Содержание

[Введение 5](#_Toc469834242)

[1 Теоритическая часть 6](#_Toc469834243)

[1.1 История создания C# 6](#_Toc469834244)

[1.2 Особенности С# 8](#_Toc469834245)

[2 Практическая часть 9](#_Toc469834246)

[2.1 Текст программы 9](#_Toc469834247)

[2.2 Алгоритм работы программы 21](#_Toc469834248)

[2.3 Инструкция пользователя 22](#_Toc469834249)

[2.4 Тестовый пример 23](#_Toc469834250)

[Заключение 25](#_Toc469834251)

[Список литературы 26](#_Toc469834252)

# Введение

Математика играет важнейшую роль в жизни человека. Компьютеры, современные телефоны и прочая техника сопровождают нас каждый день, а их создание невозможно без использования законов и расчетов данной науки.

В детстве мы учимся считать, выполнять простейшие арифметические действия с небольшими цифрами. Все это дети познают в виде игровых упражнения, да и самим это интересно.

Математика в детском саду изучают наглядно-действенным способом. Новые знания ребенок усваивает на основе непосредственного восприятия, когда следит за действием педагога, слушает его пояснения и указания.

Изучение математики в начальной школе направлена на представление о числах, величинах, геометрических фигурах, изучение математических языков, алгоритмов.

В средней школе, начинаются изучаться отдельные предметы, алгебра и геометрия. Возрастает уровень сложности заданий, количество формул и теоритического материала.

Математика следует за человеком везде, помогает ему решать задачи, делает его жизнь намного удобнее. Она затрагивает огромное количество областей. Некоторые профессии без нее не имеют смысла, многие появились только благодаря развитию отдельных ее направлений.

Чтобы упростить решение заданий и «освежить» память была создана программа помогающая учащимся в математике. Программа поможет учащимся в решении квадратных уравнений, сложение дробей, вычисления площади фигур через координаты или стороны, напомнит формулы сокращенного умножения.

# 1 Теоритическая часть

## 1.1 История создания C#

Язык появился на свет в июне 2000 г. в результате кропотливой работы большой группы разработчиков компании Microsoft, возглавляемой Андерсом Хейлсбергом. Этот человек известен как автор одного из первых компилируемых языков программирования для персональных компьютеров IBM — Turbo Pascal. Наверное, на территории бывшего Советского Союза многие разработчики со стажем, да и просто люди, обучавшиеся в той или иной форме программированию в вузах, испытали на себе очарование и удобство использования этого продукта. Кроме того, во время работы в корпорации Borland Андерс Хейлсберг прославился созданием интегрированной среды Delphi (он руководил этим проектом вплоть до выхода версии 4.0).

С# и .NET являются той самой отправной точкой. Если говорить упрощенно, то .NET представляет собой новую платформу, новый API для программирования в Windows, а С# ее новый язык, созданный с нуля, для работы с этой платформой, а также для извлечения всех выгод из прогресса сред разработки и нашего понимания принципов объектно-ориентированного программирования в течение последних 20 лет.

Необходимо отметить, что обратная совместимость не потеряна. Существующие программы будут выполняться, а платформа .NET была спроектирована таким образом, чтобы она могла работать с имеющимся программным обеспечением. Связь между компонентами в Windows сейчас почти целиком осуществляется при помощи СОМ. С учетом этого .NET обладает способностью создавать оболочки (wrappers) вокруг существующих компонентов СОМ, так что компоненты .NET могут общаться с ними, и создавать оболочки вокруг компонентов .NET, что позволяет им выглядеть как обычные СОМ-компоненты.

Авторы C# стремились создать язык, сочетающий простоту и выразительность современных объектно-ориентированных языков (вроде Java) c богатством возможностей и мощью C++. По словам Андерса Хейлсберга, C# позаимствовал большинство своих синтаксических конструкций из C++. В частности, в нем присутствуют такие удобные типы данных, как структуры и перечисления (другой потомок C++ — Java лишен этих элементов, что создает определенные неудобства при программировании). Синтаксические конструкции С# унаследованы не только от C++, но и от Visual Basic. Например, в С#, как и в Visual Basic, используются свойства классов. Как и C++ позволяет производить перегрузку операторов для созданных вами типов, Java не поддерживает ни ту, ни другую возможность. С# — это фактически гибрид разных языков, при этом синтаксически не менее (если не более) чист чем Java, так же прост как Visual Basic, и обладает практически той же мощью и гибкостью, что и C++.

## 1.2 Особенности С#

1. Полная поддержка классов и объектно-ориентированного программирования, включая наследование интерфейсов и реализаций, виртуальных функций и перегрузки операторов.
2. Полный и хорошо определенный набор основных типов.
3. Встроенная поддержка автоматической генерации XML-документации.
4. Автоматическое освобождение динамически распределенной памяти.
5. Возможность отметки классов и методов атрибутами, определяемыми пользователем. Это может быть полезно при документировании и способно воздействовать на процесс компиляции.
6. Полный доступ к библиотеке базовых классов .NET, а также легкий доступ к Windows API.
7. Указатели и прямой доступ к памяти, если они необходимы. Однако язык разработан таким образом, что практически во всех случаях можно обойтись и без этого.
8. Поддержка свойств и событий в стиле VB.
9. Простое изменение ключей компиляции. Позволяет получать исполняемые файлы или библиотеки компонентов .NET, которые могут быть вызваны другим кодом так же, как элементы управления ActiveX.
10. Возможность использования С# для написания динамических web-страниц ASP.NET.

