Министерство образования и науки Российской Федерации Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»**

Институт радиоэлектроники и информационных технологий – РтФ

Кафедра «Информационных технологий»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

Студенка группы:

РИ-240004

Патракова Екатерина Сергеевна

Преподаватель:

Быков Е.А.

Екатеринбург

2016

Содержание:

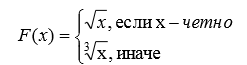
1. Введение3
2. Описание задания3
3. Описание библиотек, классов, основных методов5
   1. Использованные готовые классы5
   2. Созданные классы6
4. Результаты работы9
5. Анализ полученных данных12
6. Код программы13
7. Выводы23
8. **Введение:**

Целью данной лабораторной работы является ознакомление с основными возможностями языка программирования C# посредством написания простого консольного приложения для решения ряда логических (математических) задач.

В ходе выполнения работы подлежит решению следующая задача:

Написание функционального консольного приложения для решения нескольких задач из различных предметных областей и с привлечением различных средств и особенностей языка программирования.

1. **Описание задания:**
   1. Создать консольное приложение позволяющее принимать сообщение с консоли обрабатывать его, то есть подсчитывать количество символов и слов в сообщении и выводить всю информацию на консоли.
   2. Определение является ли заданный год високосным с помощью класса DateTime. Пользователь вводит год интересующий его, программа должна определить является ли она високосным (366 дней).
   3. Анализ генерации случайных чисел. Пользователь вводит 3 параметра: число определяющее сколько раз должно сгенерироваться число, минимальная граница генерации чисел и максимальная граница генерации чисел. После чего должно сгенерироваться n-ое количество чисел в заданных границах. Далее необходимо вычислить сколько раз было сгенерировано каждое из чисел и вывести эту информацию пользователю.
   4. Расчет разницы между заданной датой и текущей используя класс DateTime.
   5. Вычисление площади треугольника по заданным: стороне и высоте, трем сторонам, двум сторонам и углу между ними. Пользователь выбирает, какую именно хочет решать задачу сейчас, далее вводи необходимую информацию. Используя пакет System.Math.
   6. Дана последовательность натуральных чисел, разделенных пробелами, в виде строки. Необходимо вычислить их сумму.
   7. Вычислить значение функции при условии, что х - целое:



* 1. Вывести на экран сообщение в зависимости от введенного значения оценки (по десятибалльной системе), например: от 1 до 2 плохо, от 3 до 5 удовлетворительно, и т.д., иначе – неправильный ввод данных. Описание библиотек, классов и основных методов:
  2. Описать символьный массив D, размерностью 30 элементов, заполнять массив до тех пор, пока не встретится символ ?. Вывести все элементы в три строки.
  3. Найти максимальные элементы каждого столбца действительной матрицы К размерности NхN и поместить их на главную диагональ, а диагональные элементы – на место соответствующего элемента столбца.
  4. Описание: Напишите программу, которая вычисляет целочисленные арифметические выражения любой длины, не содержащие скобок. Учесть все арифметические операции: сложение, вычитание, умножение и целочисленное деление.

1. **Описание библиотек, классов и основных методов:**

#### 3.1 Использованные готовые классы

* Класс **Console** пространства имен System

Предоставляет стандартные потоки для консольных приложений: входной, выходной и поток сообщений об ошибках.

Метод WriteLine() для вывода текста на консоль.

Метод ReadLine() для считывания текста с консоли

Метод ReadKey() для посимвольного считывания.

Свойство KeyChar возвращает или задает символ, соответствующий нажатой клавише.

* Структура **Int** пространства имен System

Представляет целое число со знаком.

Метод Parse(String) для преобразования считанной с консоли входной строки к формату целого числа.

* Структура **Double** пространства имен System

Представляет число двойной точности с плавающей точкой.

Метод Parse(String) для преобразования считанной с консоли входной строки к формату числа с плавающей точкой.

ToString(double) для преобразования числа двойной точности с плавающей точкой к формату строки;

* Структура **DateTime** пространства имен System:

Представляет текущее время, обычно выраженное как дата и время суток.

Метод Parse(String) для преобразования считанной с консоли входной строки к формату даты.

Метод IsLeapYear(Int), позволяющий определить, является ли заданный год високосным.

Свойство Now, возвращающее текущие дату и время.

* Структура **TimeSpan** пространства имен System

Представляет интервал времени.

