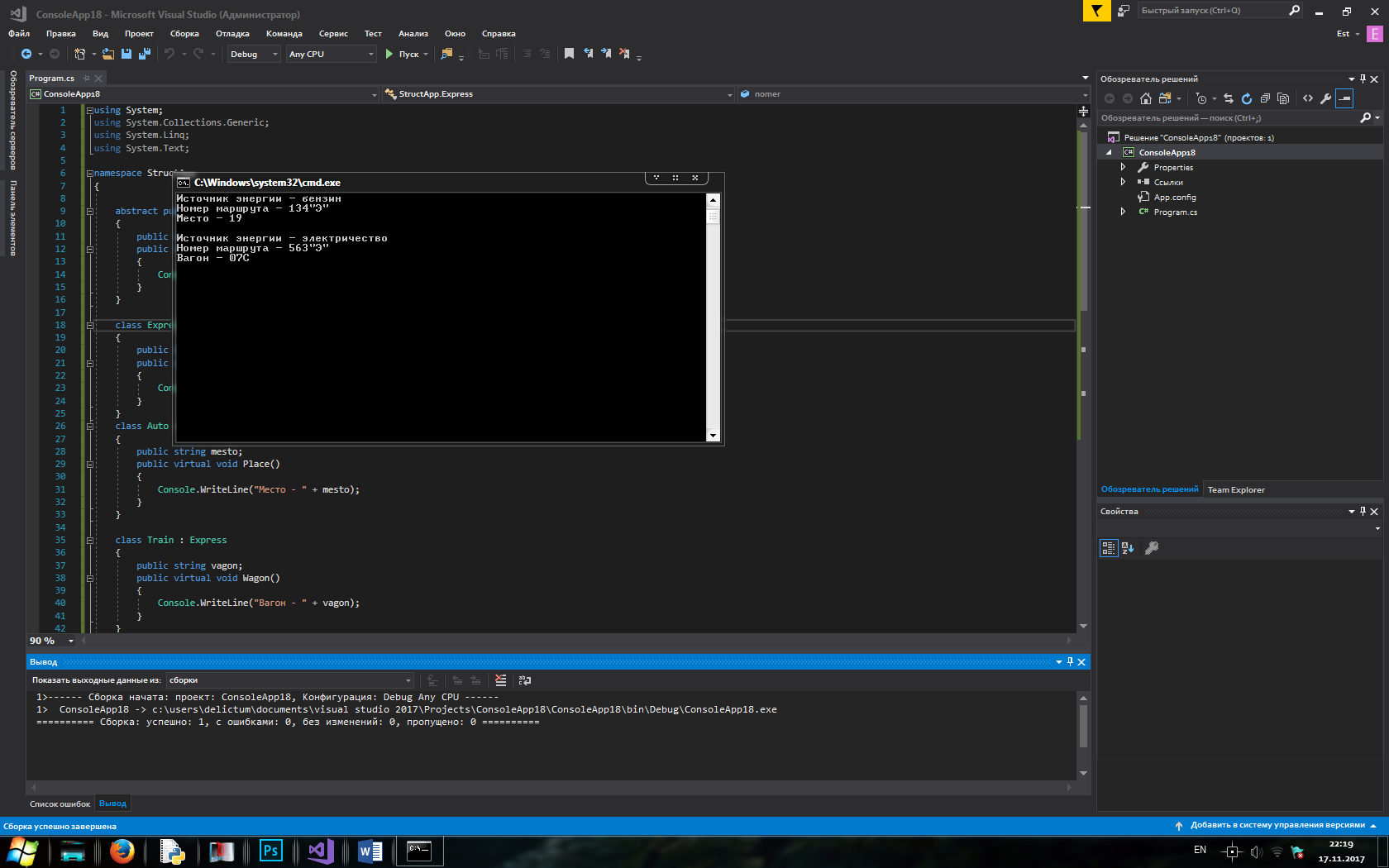


Рисунок 1 – Выполнение программы

Алгоритм: вывод реализованных команд, в зависимости от введённой команды, выполнение соответствующих действий. Запись объектов в коллекцию. Вывод данных из коллекции.

Лабораторная работа №11

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка интерфейсов, исследование механизма наследования интерфейсов и применение их в программе.

Цель: сформировать понятия класса, конструктора класса, деструктора класса и их реализации в С#.

Упражнение 1: добавьте в интерфейс IDemo свойство Y, которое позволит обращаться для чтения к значению поля у. Реализуйте работу с данным свойством в классах DemoPoint и DemoShape. Добавьте свойство Z для обращения к полю z класса DemoShape.

Листинг программы:

using System;

using System.Text.RegularExpressions;

using System.Collections;

using System.Xml.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

using System.IO;

namespace ZKI4

{

interface IDemo

{

void Show();

double Dlina();

int X { get; }

int Y { get; }

int this[int i] { get; set; }

}

class DemoPoint : IDemo

{

protected int x;

protected int y;

public DemoPoint(int x, int y)

{

this.x = x; this.y = y;

}

public void Show()

{

Console.WriteLine("точка на плоскости: ({0}, {1})", x, y);

}

public double Dlina()

{

return Math.Sqrt(x \* x + y \* y);

}

public int X

{

get

{

return x;

}

}

public int Y

{

get

{

return y;

}

}

public int this[int i]

{

get

{

if (i == 0) return x;

else if (i == 1) return y;

else throw new Exception("недопустимое значение индекса");

}

set

{

if (i == 0) x = value;

else if (i == 1) y = value;

else throw new Exception("недопустимое значение индекса");

}

}

}

class DemoShape : DemoPoint, IDemo

{

protected int z;

public int Z

{

get

{

return z;

}

}

public DemoShape(int x, int y, int z) : base(x, y)

{

this.z = z;

}

public new void Show()

{

Console.WriteLine("точка в пространстве: ({0}, {1}, {2})", x, y,

z);

}

public new double Dlina()

{

return Math.Sqrt(x \* x + y \* y + z \* z);

}

public new int this[int i]

{

get

{

if (i == 0) return x;

else if (i == 1) return y;

else if (i == 2) return z;

else throw new Exception("недопустимое значение индекса");

}

set

{

if (i == 0) x = value;

else if (i == 1) y = value;

else if (i == 2) z = value;

else throw new Exception("недопустимое значение индекса");

}

}

}

class Program

{

static void Main()

{

IDemo[] a = new IDemo[4];

a[0] = new DemoPoint(0, 1);

a[1] = new DemoPoint(-3, 0);

a[2] = new DemoShape(3, 4, 0);

a[3] = new DemoShape(0, 5, 6);

foreach (IDemo x in a)

{

x.Show();

Console.WriteLine("Dlina={0:f2}", x.Dlina());

Console.WriteLine("x=" + x.X);

Console.WriteLine("x=" + x.Y);

x[1] += x[0];

Console.Write("новые координаты - ");

x.Show();

Console.WriteLine();

}

Console.ReadKey();

}

}

}

Тестирование:

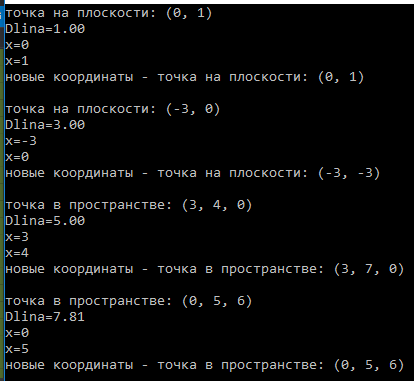


Рисунок 1 – Выполнение программы

Упражнение 2: измените метод CompareTo так, чтобы метод Sort сортировал массив точек по убыванию расстояния между точкой и началом координат.

Листинг программы:

using System;

namespace ZKI4

{

class DemoPoint : IComparable

{

protected int x;

protected int y;

public DemoPoint(int x, int y)

{

this.x = x; this.y = y;

}

public DemoPoint()

{

}

public void Show()

{

Console.WriteLine("точка на плоскости: ({0}, {1})", x, y);

}

int IComparable.CompareTo(object obj)

{

DemoPoint ob = obj as DemoPoint;

if (Dlina > ob.Dlina)

return -1;

else if (Dlina < ob.Dlina)

return 1;

else if (Dlina == ob.Dlina)

return 0;

else

throw new ArgumentException("Object is not a DemoPoint!");

}

public double Dlina

{

get

{

return Math.Sqrt(x \* x + y \* y);

}

}

}

class Program

{

static void Main()

{

try

{

DemoPoint[] a = new DemoPoint[4]

{

//DemoPoint[] a = new DemoPoint[4];

new DemoPoint(5, -1),

new DemoPoint(-3, 3),

new DemoPoint(3, 4),

new DemoPoint(0, 1)

};

Console.WriteLine("До");

foreach (DemoPoint x in a)

{

x.Show();

Console.WriteLine("Dlina={0:f2} ", x.Dlina);

}

Array.Sort(a);

Console.WriteLine("\nПосле");

foreach (DemoPoint x in a)

{

x.Show();

Console.WriteLine("Dlina={0:f2} ", x.Dlina);

}

}

catch (Exception e)

{

Console.WriteLine(e.ToString());

}

finally

{

Console.ReadKey();

}

}

}

}

Скриншот выполнения:

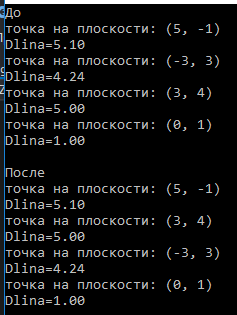


Рисунок 2 – Выполнение программы

Задание: КАРТИНА: ФИО автора, название, жанр, список владельцев.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

interface Realese

{

void FIO(string F);

void Name(string N);

void Zanr(string Z);

void List(string L);

}

class Picture : Realese

{

public string fio;

public string name;

public string zanr;

public string list;

public Picture(string fio, string name, string zanr, string list)

{

this.fio = fio;

this.name = name;

this.zanr = zanr;

this.list = list;

}

public void FIO(string F)

{

Console.WriteLine(F);

}

public void Name(string N)

{

Console.WriteLine(N);

}

public void Zanr(string Z)

{

Console.WriteLine(Z);

}

public void List(string L)

{

Console.WriteLine(L);

}

}

static void Main(string[] args)

{

Picture[] pic = new Picture[2];

for (int i = 0; i < 2; i++)

{

string fio = Console.ReadLine();

string name = Console.ReadLine();

string zanr = Console.ReadLine();

string list = Console.ReadLine();

pic[i] = new Picture(fio, name, zanr, list);

pic[i].FIO(pic[i].fio);

pic[i].Name(pic[i].name);

pic[i].Zanr(pic[i].zanr);

pic[i].List(pic[i].list);

}

}

}

}

Тестирование:

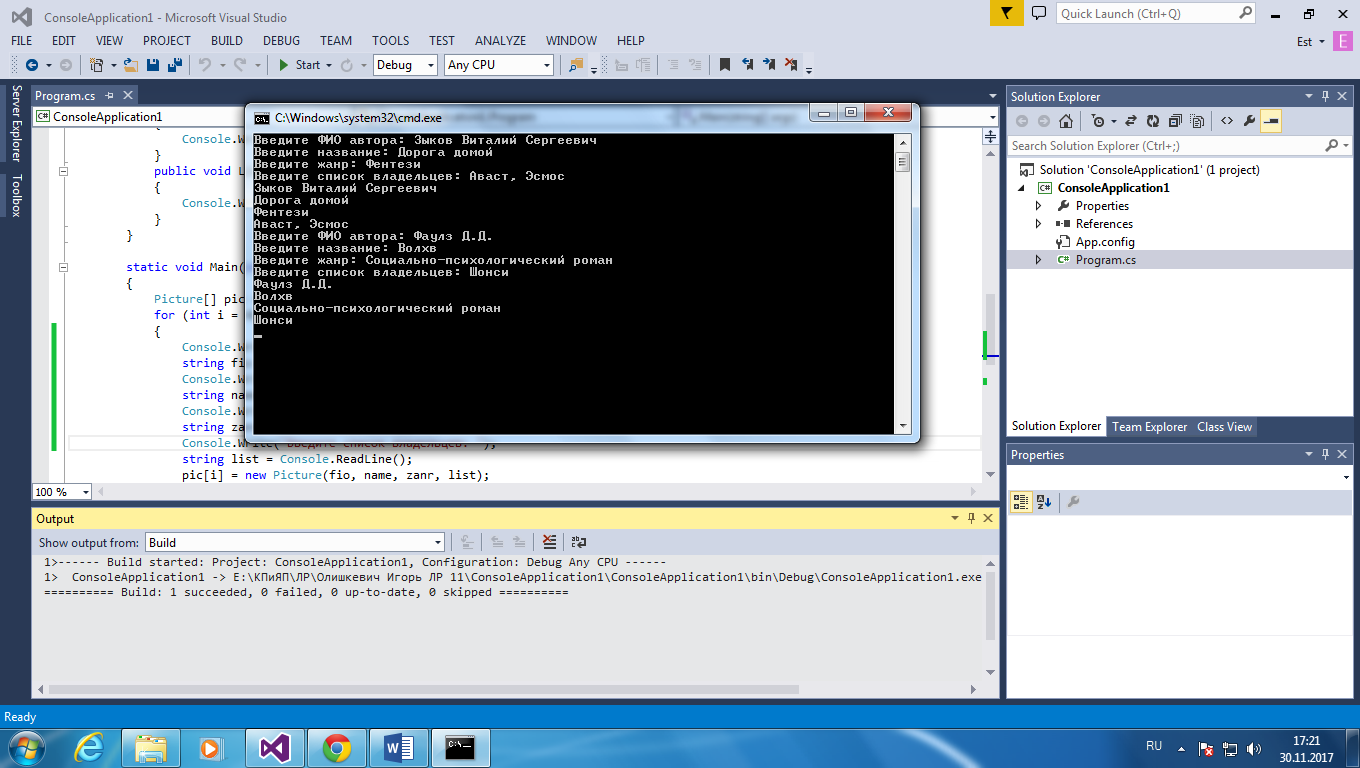


Рисунок 3 – Выполнение программы

Алгоритм: описание интерфейса проектируемого типа, реализация интерфеса, в типе. Создание объкта с данными, полученными из консли. Вывод данных с помощью реализованных методов типа.

Лабораторная работа №12

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка делегатов, событий, анонимных методов и использование их в программах

Цель: сформировать умения инициализации и использования делегата, события и анонимного метода в С#.

Задание: в классы, созданные в лабораторной работе №10, добавить три события, три делегата и три анонимных метода.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

delegate void Deleg1();

delegate void Deleg2();

delegate string Deleg3(string s);

class Events

{

public event Deleg1 Ev1;

public void OnEv1()

{

Ev1();

Console.WriteLine("Шепот: \"Свет меня предал, во тьме я обрету отмщение\".");

}

public event Deleg2 Ev2;

public void OnEv2()

{

Ev2();

Console.WriteLine("Событие произошло.");

}

public event Deleg3 Ev3;

public string OnEv3(string s)

{

s += "! ";

return Ev3(s);

}

}

class Program

{

abstract public class Transport

{

public string toplivo;

public void Vyvod()

{

Console.WriteLine("Источник энергии - " + toplivo);

}

}

class Express : Transport

{

public string nomer;

public virtual void Number()

{

Console.WriteLine("Номер маршрута - " + nomer + "\"Э\"");

}

}

class Auto : Express

{

public string mesto;

public virtual void Place()

{

Console.WriteLine("Место - " + mesto);

Deleg2 del2 = delegate ()

{

Console.WriteLine("Анонимный метод автомобиля.");

};

del2();

}

}

class Train : Express

{

public string vagon;

public virtual void Wagon()

{

Console.WriteLine("Вагон - " + vagon);

Deleg1 del1 = delegate ()

{

Console.WriteLine("Аноним: 'Справедливость восторжествует!'");

};

del1();

}

}

static Deleg3 D(string sss)

{

string ss = "Hey, ";

Deleg3 del3 = delegate (string s)

{

ss += s + "Very well.";

return ss;

};

return del3;

}

static void Main(string[] args)

{

{

Events evt = new Events();

string sss = "good play";

evt.Ev3 += D(sss);

Console.WriteLine("Событие: {0}", evt.OnEv3(sss));

Console.WriteLine();

Auto transport1 = new Auto();

transport1.toplivo = "бензин";

transport1.nomer = "134";

transport1.mesto = "19";

transport1.Vyvod();

evt.Ev2 += transport1.Place;

transport1.Number();

evt.OnEv2();

Console.WriteLine();

Train transport2 = new Train();

transport2.toplivo = "электричество";

transport2.nomer = "563";

transport2.vagon = "07С";

transport2.Vyvod();

evt.Ev1 += transport2.Wagon;

transport2.Number();

evt.OnEv1();

Console.ReadKey();

}

}

}

}

Скриншоты выполнения:

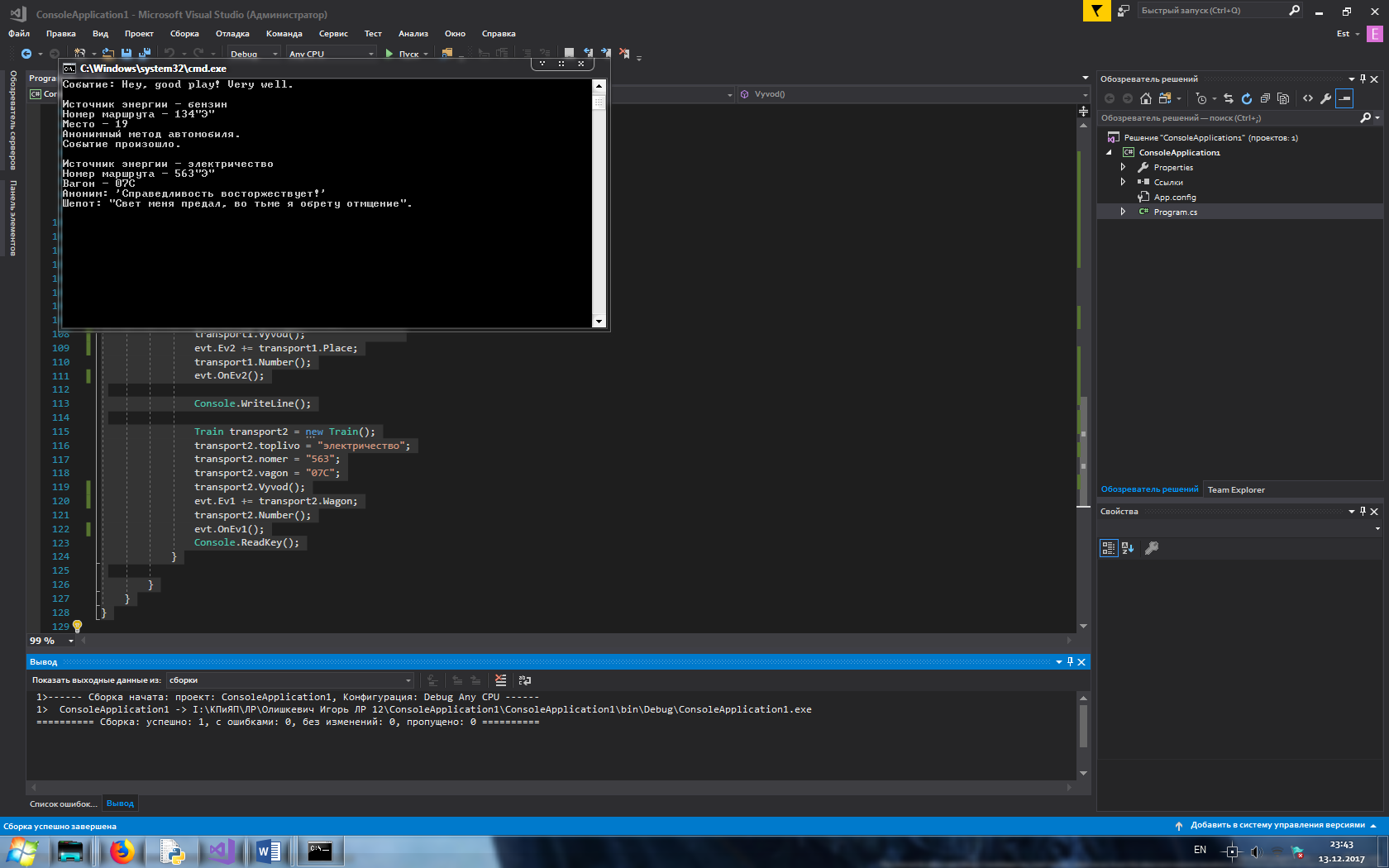


Рисунок 1 – Выполнение программы

Алгоритм: вывод реализованных команд, в зависимости от введённой команды, выполнение соответствующих действий. При выполнении экземплярных методов, выполняется вызов события, на события заранее подписаны анонимные методы. При генерации события выполняются подписанные методы, и с помощью аргументов выводятся необходимая информация об объекте.

Лабораторная работа №13

Выполнил: Олишкевич Игорь Русланович.

Тема: обработка исключительных ситуаций.

Цель: сформировать умения обработки исключительной ситуации в С#.

Упражнение 1: один try-блок можно вложить в другой. Исключение, сгенерированное во внутреннем try-блоке и не перехваченное catch-инструкцией, которая связана с этим try-блоком, передается во внешний try-блок. Например, в следующей программе исключение типа ArithmeticException перехватывается не внутренним try-блоком, а внешним.