# 2 Практическая часть

## 2.1 Текст программы

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics// пространство имен

{//открывающая операторная скобка

class Program//объявление класса

{//открывающая операторная скобка

static void Main(string[] args) //статический главный метод

{//открывающая операторная скобка

int a=0; //описание переменной

Console.WriteLine(" Введите номер операции");//оператор вывода текста

Console.WriteLine("1. Калькулятор");// оператор вывода текста

Console.WriteLine("2. Сложение дроби");// оператор вывода текста

Console.WriteLine("3. Решение квадратного уравнения");// оператор вывода текста

Console.WriteLine("4. Площадь фигур");// оператор вывода текста

Console.WriteLine("5. Площадь фигур через координаты");// оператор вывода текста

Console.WriteLine("6. Формулы сокращённого умнажения");// оператор вывода текста

a = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

if (a==1) //условный оператор

{//открывающая операторная скобка

calculator c = new calculator();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (a == 2) // условный оператор

{//открывающая операторная скобка

drob tr = new drob();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (a == 3) // условный оператор

{//открывающая операторная скобка

equation r = new equation();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (a == 6) // условный оператор

{//открывающая операторная скобка

Console.WriteLine("1. Квадрат суммы: (a+b)^2=a^2+2ab+b^2");// оператор вывода текста

Console.WriteLine("2. Квадрат разности: (a-b)^2=a^2-2ab+b^2");// оператор вывода текста

Console.WriteLine("3. Куб суммы: (a+b)^3=a^3+3a^2b+3ab^2+b^3");// оператор вывода текста

Console.WriteLine("4. Куб разности: (a-b)^3=a^3-3a^2b+3ab^2-b^3");// оператор вывода текста

Console.WriteLine("5. Разность квадратов: a^2-b^2=(a-b)(a+b)"); // оператор вывода текста

Console.WriteLine("6. Сумма кубов: a^3+b^3=(a+b)(a^2-ab+b^2)"); // оператор вывода текста

Console.WriteLine("7. Разность кубов: a^3-b^3=(a-b)(a^2+ab+b^2)"); // оператор вывода текста

}//закрывающая операторная скобка

if (a > 6 || a < 1) // условный оператор

{//открывающая операторная скобка

Console.WriteLine("Error"); ; //оператор вывода текста

}//закрывающая операторная скобка

if (a == 4) // условный оператор

{//открывающая операторная скобка

int b = 0; //описание переменной

Console.WriteLine("Введите фигуру");// оператор вывода текста

Console.WriteLine("1. Треугольник");// оператор вывода текста

Console.WriteLine("2. Трапеция");// оператор вывода текста

Console.WriteLine("3. Прямоугольник");// оператор вывода текста

Console.WriteLine("4. Параллелограм");// оператор вывода текста

b = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

if (b == 1) //условный оператор

{//открывающая операторная скобка

Triangle1 c = new Triangle1();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 2) // условный оператор

{//открывающая операторная скобка

trapeze1 tr = new trapeze1();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 3) // условный оператор

{//открывающая операторная скобка

rectangle1 r = new rectangle1();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 4) // условный оператор

{//открывающая операторная скобка

Parallelogram1 r = new Parallelogram1();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b > 4 || b < 1) // условный оператор

{//открывающая операторная скобка

Console.WriteLine("Error"); ; // оператор вывода текста

}//закрывающая операторная скобка

}//закрывающая операторная скобка

if (a==5) // условный оператор

{//открывающая операторная скобка

int b=0; // описание переменной

Console.WriteLine("Введите фигуру");// оператор вывода текста

Console.WriteLine("1. Треугольник");// оператор вывода текста

Console.WriteLine("2. Трапеция");// оператор вывода текста

Console.WriteLine("3. Прямоугольник");// оператор вывода текста

Console.WriteLine("4. Параллелограм");// оператор вывода текста

b = Convert.ToInt32(Console.ReadLine());//оператор вывода текста

if (b == 1) //условный оператор

{//открывающая операторная скобка

Triangle c = new Triangle();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 2) // условный оператор

{//открывающая операторная скобка

trapeze tr = new trapeze();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 3) // условный оператор

{//открывающая операторная скобка

rectangle r = new rectangle();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b == 4) // условный оператор

{//открывающая операторная скобка

parallelogram r = new parallelogram();//объявление переменной и выделение места

}//закрывающая операторная скобка

if (b > 4 || b < 1) // условный оператор

{//открывающая операторная скобка

Console.WriteLine("Error"); ; //оператор вывода текста

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Console.ReadKey();//оператор останавливает программу и ждет нажатия клавиши