В программе создается экземпляр структуры для хранения интервала между двумя датами, выраженного в днях (свойство Days);

* Класс **Math** пространства имен System

Предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

Метод Sqrt(Double) для получения квадратного корня из числа с плавающей точкой.

Метод Sin(Double) для получения соответственно синуса заданного числа.

Метод Pow(Double, Double) для возведения заданного числа с плавающей точкой в заданную степень;

Поле PI со значением равным 3,14159265358979323846.

* Класс **Random** пространства имен System

Представляет генератор псевдослучайных чисел.

Создается экземпляр класса и используется метод Next(Int32, Int32) для получения последовательности псевдослучайных чисел в заданном диапазоне;

* Класс **String** пространства имен System

Представляет текст как последовательность знаков Юникода.

Метод Split(Char[], StringSplitOptions)для разбивания строки, включающей разделитель (пробел), на подстроки без включения в массив подстрок самих разделяющих символов.

Метод Replace(String, String) для замены одной подстроки в строке другой.

Метод Remove(Int) возвращает новую строку, в которой были удалены все символы, начиная с указанной позиции и до конца в текущем экземпляре.

Метод Remove(Int, Int) возвращает новую строку, в которой было удалено указанное число символов в указанной позиции.

Метод CompareTo(String) сравнивает данный экземпляр с заданным объектом String и показывает, расположен ли данный экземпляр перед, после или на той же позиции в порядке сортировки, что и заданная строка.

Свойство Length возвращает число знаков в текущем объекте String.

* Класс List <T> пространства имен [System.Collections.Generic](https://msdn.microsoft.com/ru-ru/library/system.collections.generic(v=vs.110).aspx)

Представляет строго типизированный список объектов, доступных по индексу. Поддерживает методы для поиска по списку, выполнения сортировки и других операций со списками.

Метод Add(T) добавляет объект в конец очереди.

#### 3.2. Созданные классы

1. Класс Program пространства имен ЛР1

Включает в себя cстатический метод Main(), являющийся точкой входа в программу и обеспечивающий диалог программы и пользователя.

Метод EXIT()обеспечивающий диалог о выходе или продолжении работы. Прототип: *public bool* EXIT();

1. Класс Line пространства имен ЛР1

Включает в себя закрытое поле для хранения исходной строки, конструктор по умолчанию и метод CountWordsAndLetters(), подсчитывающий количество слов и символов.

Прототип метода CountWordsAndLetters(): *public void* CountWordsAndLetters();

1. Класс Year пространства имен ЛР1

Включает в себя закрытое поле для хранения года (целое число), конструктор по-умолчанию и метод IsLeapYear(), выводящий на консоль информацию, является ли хранимый в виде поля год високосным.

Прототип метода IsLeapYear ():*public* *void* IsLeapYear ();

1. Класс RandomNumbers пространства имен ЛР1

Представляет собой генератор псевдослучайных чисел со встроенным анализатором. Включает в себя закрытые целые поля для количества чисел, которые необходимо сгенерировать, верхней и нижней границ генерации, и контейнера для хранения массива сгенерированных чисел. Класс так же содержит конструктор по умолчанию и методы AnalyzeRandomlyNumbers() для ввода исходных данных и GetAnalysGenerateNumbers() подсчитывающий, сколько раз встретилось каждое из чисел.

Прототип GetAnalysGenerateNumbers(): private void GetAnalysGenerateNumbers();

Прототип AnalyzeRandomlyNumbers(): public void AnalyzeRandomlyNumbers();

1. Класс Date пространства имен ЛР1

Представляет собой контейнер для хранения даты с возможностью вычисления разницы между двумя датами.

Включает в себя конструктор по умолчанию, два поля типа DateTime и метод ToDateTimeDiff(), вычисляющий разницу между текущей датой и введенной, и выводящий результат обработки на консоль.

Прототип метода ToDateTimeDiff(): public void ToDateTimeDiff();

1. Класс Triangle пространства имен ЛР1

Класс описывает треугольник и содержит все основные элементы фигуры. Включает в себя закрытые действительные поля для сторон треугольника и его высоты, целое поле для одного из его углов и поле для площади, пять методов. Главный метод CalculateTriangleSquare() предназначен для реализации меню выбора пути подсчета плозади треугольника. Для непосредственного вычисления площади различными способами: GetSquare\_HeightSide() – по заданным стороне и высоте, GetSquare\_ThreeSides() – по заданным трем сторонам, GetSquare\_SidesAngle() – по заданным двум сторонам и углу между ними. Метод IsTriangle() предназначен для проверки того, что введенные три стороны образуют треугольник. Вычисления проводятся по стандартным математическим формулам с использованием арсенала класса System.Math.