Листинг программы:

using System;

using System.Text.RegularExpressions;

using System.Collections;

using System.Xml.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

using System.IO;

namespace ZKI4

{

class Program

{

static void Main()

{

Console.WriteLine("a=");

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

Console.WriteLine("b=");

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

int f = 1;

try //Внешний блок-try

{

for (byte i = a; i <= b; ++i)

{

try //Внутренний блок-try

{

f = checked((int)(f \* i));

Console.WriteLine("y({0})={1:f6}", i, 100 / (f - 1));

}

catch (DivideByZeroException)

{

Console.WriteLine("y({0})=Деление на 0", i);

}

}

}

catch (ArithmeticException){

Console.WriteLine("ERROR");

}

Console.ReadKey();

}

}

}

Скриншот выполнения:

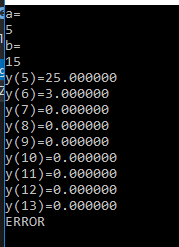


Рисунок 1 – Выполнение программы

Упражнение 2: исключение, перехваченное одной catch-инструкцией, можно сгенерировать повторно, чтобы обеспечить возможность его перехвата другой (внешней) catch-инструкцией. Это позволяет нескольким обработчикам получить доступ к исключению.

Листинг программы:

using System;

using System.Text.RegularExpressions;

using System.Collections;

using System.IO;

namespace ZKI4

{

class Program

{

static void genException()

{

Console.WriteLine("a=");

double a = double.Parse(Console.ReadLine());

Console.WriteLine("b=");

double b = double.Parse(Console.ReadLine());

int f = 1;

try //Внешний блок-try

{

for (double i = a; i <= b; ++i)

{

try //Внутренний блок-try

{

f = checked((int)(f \* i));

Console.WriteLine("y({0})={1:f6}", i, 100 / (f - 1));

}

catch (DivideByZeroException)

{

Console.WriteLine("y({0})=Деление на 0", i);

}

}

}

catch (ArithmeticException)

{

Console.WriteLine("ERROR");

throw; //повторная генерация исключения

}

}

static void Main()

{

try

{

genException();

}

catch

{

Console.WriteLine("НЕИСПРАВИМАЯ ОШИБКА!!!");

}

Console.ReadKey();

}

}

}

Скриншоты выполнения:

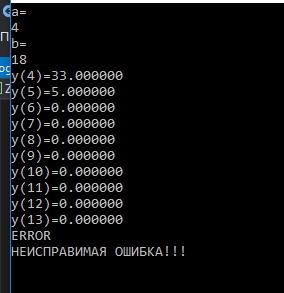


Рисунок 2 – Выполнение программы

Задание: постройте таблицу значений функции y = f(x) для x [a, b] с шагом h. Если в некоторой точке функция не определена, то выведите на экран сообщение об этом. При решении данной задачи использовать вспомогательный метод f(x), реализующий заданную функцию, а также проводить обработку возможных исключений.

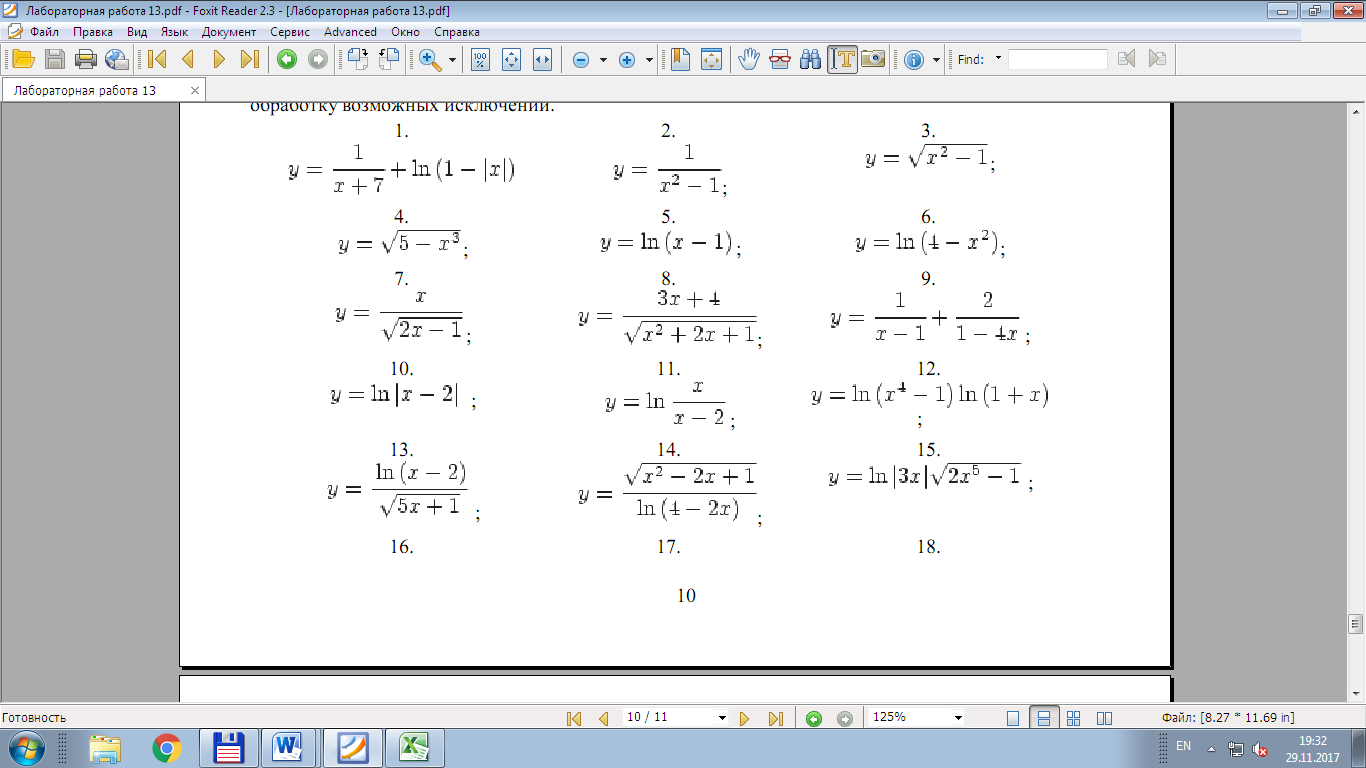


Рисунок 3 – Условие задания

Листинг программы:

using System;

namespace Hello

{

class Program

{

static double f(double x)

{

try

{

if (x <= 2 && x >= 0) throw new Exception();

else

return Math.Log10(x / (x - 2));

}

catch

{

throw;

}

}

static void Main(string[] args)

{

try

{

Console.Write("Введите a: ");

double a = double.Parse(Console.ReadLine());

Console.Write("Введите b: ");

double b = double.Parse(Console.ReadLine());

Console.Write("Введите h: ");

double h = double.Parse(Console.ReadLine());

for (double i = a; i <= b; i += h)

try

{

Console.WriteLine("y({0})={1:f4}", i, f(i));

}

catch

{

Console.WriteLine("y({0})=error", i);

}

}

catch (FormatException)

{

Console.WriteLine("Неверный формат ввода данных");

}

catch

{

Console.WriteLine("Неизвестная ошибка");

}

Console.ReadKey();

}

}

}

Скриншот выполнения:

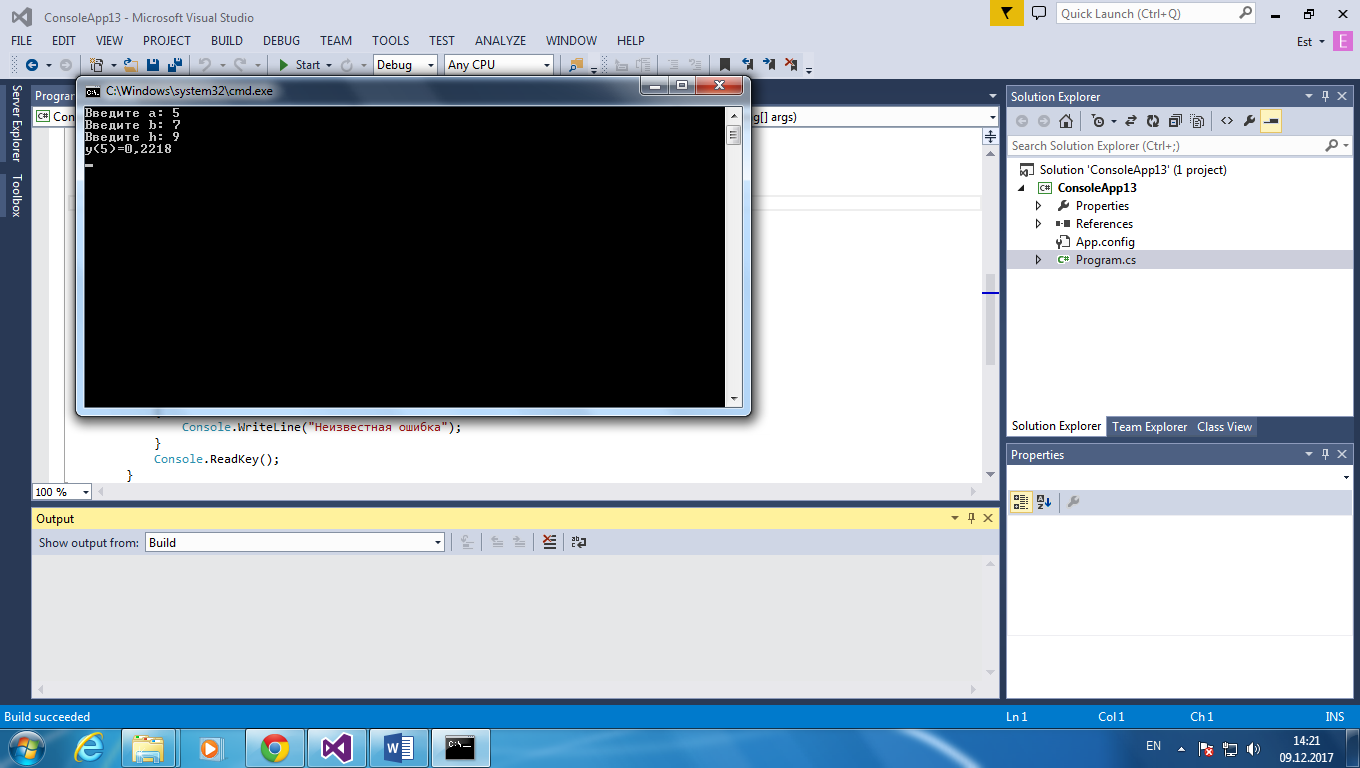


Рисунок 4 – Выполнение программы

Контрольные вопросы:

1. Как осуществляется управление обработкой исключений в С#?

С помощью ключевых слов try, catch, finaly.

1. Как обработать стандартные исключения?

Взять в скобки блок кода, после ключевого слова try, с возможным исключением, в скобках после catch, указать тип обрабатываемого исключения.

1. Каким образом в программе можно управлять генерированием исключений, вызванных переполнением результата в арифметических вычислениях?