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения длины вектора

using System; //подключение директивы

using System.Collections.Generic; //подключение директивы

using System.Linq; //подключение директивы

using System.Text; //подключение директивы

using System.Threading.Tasks; //подключение директивы

namespace Mathematics// пространство имен

{//открывающая операторная скобка

class vector//объявление класса

{//открывающая операторная скобка

private double ax; private double ay; private double bx; private double by; //описание приватных перемен

public vector ()// объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите координаты точки A");//оператор вывода текста

this.ax = Convert.ToDouble(Console.ReadLine());//ввод значение переменной

this.ay = Convert.ToDouble(Console.ReadLine());//ввод значение переменной

Console.WriteLine("Введите координаты точки B");// оператор вывода текста

this.bx = Convert.ToDouble(Console.ReadLine());//ввод значение переменной

this.by = Convert.ToDouble(Console.ReadLine());//ввод значение переменной

}//закрывающая операторная скобка

public double dlina(double D = 0) //объявление метода

{//открывающая операторная скобка

D = Math.Sqrt(Math.Pow((bx - ax), 2) + Math.Pow((by - ay), 2)); //нахождение длины вектора

Console.WriteLine("P={0}", D); // оператор вывода текста

return D; //возращение значения

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс калькулятор

using System;// подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class calculator//объявление класса

{//открывающая операторная скобка

private double a; private double b; private double v; private double s; private double y; private double d; private int x; //объявление приватных переменных

public calculator ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Выберите операцию");//оператор вывода

Console.WriteLine("1.Сложение");// оператор вывода

Console.WriteLine("2.Вычитание");// оператор вывода

Console.WriteLine("3.Умножение");// оператор вывода

Console.WriteLine("4.Деление");// оператор вывода

x = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Выберите первое число");// оператор вывода

a = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Выберите второе число");// оператор вывода

b = Convert.ToInt32(Console.ReadLine());//ввод значения переменно

switch (x) //условный оператор

{//открывающая операторная скобка

case 1: Console.WriteLine("Сумма={0}", a + b); break; //команды

case 2: Console.WriteLine("Разность={0}", a - b); break; //команды

case 3: Console.WriteLine("Произведение={0}", a \* b); break; //команды

case 4: Console.WriteLine("Разность={0}", a / b); break; //команды

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождение суммы дробей

using System;// подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class drob//объявление класса

{//открывающая операторная скобка

private double a1; private double a2; private double b1; private double b2; private double c1; private double c2; //объявление приватных переменных

public drob ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите числитель первой дроби");//оператор вывода

this.a1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите знаменатель первой дроби");// оператор вывода

this.a2 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите числитель второй дроби");// оператор вывода

this.b1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите знаменатель второй дроби");// оператор вывода

this.b2 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

if (this.a2==this.b2)//условный оператор {//открывающая операторная скобка

this.c2=a2; //присваивание

c1=a1+b1; //нахождения значение

}//закрывающая операторная скобка

else//иначе

{//открывающая операторная скобка

c2=a2\*b2; //нахождение значения

c1=a1\*b2+a2\*a1; //нахождение значения

}//закрывающая операторная скобка

Console.WriteLine("Сумма = {0}//{1}", c1, c2); //оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс решения квадратного уравнения

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class equation//объявление класса

{//открывающая операторная скобка

private double a; private double b; private double c; private double D; private double x1; private double x2; //объявление приватных переменных

public equation ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите а");//оператор вывода

this.a = Convert.ToInt32(Console.ReadLine());//ввод значения переменно

Console.WriteLine("Введите b");// ввод значения переменно

this.b = Convert.ToInt32(Console.ReadLine());//оператор вывода

Console.WriteLine("Введите c");// ввод значения переменно

this.c = Convert.ToInt32(Console.ReadLine());//оператор вывода

D = b \* b + 4 \* a \* c; //нахождение значения переменной

D=Math.Sqrt(D); //нахождение значения переменной

x1 = (-b + D) / (2 \* a); // нахождение значения переменной

x2 = (-b - D) / (2 \* a); // нахождение значения переменной

Console.WriteLine("Корни уровнения X={0},X={1}",x1,x2); // оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади параллелограмма через координаты вершин

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class parallelogram//объявление класса

{//открывающая операторная скобка

private double Ax1; private double Ay1; private double Cx1; private double Cy1; private double Bx1; private double By1; private double Dx1; private double Dy1; private double a; private double b; //объявление приватных переменных

public parallelogram()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите координаты точки А параллелограмма");//оператор вывода

this.Ax1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Ay1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки B параллелограмма");// оператор вывода

this.Bx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.By1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки C параллелограмма");// оператор вывода

this.Cx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Cy1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки D параллелограмма");// оператор вывода

this.Dx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Dy1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

}//закрывающая операторная скобка

public double perimeter(double P = 0) //объявление класса

{//открывающая операторная скобка

this.a = Math.Abs(Cx1 - Ax1); //нахождение значения переменной

this.b = Math.Abs(Ay1 - Cy1); //нахождение значения переменной

P = 2 \* (a + b); //нахождение значения переменной

Console.WriteLine("P={0}", P); //оператор вывода

return P; //возращение значения

}//закрывающая операторная скобка

public double area(double S = 0, double c = 0, double St = 0, double p = 0) //объявление метода

{//открывающая операторная скобка

c = Math.Sqrt(Math.Pow((Cx1 - Ax1), 2) + Math.Pow((Cy1 - Ay1), 2)); //нахождение значения переменной

p = (a + b + c) / 2; // нахождение значения переменной

St = Math.Sqrt(p \* (p - a) \* (p - c) \* (p - b)); // нахождение значения переменной