Прототип метода CalculateTriangleSquare(): *public* *void С*alculateTriangleSquare ();

Прототип метода GetSquare\_HeightSide(): *private* *void* GetSquare\_HeightSide ();

Прототип метода GetSquare\_ThreeSides(): *private* *void* GetSquare\_ThreeSides();

Прототип метода GetSquare\_SidesAngle(): *private* *void* GetSquare\_SidesAngle ();

Прототип метода IsTriangle(): *private* *bool* IsTriangle();

1. Класс NatureNumbers пространства имен ЛР1

Контейнер для хранения и обработки последовательности натуральных чисел, представленной в виде строки, разделенной пробелами. Включает в себя закрытое поле для строки чисел, для суммы, конструктор по умолчанию и метод SumNaturalNumbers(), вычисляющий сумму чисел.

Прототип метода SumNaturalNumbers(): *public void* SumNaturalNumbers() ;

1. Класс Function пространства имен ЛР1

Имеет закрытое действительное поле для хранения аргумента, конструктор по умолчанию и метод CalculateFunction(), вычисляющий значение определенной в задании функции от аргумента-поля класса. Вычисления проводятся с использованием арсенала класса System.Math.

Прототип метода CalculateFunction ():*public* *void* CalculateFunction ();

1. Класс Mark пространства имен ЛР1

Включает в себя закрытое действительное поле для хранения оценки, конструктор по умолчанию и метод GetMark(), выводящий интерпретацию оценки введенной по 10-ти бальной шкале.

Прототип метода GetMark():*public* *void* GetMark();

1. Класс Array пространства имен ЛР1

Включает в себя одно закрытое поле типа string, целое поле для размерности последовательности, конструктор по умолчанию, метод GetThreeLine() для вывода исходной строки в три строки.

Прототип метода GetThreeLine ():*public* *void* GetThreeLine();

1. Класс Matrix пространства имен ЛР1

Включает в себя одно закрытое поле типа «список списков действительных чисел» для хранения матрицы, целое поле для размерности матрицы, конструктор по умолчанию, метод Replace() для обмена наибольшего элемента в столбце и диагонального элемента.

Прототип метода Replace():*public* *void* Replace();

1. Класс Calculator пространства имен ЛР1

Для вычисления строкового математического выражения с множеством действий.

Включает в себя два поля массивы строк, один для хранения чисел, другой для хранения математических знаков, конструктор по умолчанию, метод InputExpression() для ввода исходной строки и её разбиения на массив строк, каждый элемент которого содержит либо число, либо математический знак. CalculateSmth() вызова методов MultiplAndDivision() и PlusAndMinus(), и вывода результата, Метод MultiplAndDivision() для вычисления всех действий с умножением и делением, метод PlusAndMinus() для суммирование и вычитания.

Прототип метода InputExpression (): *private* *void* InputExpression ();

Прототип метода MultiplAndDivision(): *private void* MultiplAndDivision();

Прототип метода PlusAndMinus(): *private void* PlusAndMinus ();

Прототип метода CalculateSmth ():*public* *void* CalculateSmth();

1. **Результаты работы программ:**
   1. Меню:

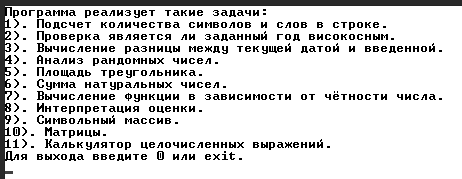


Рис. 1. Меню приложения

* 1. Количество символов и слов в строке:



Рис. 2. Нормальный ход программы

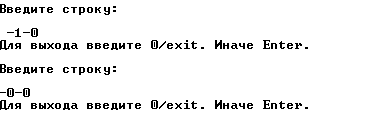


Рис. 3. Обработка пустой строки \ строки с пробелами



Рис. 4. Цифры и знаки препинания не считаются словом

* 1. Високосный ли год?