С помощь циклов и блока try…catch… finaly.

Лабораторная работа №14

Выполнил: Олишкевич Игорь Русланович.

Тема: разработка программ с использованием файлов.

Цель: научиться применять знания по созданию, корректировке, просмотру и обработке файлов последовательного доступа при решении задач в С#.

Задание: Составить программу для ввода в диалоге значений пере¬менных A, I, C, L, Name и форматного вывода на экран монитора введенных переменных (значення вводимых переменных даны в таблице 3).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 11 | 3,125 | 6006006 | -13,24 | ложь | Имя |

Составить программу для вычисления и печати значений функции из таблицы 4. Вычислить 8 значений функции на задан¬ном интервале. Исходные данные задать в файле LAB2.TXT. Результат поместить в файл вывода с именем LAB2.RES в заданной форме.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 |  | -0,6 | 0,43 | -1 | 1,5 | 3 |

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Задание 1.");

double A = 3.125;

int I = 6006006;

double C = -13.24;

bool L = false;

string Name = "Имя";

Console.WriteLine("Результаты форматирования: \nName = {0:6}, l = {1:4}", Name, L);

Console.WriteLine("a = {0:4}, c = {1:f5}, i = {2:e8}", A, C, I);

Console.WriteLine();

Console.WriteLine("Задание 2.");

string xmin, xmax, x; double y;

StreamWriter f = new StreamWriter("LAB2.res");

StreamReader f1 = new StreamReader("LAB2.txt");

f.WriteLine("Таблица значений функции: ");

xmin = f1.ReadLine();

xmax = f1.ReadLine();

x = f1.ReadLine();

Console.WriteLine("Xmin = {0}, Xmax = {1}, X = {2}", xmin, xmax, x);

f.WriteLine("| Аргумент x | Функция y |");

Console.WriteLine("| Аргумент x | Функция y |");

f.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for (double i = Convert.ToDouble(xmax); i >= Convert.ToDouble(xmin); i += Convert.ToDouble(x))

{

y = Math.Log10(Math.PI) \* Math.Sqrt(Math.Pow(i, 3) + Math.Pow(i, 2));

f.WriteLine("| {0:F3} | {1:e3} |\n", i, y);

Console.WriteLine("| {0:F3} | {1:e3} |", i, y);

}

f.Close();

f1.Close();

Console.ReadKey();

}

}

}

Тестирование:

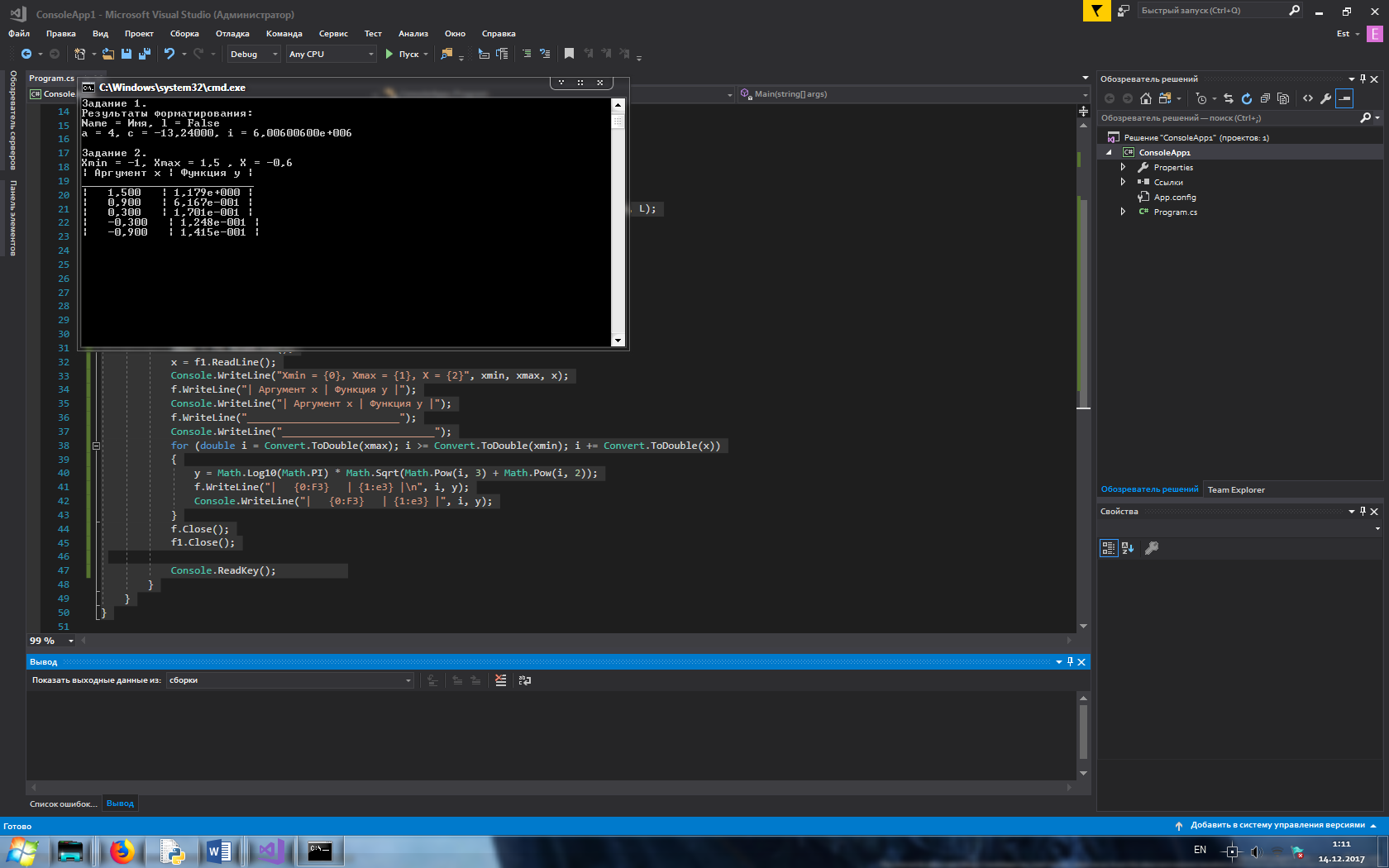


Рисунок 1 – Выполнение программы

Алгоритм:

1. считывание массива байтов из файла, преобразование байтов в стоку.
2. Вычисление функции по заданной формуле, в интервале от 1,5 до 5 с шагом 0,4. Запись данных в строку, вывод данных в консоль, запись строки, преобразованной в байты, в файл LAB2.RES.

Контрольные вопросы:

1. Файл – Специально организованная структура данных, распознаваемая компьютером как единое целое;
2. Множество классов в пространстве имен System.IO;
3. Класс Directory предназначен для работы с директориями, выводом информации директориях
4. Класс File предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла, а также помогает при создании объектов FileStream.

enum FileMode - Указывает, способ доступа к файлу.

Ответы на контрольные вопросы:

1. Файл – это специально организованная структура данных, распознаваемая компьютером как единое целое.

Используются классы File и FileInfo.

1. Класс Directory Предоставляет статические методы для создания, перемещения и перечисления в каталогах и вложенных каталогах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Имя** | **Описание** | |
| System_CAPS_pubmethodSystem_CAPS_static | | [CreateDirectory(String)](https://msdn.microsoft.com/ru-ru/library/54a0at6s(v=vs.110).aspx) | Создает все каталоги и подкаталоги по указанному пути, если они еще не существуют. | |
| System_CAPS_pubmethodSystem_CAPS_static | | [CreateDirectory(String, DirectorySecurity)](https://msdn.microsoft.com/ru-ru/library/9h4z99zb(v=vs.110).aspx) | Создает все каталоги по указанному пути, если они еще не существуют, с применением заданных параметров безопасности Windows. | |
| System_CAPS_pubmethodSystem_CAPS_static | | [Delete(String)](https://msdn.microsoft.com/ru-ru/library/62t64db3(v=vs.110).aspx) | Удаляет пустой каталог по заданному пути. | |
| System_CAPS_pubmethodSystem_CAPS_static | | [Delete(String, Boolean)](https://msdn.microsoft.com/ru-ru/library/fxeahc5f(v=vs.110).aspx) | Удаляет заданный каталог и, при наличии соответствующей инструкции, все подкаталоги и файлы в нем. | |
| System_CAPS_pubmethodSystem_CAPS_static | | [EnumerateDirectories(String)](https://msdn.microsoft.com/ru-ru/library/dd383304(v=vs.110).aspx) | Возвращает перечисляемую коллекцию имен каталогов, расположенных по указанному пути. | |
| System_CAPS_pubmethodSystem_CAPS_static | [Exists(String)](https://msdn.microsoft.com/ru-ru/library/system.io.directory.exists(v=vs.110).aspx) | | | Определяет, указывает ли заданный путь на существующий каталог на диске. |
| System_CAPS_pubmethodSystem_CAPS_static | [GetDirectories(String)](https://msdn.microsoft.com/ru-ru/library/c1sez4sc(v=vs.110).aspx) | | | Возвращает имена подкаталогов (включая пути) в указанном каталоге. |
| System_CAPS_pubmethodSystem_CAPS_static | [GetFiles(String)](https://msdn.microsoft.com/ru-ru/library/07wt70x2(v=vs.110).aspx) | | | Возвращает имена файлов (с указанием пути к ним) в указанном каталоге. |
| System_CAPS_pubmethodSystem_CAPS_static | [Move(String, String)](https://msdn.microsoft.com/ru-ru/library/system.io.directory.move(v=vs.110).aspx) | | | Перемещает файл или каталог со всем его содержимым в новое местоположение. |

1. Класс File предоставляет статические методы для создания, копирования, удаления, перемещения и открытия одного файла, а также помогает при создании объектов [FileStream](https://msdn.microsoft.com/ru-ru/library/system.io.filestream(v=vs.110).aspx).