S = St \* 2; // нахождение значения переменной

Console.WriteLine("S={0}", S); //оператор вывода

return S; //возращение значения

}//закрывающая операторная скобка

public void rhombus(bool sq = false) //объявление метода

{//открывающая операторная скобка

if (a == b) //условный оператор

sq = true; //присваивание значение переменно

else sq = false; // присваивание значение переменно

Console.WriteLine("Ромб - {0}", sq); //оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождение площади параллелограмма

using System;//подключение директива

using System.Collections.Generic; // подключение директива

using System.Linq; // подключение директива

using System.Text; // подключение директива

using System.Threading.Tasks; // подключение директива

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class Parallelogram1//объявление класса

{//открывающая операторная скобка

private double a; private double h; private int v; double b; private double al; //объявление приватных переменных

public Parallelogram1 ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Выберите как искать площадь");//оператор вывода

Console.WriteLine("1.Через основание и высоту");//оператор вывода

Console.WriteLine("2.Через две стороны и угол между ними");//оператор вывода

this.v = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

}//закрывающая операторная скобка

public void are()//объявление метода

{//открывающая операторная скобка

if (v==1) // условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите основание");//оператор вывода

this.a=Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите высоту");//оператор вывода

this.h=Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S="+(a\*h)); //оператор вывода

}//закрывающая операторная скобка

if (v == 2) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите первую сторону");//оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите вторую сторону");//оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите угол");//оператор вывода

this.al = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (a \* b \* Math.Sin(al))); //оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади прямоугольника через координаты вершин

using System;// подключение директива

using System.Collections.Generic; // подключение директива

using System.Linq; // подключение директива

using System.Text; // подключение директива

using System.Threading.Tasks; // подключение директива

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class rectangle//объявление класса

{//открывающая операторная скобка

private double Ax1; private double Ay1; private double Cx1; private double Cy1; private double a; private double b; //объявление приватных переменных

public rectangle()//объявлений конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите координаты точки А прямоугольника");//оператор вывода

this.Ax1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Ay1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки C прямоугольника");// оператор вывода

this.Cx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Cx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

}//закрывающая операторная скобка

public double perimeter(double P = 0) //объявление метода

{//открывающая операторная скобка

a = Math.Abs(Cx1 - Ax1); //нахождение значения

b = Math.Abs(Ay1 - Cy1); // нахождение значения

P = 2 \* (a + b); // нахождение значения

Console.WriteLine("P={0}", P); // оператор вывода

return P; //возращение значения

}//закрывающая операторная скобка

public double area(double S = 0) //объявление метода

{//открывающая операторная скобка

S = a \* b; // нахождение значения

Console.WriteLine("S={0}", S); // оператор вывода

return S; //возращение значения

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади прямоугольника

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class rectangle1//

{//открывающая операторная скобка

private double a; private double b; //объявление приватных переменных

public rectangle1 ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите сторону а");//оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите сторону b");//оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (a \* b)); //оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади трапеции через координаты вершин

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class trapeze//объявление класса

{//открывающая операторная скобка

private double Ax1; private double Ay1; private double Cx1; private double Cy1; private double Bx1; private double By1; private double Dx1; private double Dy1; private double a; private double b; private double c; private double d; //объявление приватных переменных

public trapeze()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите координаты точки А трапеции");//оператор вывода

this.Ax1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Ay1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки B трапеции");// оператор вывода

this.Bx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.By1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки C трапеции");// оператор вывода

this.Cx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Cy1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки D трапеции");// оператор вывода

this.Dx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Dy1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

}//закрывающая операторная скобка

public double perimeter(double P = 0) //объявление метода

{//открывающая операторная скобка

this.a = Math.Sqrt(Math.Pow((Bx1 - Ax1), 2) + Math.Pow((By1 - Ay1), 2)); //нахождение значения переменной

this.b = Math.Sqrt(Math.Pow((Cx1 - Bx1), 2) + Math.Pow((Cy1 - By1), 2)); // нахождение значения переменной

this.c = Math.Sqrt(Math.Pow((Cx1 - Dx1), 2) + Math.Pow((Cy1 - Dy1), 2)); // нахождение значения переменной

this.d = Math.Sqrt(Math.Pow((Dx1 - Ax1), 2) + Math.Pow((Dy1 - Ay1), 2)); // нахождение значения переменной

P = a + b + c + d; // нахождение значения переменной

Console.WriteLine("P={0}", P); //оператор вывода

return P; //возращение значения

}//закрывающая операторная скобка

public double area(double S = 0, double St = 0, double p = 0, double ac = 0, double St2 = 0, double p2 = 0) //объявление метода

{//открывающая операторная скобка

ac = Math.Sqrt(Math.Pow((Cx1 - Ax1), 2) + Math.Pow((Cy1 - Ay1), 2)); // нахождение значения переменной

p = (a + b + ac) / 2; // нахождение значения переменной

St = Math.Sqrt(p \* (p - ac) \* (p - a) \* (p - b)); // нахождение значения переменной

p2 = (d + c + ac) / 2; // нахождение значения переменной

St2 = Math.Sqrt(p \* (p - ac) \* (p - d) \* (p - c)); // нахождение значения переменной

S = St + St2; // нахождение значения переменной

Console.WriteLine("S={0}", S); //оператор вывода

return S; //возращение значения

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади трапеции

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class trapeze1//объявление класса