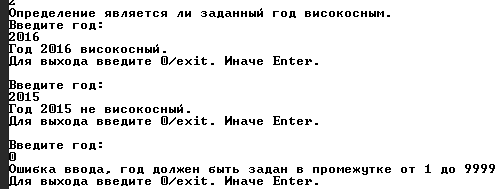


Рис. 5. Нормальный ход программы и обработка допустимого промежутка

* 1. Анализ рандомных чисел

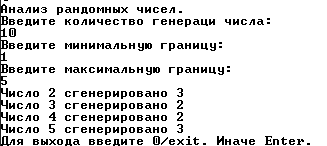


Рис. 6. Нормальный ход программы

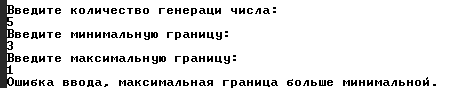


Рис. 7. Ошибка. Минимальная граница больше максимальной

* 1. Разница между введенной датой и сегодняшней

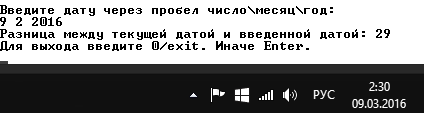


Рис. 8. Нормальный ход программы



Рис. 9. Разница между введенной сегодняшней датой и взятой в программе

* 1. Площадь треугольника:

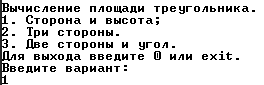


Рис. 10. Меню для выбора пути подсчета площади

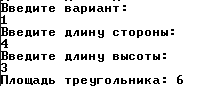


Рис. 11. Площадь через сторону и высоту

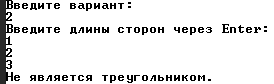


Рис. 12. Площадь через три стороны. Проверка.

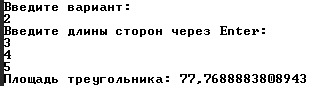


Рис. 13. Площадь через три стороны

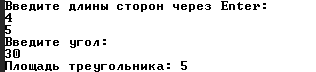


Рис. 14. Площадь через две стороны и угол

* 1. Сумма натуральных чисел:



Рис. 15.Нормальный ход программы



Рис. 16. Если введено только одно число. Проверка.

* 1. Вычисление функции:

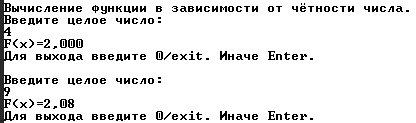


Рис. 17. Вычисление функции в зависимости от четности

* 1. Перевод десятибалльной оценки:



Рис. 18. Результат при вводе 2



Рис. 19. Результат при вводе 3



Рис. 20. Результат при вводе 6

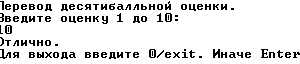


Рис. 21. Результат при вводе 10

* 1. Символьный массив:

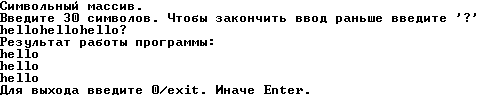


Рис. 22. Вывод строки в три

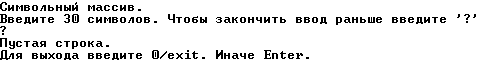


Рис. 23. Вывод при пустой строке

* 1. Матрица:

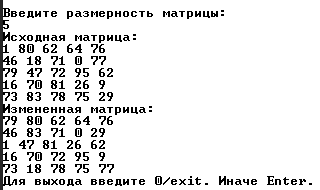


Рис. 24. Матрица 5х5

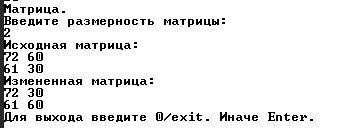


Рис. 25. Матрица 2х2

* 1. Калькулятор:

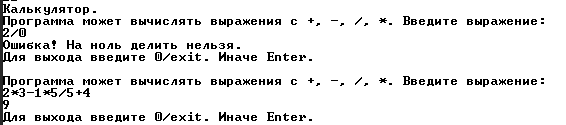


Рис. 26. Результат работы калькулятора

1. **Анализ полученных данных:**

В результате выполнения лабораторной работы было разработано многофункциональное консольное приложение, реализующее решение математических и логических задач, предлагаемых пользователю на выбор. Для осуществления диалога программы и пользователя в приложении реализовано главное меню, позволяющее выбрать одну из решаемых им подзадач для выполнения.