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubmethodSystem_CAPS_static | [AppendText(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.appendtext(v=vs.110).aspx) | Создает объект [StreamWriter](https://msdn.microsoft.com/ru-ru/library/system.io.streamwriter(v=vs.110).aspx), добавляющий текст с кодировкой UTF-8 в существующий файл, или в новый файл, если указанный файл не существует. |
| System_CAPS_pubmethodSystem_CAPS_static | [Copy(String, String)](https://msdn.microsoft.com/ru-ru/library/c6cfw35a(v=vs.110).aspx) | Копирует существующий файл в новый файл. Перезапись файла с тем же именем не разрешена. |
| System_CAPS_pubmethodSystem_CAPS_static | [Create(String)](https://msdn.microsoft.com/ru-ru/library/d62kzs03(v=vs.110).aspx) | Создает или перезаписывает файл по указанному пути. |
| System_CAPS_pubmethodSystem_CAPS_static | [Delete(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.delete(v=vs.110).aspx) | Удаляет указанный файл. |
| System_CAPS_pubmethodSystem_CAPS_static | [Encrypt(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.encrypt(v=vs.110).aspx) | Шифрует файл таким образом, чтобы его можно было расшифровать только с помощью учетной записи, которая использовалась для шифрования. |
| System_CAPS_pubmethodSystem_CAPS_static | [Exists(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.exists(v=vs.110).aspx) | Определяет, существует ли заданный файл. |
| System_CAPS_pubmethodSystem_CAPS_static | [Move(String, String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.move(v=vs.110).aspx) | Перемещает заданный файл в новое местоположение и разрешает переименование файла. |
| System_CAPS_pubmethodSystem_CAPS_static | [Open(String, FileMode)](https://msdn.microsoft.com/ru-ru/library/b9skfh7s(v=vs.110).aspx) | Открывает объект [FileStream](https://msdn.microsoft.com/ru-ru/library/system.io.filestream(v=vs.110).aspx) по указанному пути с доступом для чтения и записи. |
| System_CAPS_pubmethodSystem_CAPS_static | [OpenRead(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.openread(v=vs.110).aspx) | Открывает для чтения существующий файл. |
| System_CAPS_pubmethodSystem_CAPS_static | [OpenText(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.opentext(v=vs.110).aspx) | Открывает для чтения существующий файл, содержащий текст в кодировке UTF-8. |
| System_CAPS_pubmethodSystem_CAPS_static | [OpenWrite(String)](https://msdn.microsoft.com/ru-ru/library/system.io.file.openwrite(v=vs.110).aspx) | Открывает существующий файл или создает новый файл для записи. |

4) Класс DirectoryInfo – Предоставляет методы экземпляра класса для создания, перемещения и перечисления в каталогах и подкаталогах.

Класс FileInfo – Предоставляет свойства и методы экземпляра для создания, копирования, удаления, перемещения и открытия файлов, а также позволяет создавать объекты [FileStream](https://msdn.microsoft.com/ru-ru/library/system.io.filestream(v=vs.110).aspx).

Свойства:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubproperty | [Attributes](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.attributes(v=vs.110).aspx) | Получает или задает атрибуты для текущего файла или каталога.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [CreationTime](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.creationtime(v=vs.110).aspx) | Получает или задает время создания текущего файла или каталога.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [CreationTimeUtc](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.creationtimeutc(v=vs.110).aspx) | Получает или задает время создания текущего файла или каталога в формате UTC.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [Directory](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.directory(v=vs.110).aspx) | Получает экземпляр родительского каталога. |
| System_CAPS_pubproperty | [DirectoryName](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.directoryname(v=vs.110).aspx) | Получает строку, представляющую полный путь к каталогу. |
| System_CAPS_pubproperty | [Exists](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.exists(v=vs.110).aspx) | Получает значение, показывающее, существует ли файл.(Переопределяет [FileSystemInfo.Exists](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.exists(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [Extension](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.extension(v=vs.110).aspx) | Получает строку, содержащую расширение файла.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [FullName](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.fullname(v=vs.110).aspx) | Получает полный путь к каталогу или файлу.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [IsReadOnly](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.isreadonly(v=vs.110).aspx) | Возвращает или задает значение, позволяющее определить, является ли текущий файл доступным только для чтения. |
| System_CAPS_pubproperty | [LastAccessTime](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.lastaccesstime(v=vs.110).aspx) | Получает или задает время последнего доступа к текущему файлу или каталогу.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [LastAccessTimeUtc](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.lastaccesstimeutc(v=vs.110).aspx) | Получает или задает дату и время последнего доступа к заданному файлу или каталогу в формате UTC.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [LastWriteTime](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.lastwritetime(v=vs.110).aspx) | Получает или задает время последней операции записи в текущий файл или каталог.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [LastWriteTimeUtc](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.lastwritetimeutc(v=vs.110).aspx) | Получает или задает время последней операции записи в текущий файл или каталог в формате UTC.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubproperty | [Length](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.length(v=vs.110).aspx) | Получает размер текущего файла в байтах. |
| System_CAPS_pubproperty | [Name](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.name(v=vs.110).aspx) | Возвращает имя файла.(Переопределяет [FileSystemInfo.Name](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.name(v=vs.110).aspx).) |

Методы:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubmethod | [AppendText()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.appendtext(v=vs.110).aspx) | Создает [StreamWriter](https://msdn.microsoft.com/ru-ru/library/system.io.streamwriter(v=vs.110).aspx), который добавляет текст в файл, представленный этим экземпляром FileInfo.. |
| System_CAPS_pubmethod | [CopyTo(String)](https://msdn.microsoft.com/ru-ru/library/f0e105zt(v=vs.110).aspx) | Копирует существующий файл в новый файл и запрещает перезапись существующего файла. |
| System_CAPS_pubmethod | [Create()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.create(v=vs.110).aspx) | Создает файл. |
| System_CAPS_pubmethod | [Decrypt()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.decrypt(v=vs.110).aspx) | Расшифровывает файл, зашифрованный текущей учетной записью с помощью метода [Encrypt](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.encrypt(v=vs.110).aspx). |
| System_CAPS_pubmethod | [Delete()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.delete(v=vs.110).aspx) | Удаляет файл без возможности восстановления.(Переопределяет [FileSystemInfo.Delete()](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.delete(v=vs.110).aspx).) |
| System_CAPS_pubmethod | [Encrypt()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.encrypt(v=vs.110).aspx) | Шифрует файл таким образом, чтобы его можно было расшифровать только с помощью учетной записи, которая использовалась для шифрования. |
| System_CAPS_pubmethod | [Equals(Object)](https://msdn.microsoft.com/ru-ru/library/bsc2ak47(v=vs.110).aspx) | Определяет, равен ли заданный объект текущему объекту.(Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx). |
| System_CAPS_pubmethod | [MoveTo(String)](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.moveto(v=vs.110).aspx) | Перемещает заданный файл в новое местоположение и разрешает переименование файла. |
| System_CAPS_pubmethod | [Open(FileMode)](https://msdn.microsoft.com/ru-ru/library/ya63c7c5(v=vs.110).aspx) | Открывает файл в заданном режиме. |
| System_CAPS_pubmethod | [GetType()](https://msdn.microsoft.com/ru-ru/library/system.object.gettype(v=vs.110).aspx) | Возвращает объект [Type](https://msdn.microsoft.com/ru-ru/library/system.type(v=vs.110).aspx) для текущего экземпляра.(Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |
| System_CAPS_pubmethod | [OpenRead()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.openread(v=vs.110).aspx) | Создает доступный только для чтения поток [FileStream](https://msdn.microsoft.com/ru-ru/library/system.io.filestream(v=vs.110).aspx). |
| System_CAPS_pubmethod | [OpenText()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.opentext(v=vs.110).aspx) | Создает поток [StreamReader](https://msdn.microsoft.com/ru-ru/library/system.io.streamreader(v=vs.110).aspx) с кодировкой UTF-8, который считывает данные из существующего текстового файла. |
| System_CAPS_pubmethod | [OpenWrite()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.openwrite(v=vs.110).aspx) | Создает доступный только для записи поток [FileStream](https://msdn.microsoft.com/ru-ru/library/system.io.filestream(v=vs.110).aspx). |
| System_CAPS_pubmethod | [Refresh()](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo.refresh(v=vs.110).aspx) | Обновляет состояние объекта.(Наследуется от [FileSystemInfo](https://msdn.microsoft.com/ru-ru/library/system.io.filesysteminfo(v=vs.110).aspx).) |
| System_CAPS_pubmethod | [Replace(String, String)](https://msdn.microsoft.com/ru-ru/library/8a3t4ek5(v=vs.110).aspx) | Заменяет содержимое заданного файла на содержимое файла, которое описано в текущем объекте FileInfo, удаляет исходный файл и создает резервную копию замененного файла. |
| System_CAPS_pubmethod | [ToString()](https://msdn.microsoft.com/ru-ru/library/system.io.fileinfo.tostring(v=vs.110).aspx) | Возвращает путь в виде строки.(Переопределяет [Object.ToString()](https://msdn.microsoft.com/ru-ru/library/system.object.tostring(v=vs.110).aspx).) |

Лабораторная работа №15

Выполнил: Олишкевич Игорь Русланович.

Тема: создание библиотек.

Цель: научиться применять знания по созданию, корректировке, просмотру и обработке файлов последовательного доступа при решении задач в С#.

Задание: выполнить индивидуальное задание (создать собственную библиотеку на основе лабораторной работы № 9).

Основная программа:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

var g = new Goods.Goods("Йогурт", DateTime.Today, 89, 73, 124235);

g.ToString();

Console.Write("Введите на какую сумму изменить цену. Для уменьшения ввод отрицательного числа, иначе - положительного: ");

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

g.DisPrice(buf);

g.ToString();

Console.Write("Введите на какое количество изменить товары. Для уменьшения ввод отрицательного числа, иначе - положительного: ");

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

g.DisCount(buf);

g.ResPrice();

Console.ReadKey();

}

}

}

Библиотека:

using System;

namespace Goods

{

public class Goods

{

string Name;

DateTime DateO;

int Price, Count, NumerN;

public Goods(string Name, DateTime DateO, int Price, int Count, int NumerN)

{

this.Name = Name;

this.DateO = DateO;

this.Price = Price;

this.Count = Count;

this.NumerN = NumerN;

}

public double DisPrice(int n)

{

return Price += n;

}

public double DisCount(int n)

{

return Count += n;

}

public void ResPrice()

{

double n = Price \* Count;

Console.WriteLine("Общая стоимость товара: {0}", n);

}

public void ToString()

{

Console.WriteLine("Стоимость товара: {0}", Price);