{//открывающая операторная скобка

private double a; private double h; private int v; double b;private double al;private double c; private double d;private double m; //объявление приватных переменных

public trapeze1 ()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Выберите как искать площадь");//оператор вывода

Console.WriteLine("1.Через основания и высоту");// оператор вывода

Console.WriteLine("2.Через диагонали и угол между ними");// оператор вывода

Console.WriteLine("3.Через четыре стороны");// оператор вывода

this.v = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

if (v == 1) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите нижнее основание");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите верхнее основание//"); оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите высоту");// оператор вывода

this.h = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (((a + b) / 2) \* h)); // оператор вывода

}//закрывающая операторная скобка

if (v == 2) //условный оператор

{

Console.WriteLine("Введите первую диоганаль");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите вторую диоганаль");// оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите угол между диагоналями");// оператор вывода

this.al = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (((a \* b) / 2) \* Math.Sin(al))); // оператор вывода

}//закрывающая операторная скобка

if (v == 3) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите нижнее основание");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите верхнее основание");// оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите первую сторону");// оператор вывода

this.c = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите вторую сторону");// оператор вывода

this.d = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

m = (((a - b) \* (a - b) + c \* c - d \* d)) / (2 \* (a - b)); //нахождение значения переменной

m = Math.Pow(m, 2); //нахождение значения переменной

Console.WriteLine("S=" + (((a + b) / 2) \* Math.Sqrt(c \* c - m))); // оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождение площади треугольника через координаты вершин

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

public class Triangle//объявление класса

{//открывающая операторная скобка

private double Ax1; private double Ay1; private double Cx1; private double Cy1; private double Bx1; private double By1; private double a1; private double b1; private double c1; //объявление приватных переменных

public Triangle()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Введите координаты точки А треугольника");//оператор вывода

this.Ax1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Ay1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки B треугольника");// оператор вывода

this.Bx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.By1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите координаты точки C треугольника");// оператор вывода

this.Cx1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.Cy1 = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

}//закрывающая операторная скобка

public double perimeter(double P = 0) //объявление метода

{//открывающая операторная скобка

this.a1 = Math.Sqrt(Math.Pow((Bx1 - Ax1), 2) + Math.Pow((By1 - Ay1), 2)); //нахождение значения переменной

this.b1 = Math.Sqrt(Math.Pow((Cx1 - Bx1), 2) + Math.Pow((Cy1 - By1), 2)); // нахождение значения переменной

this.c1 = Math.Sqrt(Math.Pow((Cx1 - Ax1), 2) + Math.Pow((Cy1 - Ay1), 2)); // нахождение значения переменной

P = a1 + b1 + c1; // нахождение значения переменной

Console.WriteLine("P={0}", P); // оператор вывода

return P; //возращение значения

}//закрывающая операторная скобка

public double area(double S = 0, double p = 0) //объявление метода

{//открывающая операторная скобка

p = (a1 + b1 + c1) / 2; // нахождение значения переменной

S = Math.Sqrt(p \* (p - c1) \* (p - b1) \* (p - a1)); // нахождение значения переменной

Console.WriteLine("S={0}", S); // оператор вывода

return S; //возращение значения

}//закрывающая операторная скобка

public void type(bool t1 = false, bool t2 = false, bool t3 = false) //объявление метода

{//открывающая операторная скобка

if (a1 != b1 && b1 != c1 && c1 != a1) //условный оператор

t1 = true; //присваивание значения переменной

else t1 = false; // присваивание значения переменной

if (a1 == b1 && b1 == c1 && c1 == a1) // условный оператор

t3 = true; // присваивание значения переменной

else t3 = false; // присваивание значения переменной

if ((a1 == b1 || b1 == c1 || c1 == a1) && (a1 != b1 || b1 != c1 || c1 != a1)) // условный оператор

t2 = true; // присваивание значения переменной

else t2 = false; // присваивание значения переменной

Console.WriteLine("Треугольник разносторонний - {0}", t1); // оператор вывода

Console.WriteLine("Треугольник равнобедренный - {0}", t2); // оператор вывода

Console.WriteLine("Треугольник равносторонний - {0}", t3); // оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

Класс нахождения площади треугольника

using System;//подключение директивы

using System.Collections.Generic; // подключение директивы

using System.Linq; // подключение директивы

using System.Text; // подключение директивы

using System.Threading.Tasks; // подключение директивы

namespace Mathematics//пространство имен

{//открывающая операторная скобка

class Triangle1//объявление класса

{//открывающая операторная скобка

private int v; private double a; private double b; private double h; private double al; private double c; private double p; //объявление приватных переменных

public Triangle1()//объявление конструктора

{//открывающая операторная скобка

Console.WriteLine("Выберите как искать площадь");//оператор вывода

Console.WriteLine("1.Через сторону и высоту, проведенную к ней");// оператор вывода

Console.WriteLine("2.Через две стороны и угол между ними");// оператор вывода

Console.WriteLine("3.Через формулу Герона");// оператор вывода

this.v = Convert.ToInt32(Console.ReadLine());//ввод значения переменной

if (v == 1) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите сторону");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите высоту");// оператор вывода

this.h = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (0.5 \* a \* h)); // оператор вывода