1. **Код программы:**
2. Функция Main():

using System;

namespace ЛР1

{

class Program

{

static void Main(string[] args)

{

string task;

while (true)

{

Console.Clear();

Console.WriteLine(

"Программа реализует такие задачи:\n1). Подсчет количества символов и слов в строке.\n2). Проверка является ли заданный год високосным." +

"\n3). Вычисление разницы между текущей датой и введенной." +

"\n4). Анализ рандомных чисел.\n5). Площадь треугольника. \n6). Сумма натуральных чисел.\n7). Вычисление функции в зависимости от чётности числа."

+

"\n8). Интерпретация оценки.\n9). Символьный массив.\n10). Матрицы.\n11). Калькулятор целочисленных выражений.\nДля выхода введите 0 или exit.");

task = Console.ReadLine();

switch (task)

{

case "0":

case "exit":

return;

case "1":

Console.WriteLine("Подсчет количества символов и слов в строке.");

while (true)

{

Line line = new Line();

line.CountWordsAndLetters();

if (EXIT())

break;

}

break;

case "2":

Console.WriteLine("Определение является ли заданный год високосным.");

while (true)

{

Year year = new Year();

year.IsLeapYear();

if (EXIT())

break;

}

break;

case "3":

Console.WriteLine("Вычисление разницы между текущей датой и введенной.");

while (true)

{

Date date = new Date();

date.ToDateTimeDiff();

if (EXIT())

break;

}

break;

case "4":

Console.WriteLine("Анализ рандомных чисел.");

RandomNumbers randNumbers = new RandomNumbers();

while (true)

{

randNumbers.AnalyzeRandomlyNumbers();

if (EXIT())

break;

}

break;

case "5":

Console.WriteLine("Вычисление площади треугольника.");

Triangle triangle = new Triangle();

triangle.CalculateTriangleSquare();

break;

case "6":

Console.WriteLine("Сумма натуральных чисел.");

while (true)

{

NaturalNumbers naturalNumbers = new NaturalNumbers();

naturalNumbers.SumNaturalNumbers();

if (EXIT())

break;

}

break;

case "7":

Console.WriteLine("Вычисление функции в зависимости от чётности числа.");

while (true)

{

Function function = new Function();

function.CalculateFunction();

if (EXIT())

break;

}

break;

case "8":

Console.WriteLine("Перевод десятибалльной оценки.");

while (true)

{

Mark mark = new Mark();

mark.GetMark();

if (EXIT())

break;

}

break;

case "9":

Console.WriteLine("Символьный массив.");

while (true)

{

Array array = new Array();

array.GetThreeLine();

if (EXIT())

break;

}

break;

case "10":

Console.WriteLine("Матрица.");

while (true)

{

Matrix matrix = new Matrix();

matrix.Replace();

if (EXIT())

break;

}

break;

case "11":

Console.WriteLine("Калькулятор.");

while (true)

{

Calculator calculator = new Calculator();

calculator.InputExpression();

if (EXIT())

break;

}

break;

}

Console.WriteLine();

}

}

static bool EXIT()

{

string exit;

Console.WriteLine("Для выхода введите 0/exit. Иначе Enter.");

exit = Console.ReadLine();

if (exit == "0" || exit == "exit")

return true;

return false;

}

}

}

1. Класс Line

using System;

class Line

{

private String line;

public Line()

{

Console.WriteLine("Введите строку:");

this.line = Console.ReadLine();

}

public void CountWordsAndLetters()

{

int countWords = 0, countLetters = 0;

countLetters = this.line.Length;

string[] words = this.line.Split(' ');

countWords = words.Length;

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

{

words[i] = words[i].Trim(':', ';', '?', '.', ',', '-', '!', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0');

if (words[i].Length == 0)

countWords--;

}

this.line += "-" + countLetters + "-" + countWords;

Console.WriteLine(line);

}

}

1. Класс Year

using System;

class Year

{

private int year;

public Year()

{

Console.WriteLine("Введите год:");

this.year = int.Parse(Console.ReadLine());

}

public void IsLeapYear()

{

if (year < 1 || year > 9999)

{

Console.WriteLine("Ошибка ввода, год должен быть задан в промежутке от 1 до 9999");

return;