}

}

}

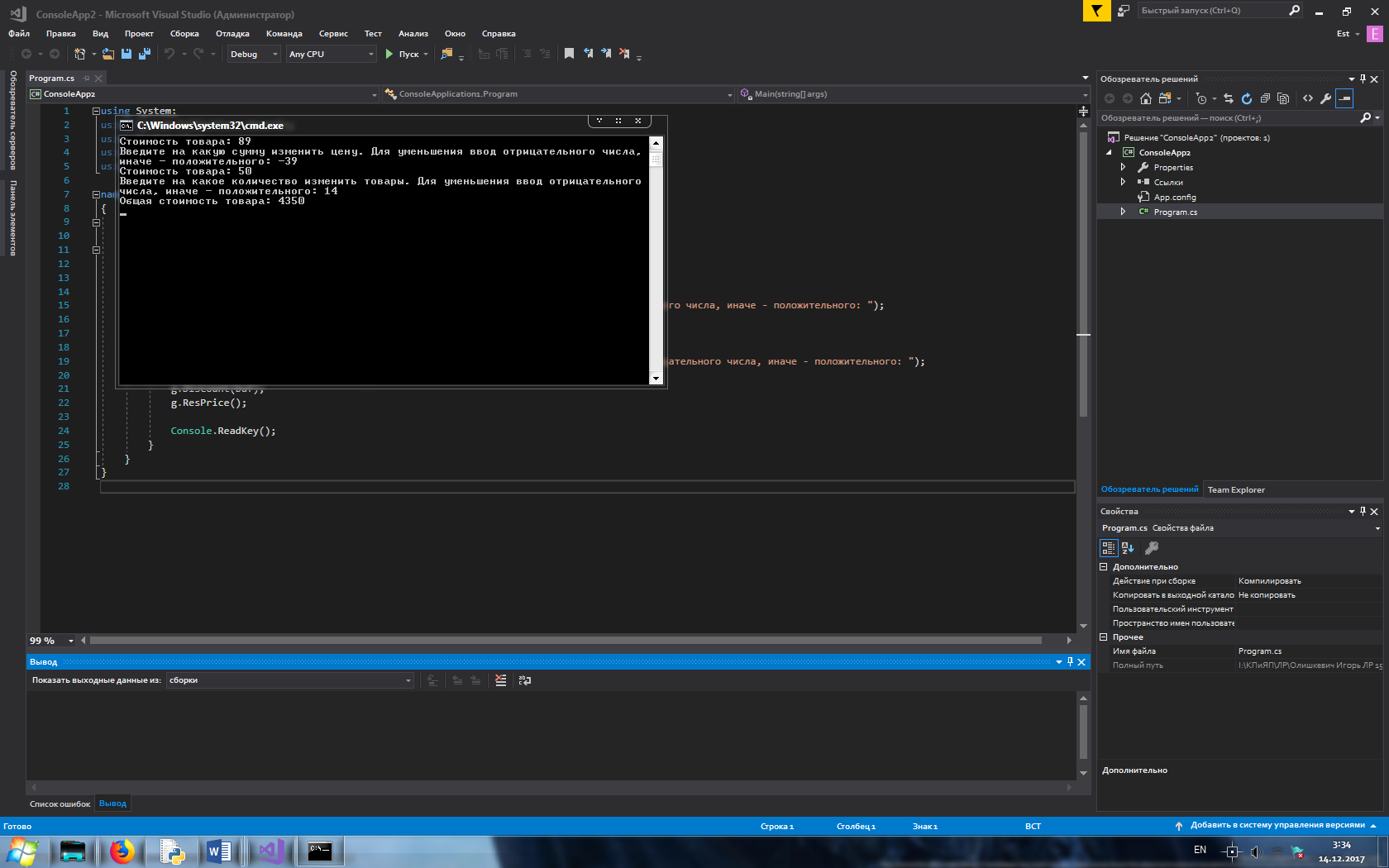


Рисунок 1 – Выполнение программы

Алгоритм: вывод реализованных команд, в зависимости от введённой команды, выполнение соответствующих действий. Запись объектов в коллекцию. Вывод данных из коллекции. Исходный код, сконструированных типов, компилируем в dll, все типы с модификатором доступа public. В основной программе добавляем в ссылки и пространство имен библиотеку, компилируем, запускаем.

Лабораторная работа №16

Выполнил: Олишкевич Игорь Русланович.

Тема: реализация механизма создания коллекции и применение коллекций в программах.

Цель: изучить принципы разработки программ с использованием коллекций в С#.

Задаение: создайте в проекте коллекцию mas1, содержащую список объектов типа int. Добавьте в коллекцию три элемента, ваши данные: Число\_рождения, Месяц\_рождения, Год\_рождения. Выведите элементы коллекции на экран.

Создайте в проекте коллекцию mas2, ваши данные: Фамилия, Имя, Отчество. Выведите элементы коллекции на экран.

Создайте в проекте параметризованный метод Sort с параметром – типом данных. Реализуйте в нем сортировку методом выбора (алгоритм ее состоит в том, что сначала выбирается наименьший элемент массива и меняется местами с первым элементом. Затем просматриваются элементы, начиная со второго, и наименьший из них меняется местами со вторым элементом и так далее. Всего произведено будет n-1 замен.

Основная программа:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CL1;

namespace ConsoleApplication1

{

public class Sorted

{

public Sorted(int[] a, double[] b, string[] c)

{

this.a = new int[a.GetLength(0)];

this.a = a;

this.b = new double[b.GetLength(0)];

this.b = b;

this.c = new string[c.GetLength(0)];

this.c = c;

}

public virtual void SW()

{

CompareTo();

Console.WriteLine("Массив a: {0}", a);

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

Console.Write("{0} ", a[i]);

Console.WriteLine();

Console.WriteLine("Массив b: {0}", b);

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

Console.Write("{0} ", b[i]);

Console.WriteLine();

Console.WriteLine("Массив c: {0}", c);

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

Console.Write("{0} ", c[i]);

Console.WriteLine();

}

private int CompareTo()

{

Array.Sort(a);

Array.Sort(b);

Array.Sort(c);

return (0);

}

int[] a;

double[] b;

string[] c;

}

class Program

{

static void Main(string[] args)

{

List<Monster> mas = new List<Monster>();

mas.Add(new Monster(2, 1, "Здельгас"));

mas.Add(new Daemon(3, 5, "Гернун", 7));

foreach (Monster x in mas)

x.Passport();

List<BD> mas1 = new List<BD>();

mas1.Add(new BD(28, 10, 1997));

foreach (BD x in mas1)

x.BDW();

List<FIO> mas2 = new List<FIO>();

mas2.Add(new FIO("Олишкевич", "Игорь", "Русланович"));

foreach (FIO x in mas2)

x.FIOW();

int[] a = { 1, 6, 4, 2, 7, 5, 3 };

double[] b = { 1.1, 6.6, 4.4, 2.4, 7.6, 5.5, 3.3 };

string[] c = { "sdef", "sd", "sdfsd", "sdf" };

List<Sorted> mas3 = new List<Sorted>();

mas3.Add(new Sorted(a, b, c));

foreach (Sorted x in mas3)

x.SW();

}

}

}

Библиотека:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CL1{

public class BD

{

public BD(int d, int m, int y)

{

this.d = d;

this.m = m;

this.y = y;

}

public virtual void BDW()

{

Console.WriteLine("Дата рождения:\n {0} {1} {2}", d, m, y);

}

public int D

{

get { return d; }

set

{

if (value > 0)

d = value;

else

d = 0;

}

}

public int M

{

get { return m; }

set

{

if (value > 0)

m = value;

else

m = 0;

}

}

public int Y

{

get { return y; }

set

{

if (value > 0)

y = value;

else

y = 0;

}

}

int d, m, y;

}

public class FIO

{

public FIO(string f, string i, string o)

{

this.f = f;

this.i = i;

this.o = o;

}

public virtual void FIOW()

{

Console.WriteLine("ФИО:\n {0} {1} {2}", f, i, o);

}

public string F

{

get {return f;}

set

{

if (value != null)

f = value;

else

f = "";

}

}

public string I

{

get {return i;}

set

{

if (value != null)

i = value;

else

i = "";

}

}

public string O

{

get {return o;}

set

{

if (value != null)

o = value;

else

o = "";

}

}

string f, i, o;

}

public class Monster

{

public Monster(int streng, int skill, string name)

{

this.streng = streng;

this.skill = skill;

this.name = name;

}

public virtual void Passport()

{

Console.WriteLine("Монстр {0} \t сила = {1} умение= {2}", name, streng, skill);

}

public int Streng

{

get { return streng; }

set

{

if (value > 0)

streng = value;

else

streng = 0;

}

}

public int Skill

{

get { return skill; }

set

{

if (value > 0)

skill = value;

else

skill = 0;

}

}

public string Name

{

get { return name; }

}

string name;

int streng, skill;

}

public class Daemon : Monster

{

public Daemon(int streng, int skill, string name, int brain)

: base(streng, skill, name)

{

this.brain = brain;

}

public override void Passport()

{

Console.WriteLine("Демон {0} сила = {1} умение {2} ум = {3}", Name, Streng, Skill, brain);

}

public void Think()

{

Console.Write(Name + "это ");

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

Console.Write(" думает ");

Console.WriteLine("...");

}

int brain;

}

}

Скриншот выполнения:

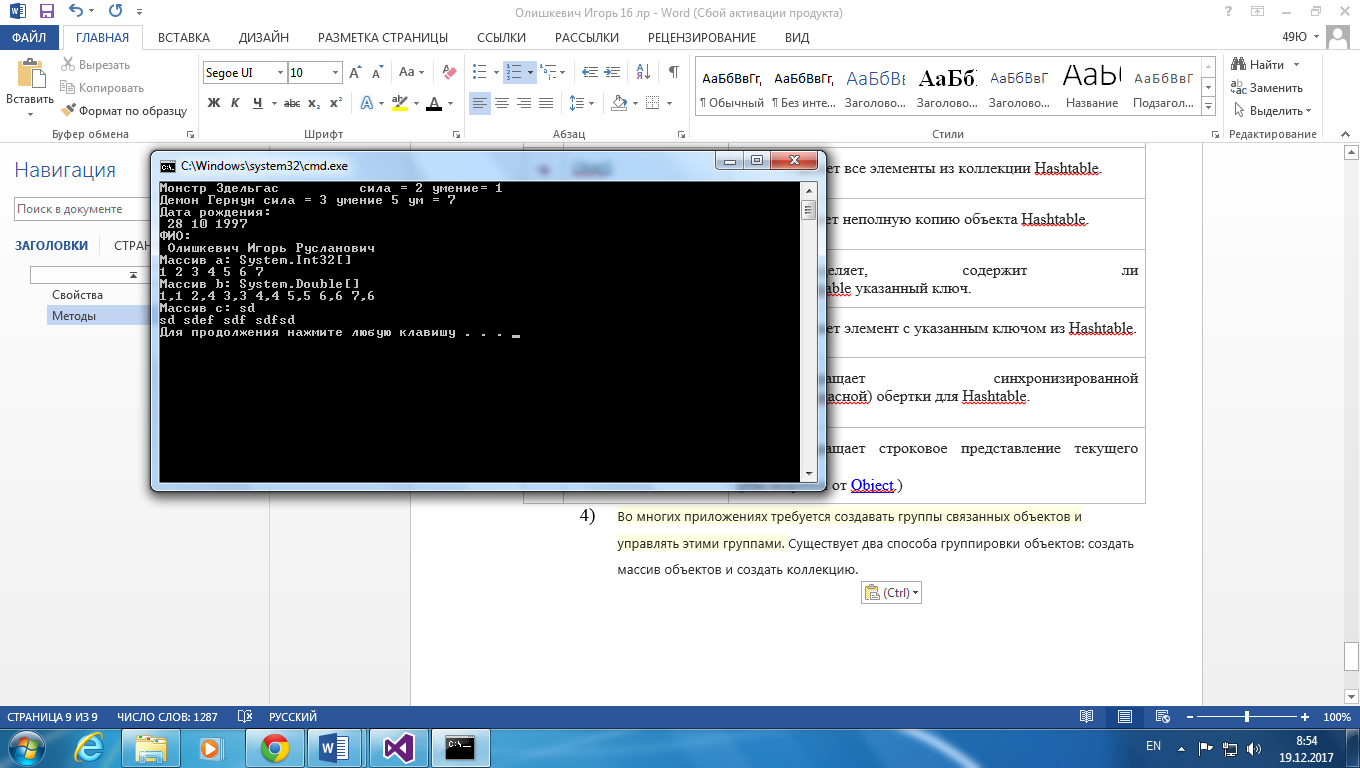


Рисунок 1 – Выполнение программы

Алгоритм: создание списка, принимающих тип «int», добавление трех элементов (дата, месяц, год рождения), вывод данных, содержащихся всписке.