}//закрывающая операторная скобка

if (v == 2) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите первую сторону");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите вторую сторону");// оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите угол между ними");// оператор вывода

this.al = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("S=" + (0.5\*a\*b\* Math.Sin(al))); // оператор вывода

}//закрывающая операторная скобка

if (v == 3) //условный оператор

{//открывающая операторная скобка

Console.WriteLine("Введите первую сторону");// оператор вывода

this.a = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите вторую сторону");// оператор вывода

this.b = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

Console.WriteLine("Введите третью сторону");// оператор вывода

this.c = Convert.ToDouble(Console.ReadLine());//ввод значения переменной

this.p = (a+b+c)/2; //нахождение значения переменной

Console.WriteLine("S=" + (Math.Sqrt(p - a) \* (p - c) \* (p - b))); // оператор вывода

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

}//закрывающая операторная скобка

## 2.2 Алгоритм работы программы

В программе осуществлены 13 классов:

1. Program.cs
2. calculator.cs
3. drob.cs
4. equation.cs
5. parallelogram.cs
6. Parallelogram1.cs
7. rectangle.cs
8. rectangle1.cs
9. trapeze.cs
10. trapeze1.cs
11. Triangle.cs
12. Triangle1.cs
13. vector.cs

Каждый класс отвечает за определенную задачу, например, класс parallelogram.cs отвечает за нахождение площади параллелограмма через координаты вершин.

После запуска программы, подключаются библиотеки для работы с функциями языка. Далее определяется единое пространство для всех файлов проекта, далее запускается класс, с главным методом. После запускается оператор вывода и оператор ввода, который ждет введение значения. После того как пользователь ввел значение операции программа запускает соответствующий класс и выполняются определенные действия в зависимости от класса.

## 2.3 Инструкция пользователя

1. Запустите программный продукт.
2. На экране появиться список операция, выбираем один из них.
3. После выбора операции следуем указанием на экране компьютера.
4. После окончания работы с программой, нажать любую клавишу или нажать кнопку закрыть.

## 2.4 Тестовый пример

Для проверки работоспособности консольного программного продукта был выполнен ряд тестов, которые помогли выявить ошибки и исправить их.

После запуска программы появляется консольное окно, в котором надо ввести значение требуемой операции, всего предлагается на выбор 6 операций. Для примера была выбрана 4-ая операция. (Рис.1)

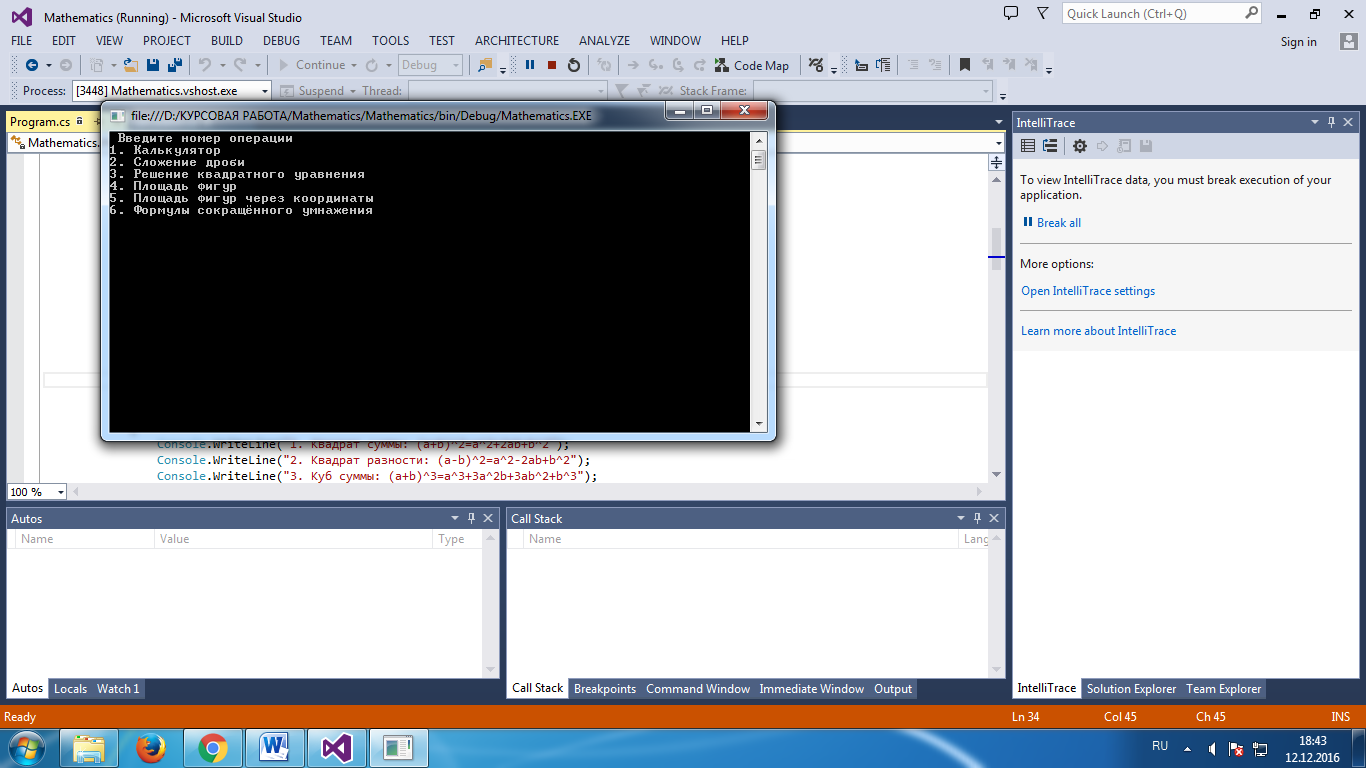


Рис.1

После того как была выбрана 4-ая операция, пользователю предложили выбрать одну из фигур, треугольник, трапеция, прямоугольник или параллелограмм. Был выбран прямоугольник, 3 фигура. (Рис.2)