}

if (DateTime.IsLeapYear(this.year))

Console.WriteLine("Год " + this.year + " високосный.");

else

Console.WriteLine("Год " + this.year + " не високосный.");

}

}

1. Класс Date

using System;

class Date

{

private DateTime firstDate;

private DateTime secondDate;

public Date()

{

Console.WriteLine("Введите дату через пробел число\\месяц\\год:");

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

secondDate = DateTime.Now;

}

public void ToDateTimeDiff()

{

TimeSpan result = this.secondDate - this.firstDate;

int differenceInDays = result.Days;

if (differenceInDays < 0)

differenceInDays = -differenceInDays;

Console.WriteLine("Разница между текущей датой и введенной датой: " + differenceInDays);

}

}

1. Класс RandomNumbers

using System;

namespace ЛР1

{

class RandomNumbers

{

private int quantity;

private int left;

private int right;

private int[] array;

void GenerateNumbers()

{

if (right < left)

{

Console.WriteLine("Ошибка ввода, максимальная граница больше минимальной.");

return;

}

this.right++;

this.array = new int[this.quantity];

Random rand = new Random();

for (int i = 0; i < this.quantity; i++)

array[i] = rand.Next(this.left, this.right);

}

void GetAnalysGenerateNumbers()

{

int countNumber = 0;

for (int i = left; i < this.right; i++)

{

for (int j = 0; j < this.quantity; j++)

{

if (this.array[j] == i)

countNumber++;

}

if (countNumber > 0)

{

Console.WriteLine("Число {0} сгенерировано {1} ", i, countNumber);

}

countNumber = 0;

}

}

public void AnalyzeRandomlyNumbers()

{

Console.WriteLine("Введите количество генераци числа:");

this.quantity = int.Parse(Console.ReadLine());

Console.WriteLine("Введите минимальную границу:");

this.left = int.Parse(Console.ReadLine());

Console.WriteLine("Введите максимальную границу:");

this.right = int.Parse(Console.ReadLine());

GenerateNumbers();

GetAnalysGenerateNumbers();

}

}

}

1. Класс Triangle

using System;

class Triangle

{

private double high;

private double side1;

private double side2;

private double side3;

private double angle;

private double square;

private string EXIT;

public void CalculateTriangleSquare()

{

while (true)

{

Console.WriteLine(

"1. Сторона и высота; \n2. Три стороны.\n3. Две стороны и угол.\nДля выхода введите 0 или exit.\nВведите вариант: ");

string task = Console.ReadLine();

switch (task)

{

case "0":

case "exit":

return;

case "1":

this.GetSquare\_HeightSide();

break;

case "2":

this.GetSquare\_ThreeSides();

break;

case "3":

this.GetSquare\_SidesAngle();

break;

}

}

}

void GetSquare\_HeightSide()

{

while (true)

{

Console.WriteLine("Введите длину стороны:");

this.side1 = int.Parse(Console.ReadLine());

Console.WriteLine("Введите длину высоты:");

this.high = int.Parse(Console.ReadLine());

this.square = this.side1\*this.high/2.0;

Console.WriteLine("Площадь треугольника: {0}", this.square);

Console.WriteLine("Для выхода введите 0/exit. Иначе Enter.");

EXIT = Console.ReadLine();

if (EXIT == "0" || EXIT == "exit")

break;

}

}

void GetSquare\_ThreeSides()

{

while (true)

{

Console.WriteLine("Введите длины сторон через Enter:");

this.side1 = double.Parse(Console.ReadLine());

this.side2 = double.Parse(Console.ReadLine());

this.side3 = double.Parse(Console.ReadLine());

if (this.IsTriangle())

{

double P = side1 + side2 + side3;

this.square = Math.Sqrt(P\*(P - side1)\*(P - side2)\*(P - side3));

}

else

{

Console.WriteLine("Не является треугольником.");

return;

}

Console.WriteLine("Площадь треугольника: {0}", this.square);

Console.WriteLine("Для выхода введите 0/exit. Иначе Enter.");

EXIT = Console.ReadLine();

if (EXIT == "0" || EXIT == "exit")

break;

}

}

void GetSquare\_SidesAngle()

{

while (true)

{

Console.WriteLine("Введите длины сторон через Enter:");

this.side1 = double.Parse(Console.ReadLine());

this.side2 = double.Parse(Console.ReadLine());