В методе «SelectionSort» обобщенный тип приводится к частному, а именно к «List<int>» сначала выбирается наименьший элемент массива и меняется местами с первым элементом. Затем просматриваются элементы, начиная со второго, и наименьший из них меняется местами со вторым элементом и так далее. Всего произведено будет n-1 замен.

Контрольные вопросы:

1. Представляет простую необобщенную коллекцию объектов, работающую по принципу «последним вошел— первым вышел».  
   Свойства:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubproperty | [Count](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.count(v=vs.110).aspx) | Получает число элементов, содержащихся в интерфейсе Stack. |
| System_CAPS_pubproperty | [IsSynchronized](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.issynchronized(v=vs.110).aspx) | Возвращает значение, показывающее, является ли доступ к коллекции Stack синхронизированным (потокобезопасным). |
| System_CAPS_pubproperty | [SyncRoot](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.syncroot(v=vs.110).aspx) | Получает объект, с помощью которого можно синхронизировать доступ к коллекции Stack. |

Методы:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubmethod | [Peek()](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.peek(v=vs.110).aspx) | Возвращает объект в верхней части Stack без его удаления. |
| System_CAPS_pubmethod | [Pop()](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.pop(v=vs.110).aspx) | Удаляет и возвращает объект, находящийся в начале Stack. |
| System_CAPS_pubmethod | [Push(Object)](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.push(v=vs.110).aspx) | Вставляет объект как верхний элемент стека Stack. |
| System_CAPS_pubmethodSystem_CAPS_static | [Synchronized(Stack)](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.synchronized(v=vs.110).aspx) | Возвращает синхронизированную (потокобезопасную) оболочку для Stack. |
| System_CAPS_pubmethod | [ToArray()](https://msdn.microsoft.com/ru-ru/library/system.collections.stack.toarray(v=vs.110).aspx) | Копирует Stack в новый массив. |
| System_CAPS_pubmethod | [ToString()](https://msdn.microsoft.com/ru-ru/library/system.object.tostring(v=vs.110).aspx) | Возвращает строковое представление текущего объекта. (Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |

1. Представляет коллекцию объектов, основанную на принципе "первым вошёл — первым вышел".

Свойства:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubproperty | [Count](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.count(v=vs.110).aspx) | Получает число элементов, содержащихся в интерфейсе Queue. |
| System_CAPS_pubproperty | [IsSynchronized](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.issynchronized(v=vs.110).aspx) | Возвращает значение, показывающее, является ли доступ к коллекции Queue синхронизированным (потокобезопасным). |
| System_CAPS_pubproperty | [SyncRoot](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.syncroot(v=vs.110).aspx) | Получает объект, с помощью которого можно синхронизировать доступ к коллекции Queue. |

Методы:

|  |  |  |
| --- | --- | --- |
| System_CAPS_pubmethod | [GetType()](https://msdn.microsoft.com/ru-ru/library/system.object.gettype(v=vs.110).aspx) | Возвращает объект [Type](https://msdn.microsoft.com/ru-ru/library/system.type(v=vs.110).aspx) для текущего экземпляра.(Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |
| System_CAPS_protmethod | [MemberwiseClone()](https://msdn.microsoft.com/ru-ru/library/system.object.memberwiseclone(v=vs.110).aspx) | Создает неполную копию текущего объекта [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).(Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |
| System_CAPS_pubmethod | [Peek()](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.peek(v=vs.110).aspx) | Возвращает объект, находящийся в начале очереди Queue, но не удаляет его. |
| System_CAPS_pubmethodSystem_CAPS_static | [Synchronized(Queue)](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.synchronized(v=vs.110).aspx) | Возвращает новый Queue включает исходную очередь и является потокобезопасным. |
| System_CAPS_pubmethod | [ToArray()](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.toarray(v=vs.110).aspx) | Копирует элементы Queue в новый массив. |
| System_CAPS_pubmethod | [ToString()](https://msdn.microsoft.com/ru-ru/library/system.object.tostring(v=vs.110).aspx) | Возвращает строковое представление текущего объекта. (Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |
| System_CAPS_pubmethod | [TrimToSize()](https://msdn.microsoft.com/ru-ru/library/system.collections.queue.trimtosize(v=vs.110).aspx) | Задает значение емкости, равное действительному количеству элементов в Queue. |

1. Предоставляет коллекцию пар «ключ-значение», которые упорядочены по хэш-коду ключа.

## Свойства

|  |  |  |
| --- | --- | --- |
|  | **Имя** | **Описание** |
| System_CAPS_protproperty | [comparer](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.comparer(v=vs.110).aspx) | **Устарело.** Возвращает или задает [IComparer](https://msdn.microsoft.com/ru-ru/library/system.collections.icomparer(v=vs.110).aspx) для Hashtable. |
| System_CAPS_pubproperty | [Count](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.count(v=vs.110).aspx) | Возвращает число пар "ключ-значение", содержащихся в словаре Hashtable. |
| System_CAPS_protproperty | [EqualityComparer](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.equalitycomparer(v=vs.110).aspx) | Возвращает [IEqualityComparer](https://msdn.microsoft.com/ru-ru/library/system.collections.iequalitycomparer(v=vs.110).aspx) для Hashtable. |
| System_CAPS_protproperty | [hcp](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.hcp(v=vs.110).aspx) | **Устарело.** Получает или задает объект, который может распределять хэш-коды. |
| System_CAPS_pubproperty | [IsFixedSize](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.isfixedsize(v=vs.110).aspx) | Получает значение, указывающее, имеет ли список Hashtable фиксированный размер. |
| System_CAPS_pubproperty | [IsReadOnly](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.isreadonly(v=vs.110).aspx) | Получает значение, указывающее, является ли объект Hashtable доступным только для чтения. |
| System_CAPS_pubproperty | [IsSynchronized](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.issynchronized(v=vs.110).aspx) | Возвращает значение, показывающее, является ли доступ к коллекции Hashtable синхронизированным (потокобезопасным). |
| System_CAPS_pubproperty | [Item[Object]](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.item(v=vs.110).aspx) | Возвращает или задает значение, связанное с указанным ключом. |
| System_CAPS_pubproperty | [Keys](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.keys(v=vs.110).aspx) | Возвращает [ICollection](https://msdn.microsoft.com/ru-ru/library/system.collections.icollection(v=vs.110).aspx) с ключами в Hashtable. |
| System_CAPS_pubproperty | [SyncRoot](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.syncroot(v=vs.110).aspx) | Получает объект, с помощью которого можно синхронизировать доступ к коллекции Hashtable. |
| System_CAPS_pubproperty | [Values](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.values(v=vs.110).aspx) | Возвращает интерфейс [ICollection](https://msdn.microsoft.com/ru-ru/library/system.collections.icollection(v=vs.110).aspx), содержащий значения из Hashtable. |

## Методы

|  |  |  |
| --- | --- | --- |
|  | **Имя** | **Описание** |
| System_CAPS_pubmethod | [Add(Object, Object)](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.add(v=vs.110).aspx) | Добавляет элемент с указанными ключом и значением в словарь Hashtable. |
| System_CAPS_pubmethod | [Clear()](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.clear(v=vs.110).aspx) | Удаляет все элементы из коллекции Hashtable. |
| System_CAPS_pubmethod | [Clone()](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.clone(v=vs.110).aspx) | Создает неполную копию объекта Hashtable. |
| System_CAPS_pubmethod | [Contains(Object)](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.contains(v=vs.110).aspx) | Определяет, содержит ли объект Hashtable указанный ключ. |
| System_CAPS_pubmethod | [Remove(Object)](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.remove(v=vs.110).aspx) | Удаляет элемент с указанным ключом из Hashtable. |
| System_CAPS_pubmethodSystem_CAPS_static | [Synchronized(Hashtable)](https://msdn.microsoft.com/ru-ru/library/system.collections.hashtable.synchronized(v=vs.110).aspx) | Возвращает синхронизированной (потокобезопасной) обертки для Hashtable. |
| System_CAPS_pubmethod | [ToString()](https://msdn.microsoft.com/ru-ru/library/system.object.tostring(v=vs.110).aspx) | Возвращает строковое представление текущего объекта. (Наследуется от [Object](https://msdn.microsoft.com/ru-ru/library/system.object(v=vs.110).aspx).) |

1. Во многих приложениях требуется создавать группы связанных объектов и управлять этими группами. Существует два способа группировки объектов: создать массив объектов и создать коллекцию. Коллекции предоставляют более гибкий способ работы с группами объектов. В отличие от массивов, группа объектов в классе может динамически возрастать и сокращаться в соответствии с потребностями приложения. Некоторые коллекции допускают назначение ключа любому объекту, который добавляется в коллекцию, чтобы в дальнейшем можно было быстро извлечь связанный с ключом объект из коллекции. Коллекция является классом, поэтому необходимо объявить новую коллекцию перед добавлением в неё элементов. Если коллекция содержит элементы только одного типа данных, можно использовать один из классов в пространстве имен System.Collections.Generic. Универсальная коллекция обеспечивает безопасность типов, так что другие типы данных не могут быть в нее добавлены. При извлечении элемента из универсальной коллекции нет необходимости определять или преобразовывать его тип данных.

Лабораторная работа №17

Выполнил: Олишкевич Игорь Русланович.

Тема: pеализация механизмов работы в сети.

Цель: изучить принципы разработки программ с классов для работы в сети в С#.