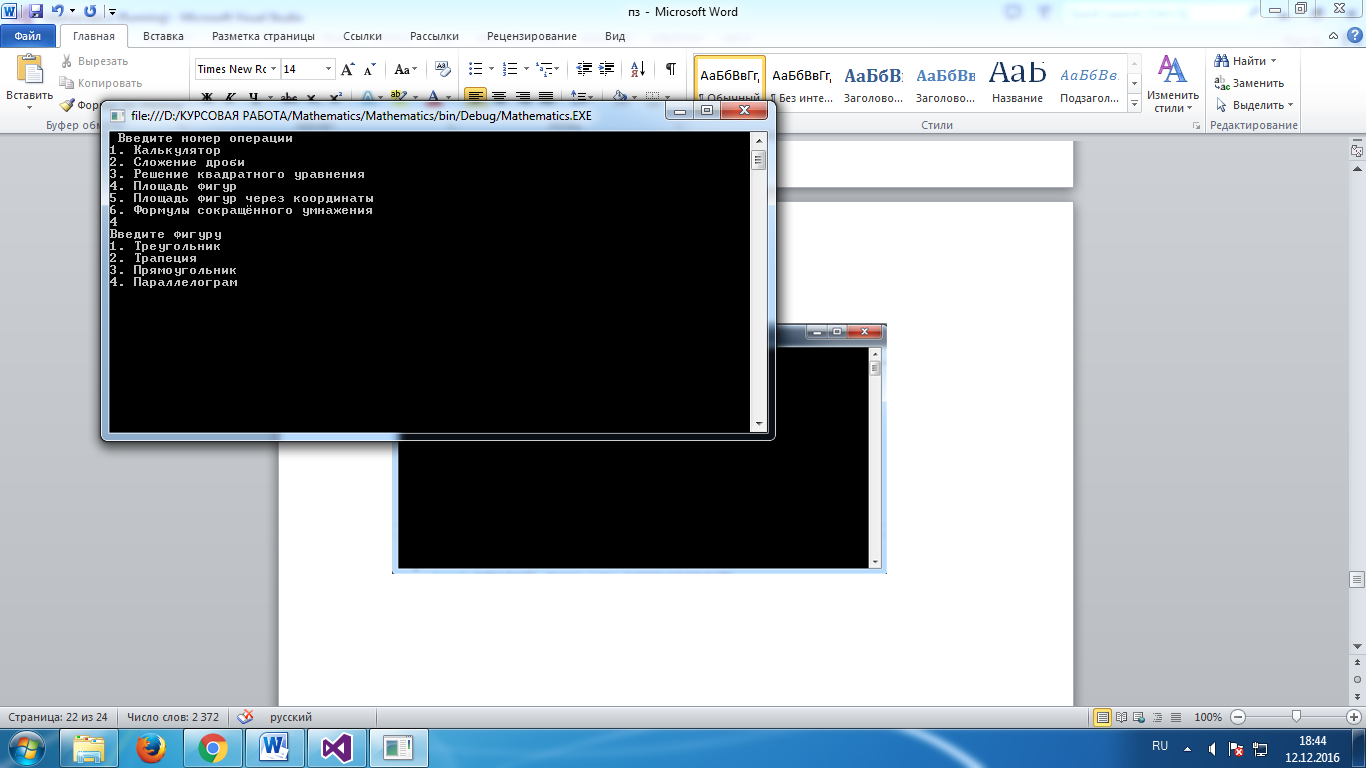


Рис.2

После выбора фигуры, пользователю, надо ввести сторону а, было введено значение 5. Далее надо ввести сторону b, было введено значение 7. После ввода значений программа показывает результат S=35. Как мы знаем, формулой нахождения площади прямоугольника является S=a\*b, действительно, если мы подставим введенные значения и посчитаем, то площадь прямоугольника будет равна 35. (Рис.3)

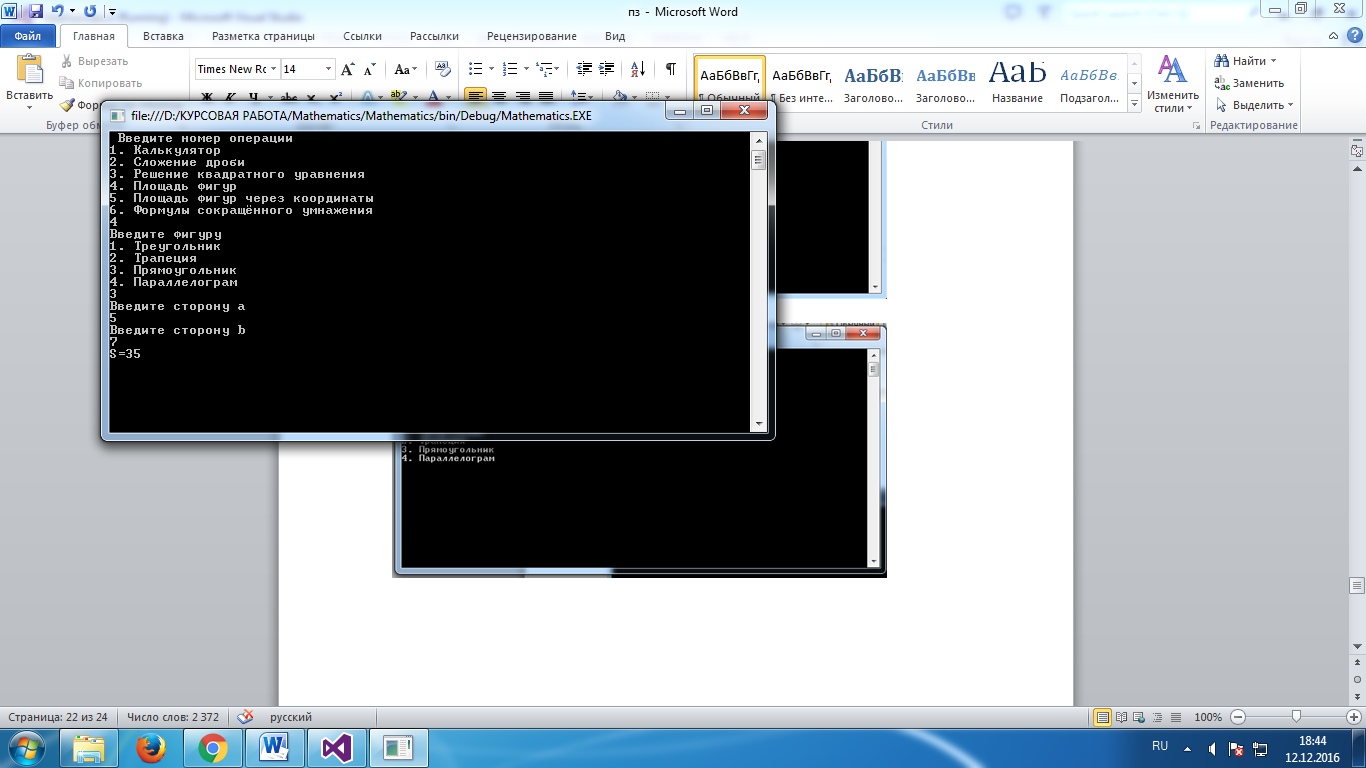


Рис.3

После выполнения ряда тестов, и исправления ошибок которые они выявили, можно с уверенностью сказать, что программа работоспособная и полностью готова к использованию.

# Заключение

В ходе работы над курсовым проектом был написан консольный программный продукт выполняющие математические расчёты.

Программный продукт может выполнять такие операции как:

1. Сложение, вычитание, умножение, деление
2. Сложение дробей
3. Решение квадратного уравнения
4. Нахождение площади фигур (Треугольник, трапеция, прямоугольник, параллелограмм)
5. Нахождение площади фигур через координаты вершин (Треугольник, трапеция, прямоугольник, параллелограмм)

Также в программе представлена теоритическая часть, а именно формулы сокращенного умножения.

В будущем планируется добавить следующие математические операции:

1. Калькулятор дробей
2. Возведение в степень
3. Извлечение квадратного корня
4. Синус, косинус, тангенс, котангенс

И следующую теоритическую информацию:

1. Формулы привидения
2. Тригонометрические функции
3. Решение квадратного уравнения по формуле Виета
4. Признаки равенства и подобия треугольников

Данный программный продукт подойдет для учащихся 3 – 11 классов, а также студентов начальных курсов университетов. Также программа подойдет для определенных профессий где надо находить площади фигур.

# Список литературы

1. Джозеф Албахари. C# 6.0. Справочник. Полное описание языка/ Джозеф Албахари, Бен Албахари – Москва : Вильямс, 2016 - 1040 с.
2. Зиборов В.В. Visual С# 2012 на примерах/ Зиборов В.В. - БХВ-Петербург, 2013 – 175 с.
3. Пол Дейтел. Как программировать на Visual C# 2012/ Пол Дейтел, Харви Дейтел. - Санкт-Петербург: Питер, 2014 – 858 с.
4. Нейгел К. C# 5.0 и платформа .NET 4.5 для профессионалов/ Нейгел К., Ивьен Б. - Москва: Вильямс, 2014- 1440 с.
5. Ишкова Э. А. Самоучитель C#. Начала программирования/ Ишкова Э.А. - Санкт-Петербург: Наука и Техника, 2013 – 496 с.
6. Джон Скит. C# для профессионалов. Тонкости программирования/ Джон Скит. – Москва: Вильмяс, 2014 – 608 с.
7. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5/ Эндрю Троелсен. – Москва: Вильмяс, 2013 – 1311 с.
8. Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#/ Джеффри Рихтер. - Санкт-Петербург: Питер, 2012 – 893 с.
9. mycsharp.ru
10. http://vbbook.ru/visual-c.net/