Задание 1: вывести HTML код заданного сайта.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.IO;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

WebClient client = new WebClient();

Stream stream = client.OpenRead("http://www.1c.easyprog.ru");

StreamReader sr = new StreamReader(stream, Encoding.Default);

string newLine;

while ((newLine = sr.ReadLine()) != null)

Console.WriteLine(newLine);

stream.Close();

string hostname = "www.google.com", message = "IP адреса для домена " + hostname + "\n";

IPHostEntry entry = Dns.GetHostEntry(hostname);

foreach (IPAddress a in entry.AddressList)

message += " --> " + a.ToString() + "\n";

message += "\nАльтернативное имя домена: ";

foreach (string aliasName in entry.Aliases)

message += aliasName + "\n";

message += "\nРеальное название хоста: " + entry.HostName;

Console.WriteLine(message);

}

}

}

Тестирование:

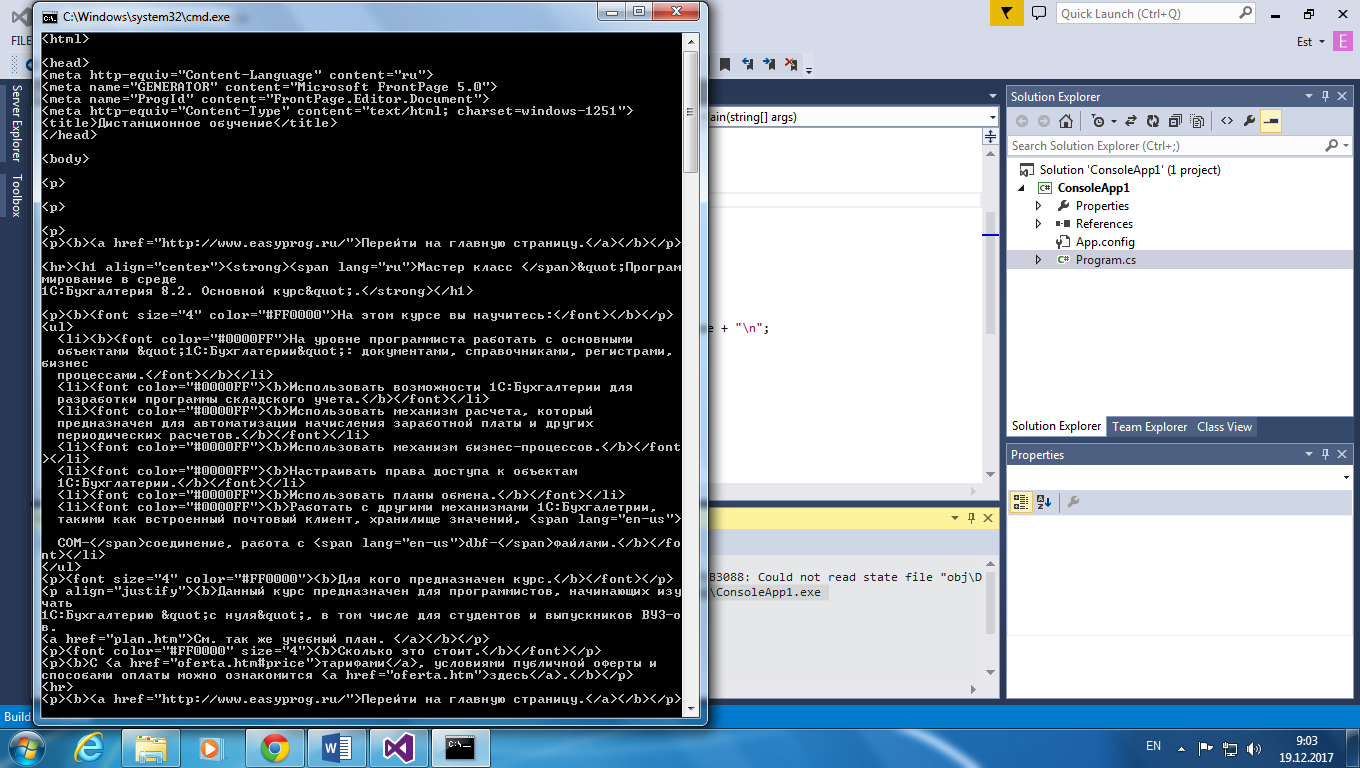


Рисунок 1 – Первая часть выполнения программы

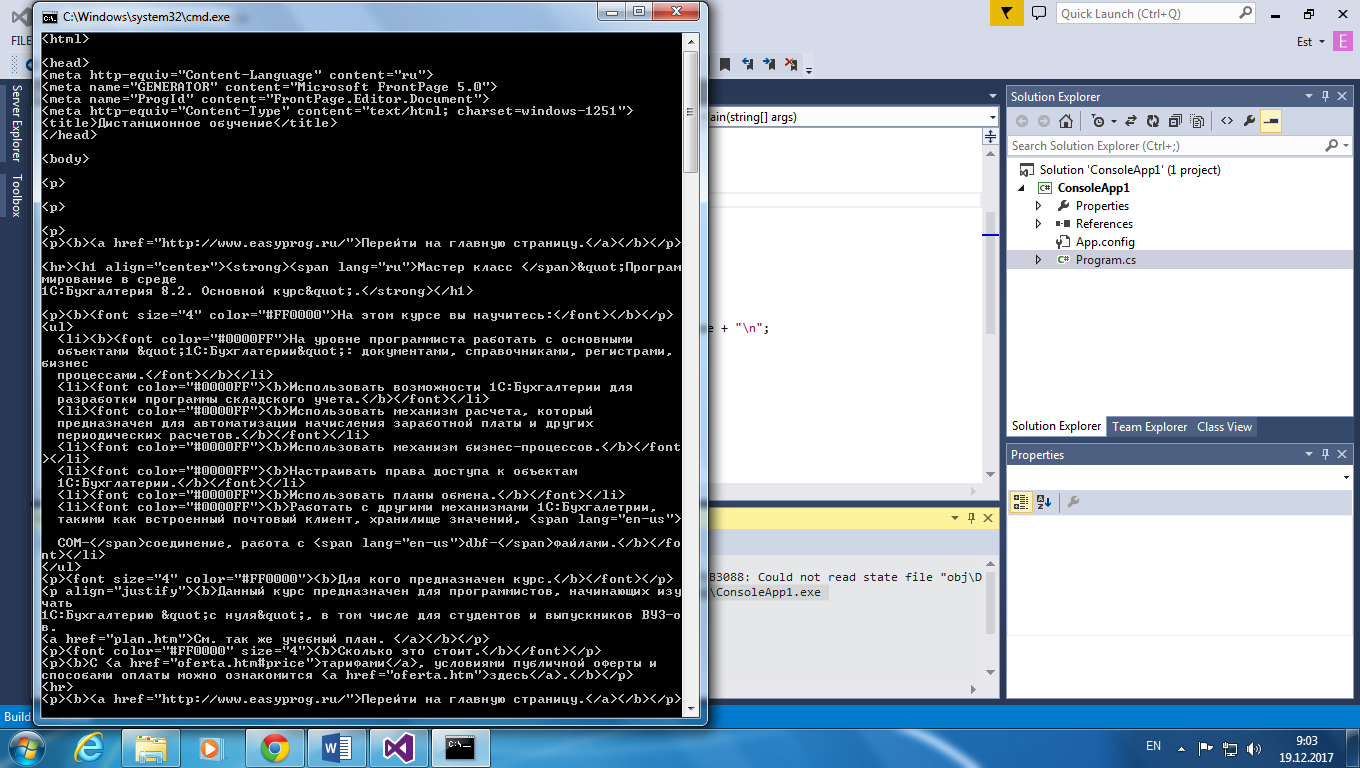


Рисунок 2 – Вторая часть выполнения программы

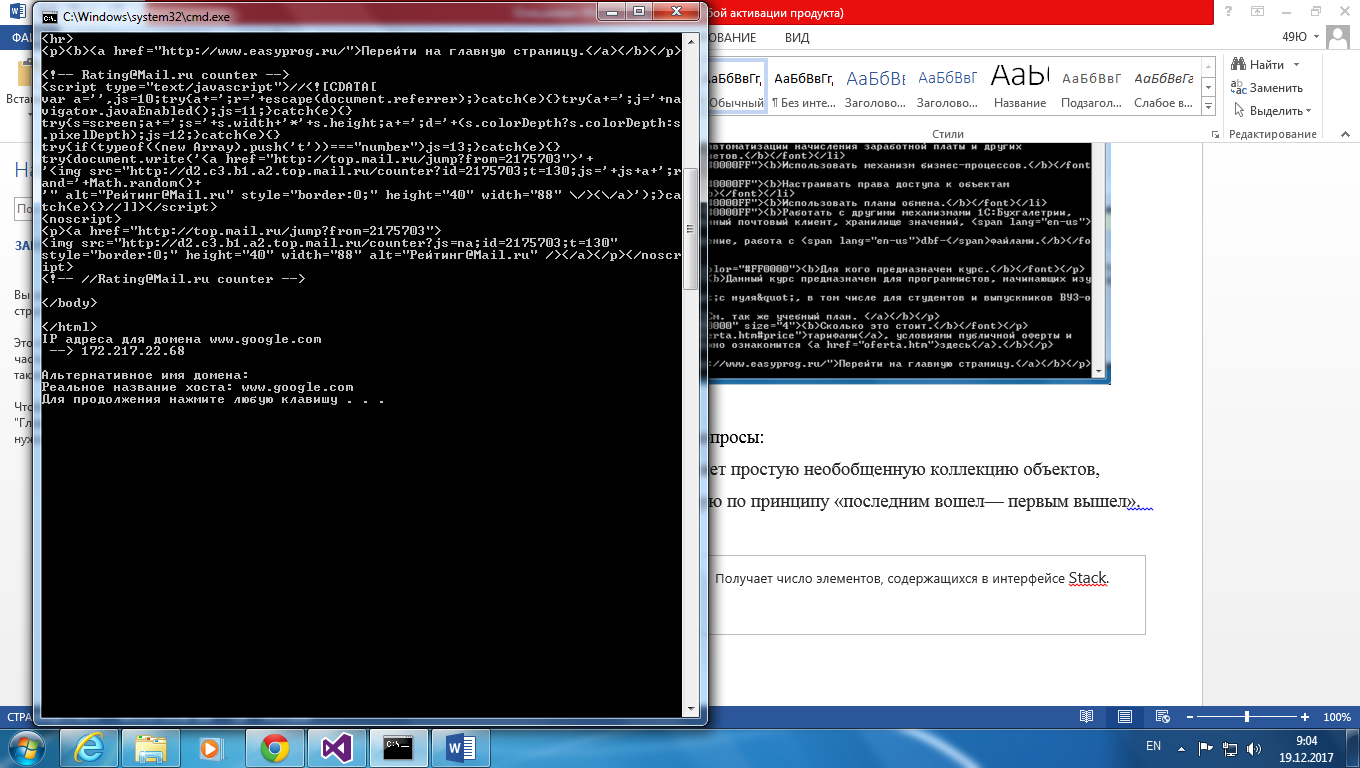


Рисунок 3 – Третяя часть выполнения программы

Алгоритм: созадние объекта класса «WebClient», класса позволяющего получать данных из сети с помощью URL, в объект «stream» записываем данные полученные с помощью объекта «client», преобразуем полученные байты в текстовый формат, вывод текста в консоль посточно. Создание объекта «entry» класса «IPHostEntry», передаем конструктору класса в качестве параметара строку содержащую название хоста, запись всех доступных IP-адресов хоста и псевдонимов хоста в строку. Вывод полученной строки в MessageBox.