Console.WriteLine("Введите угол:");

this.angle = Math.PI\*double.Parse(Console.ReadLine())/180.0;

this.square = 0.5\*side1\*side2\*Math.Sin(this.angle);

Console.WriteLine("Площадь треугольника: {0}", this.square);

Console.WriteLine("Для выхода введите 0/exit. Иначе Enter.");

EXIT = Console.ReadLine();

if (EXIT == "0" || EXIT == "exit")

break;

}

}

bool IsTriangle()

{

if ((side1 + side2) > side3 && (side2 + side3) > side1 && (side1 + side3) > side2)

return true;

return false;

}

}

1. Класс NatureNumbers

using System;

using System.Linq;

class NaturalNumbers

{

private uint[] naturalNumbers;

private uint sumNumbers;

public NaturalNumbers()

{

Console.WriteLine("Введите натуральные числа через пробел:");

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

naturalNumbers = new uint[numbers.Length];

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

{

naturalNumbers[i] = uint.Parse(numbers[i]);

}

}

public void SumNaturalNumbers()

{

for (int i = 0; i < naturalNumbers.Count(); i++)

{

this.sumNumbers += naturalNumbers[i];

}

Console.WriteLine("Сумма чисел {0}", this.sumNumbers);

}

}

1. Класс Function

using System;

class Function

{

private int x;

public Function()

{

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

this.x = int.Parse(Console.ReadLine());

}

public void CalculateFunction()

{

if (this.x%2 == 0)

Console.WriteLine("F(x)={0:f3}", Math.Sqrt(x));

else

Console.WriteLine("F(x)={0:f2}", Math.Pow(x, 1/3f));

}

}

1. Класс Mark

using System;

class Mark

{

private string point;

public Mark()

{

Console.WriteLine("Введите оценку 1 до 10:");

this.point = Console.ReadLine();

}

public void GetMark()

{

switch (point)

{

case "1":

case "2":

Console.WriteLine("Плохо.");

break;

case "3":

case "4":

case "5":

Console.WriteLine("Удовлетворительно.");

break;

case "6":

case "7":

case "8":

Console.WriteLine("Хорошо.");

break;

case "9":

case "10":

Console.WriteLine("Отлично.");

break;

case "0":

case "exit":

break;

default:

Console.WriteLine("Неправильный ввод данных");

break;

}

}

}

1. Класс Array

using System;

class Array

{

private string array;

private int size;

public Array()

{

Console.WriteLine("Введите 30 символов. Чтобы закончить ввод раньше введите '?'");

char [] arrayChars=new char[30];

for (int i =0; i < 30; i++)

{

arrayChars[i]=Console.ReadKey().KeyChar;

if (arrayChars[i] == '?')

{

this.array = new string(arrayChars);

this.size = i;

break;

}

}

this.array = new string(arrayChars);

}

public void GetThreeLine()

{

if (this.size == 0)

{

Console.WriteLine("\nПустая строка.");

return;

}

this.array = this.array.Remove(size);

int sizeLine=this.array.Length/3;

int remainder = this.array.Length%3;

string first = this.array.Remove(sizeLine+remainder);

string second = this.array.Remove(0, sizeLine + remainder);

second = second.Remove(sizeLine + remainder);

string third = this.array.Remove(0,remainder+sizeLine\*2);

Console.WriteLine("\nРезультат работы программы:\n{0}",first);

Console.WriteLine(second);

Console.WriteLine(third);

}

}

1. Класс Matrix

using System;

using System.Collections.Generic;

public class Matrix

{

private List<List<int>> matrix = new List<List<int>>();

private int n;

public Matrix()

{

Console.WriteLine("Введите размерность матрицы:");

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

var rnd = new Random();

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

{

List<int> row = new List<int>();

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

row.Add(rnd.Next(0, 100));

matrix.Add(row);

}

}

public void Replace()

{

Console.WriteLine("Исходная матрица:");

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

{

for (int j = 0; j < this.n; j++)

Console.Write(this.matrix[i][j] + " ");

Console.WriteLine();

}

for (int j = 0; j < this.n; j++)

{

int maxElement = 0;

int row = 0;

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

{

if (this.matrix[i][j] > maxElement)

{

maxElement = matrix[i][j];

row = i;

}

}

matrix[row][j] = matrix[j][j];

matrix[j][j] = maxElement;

}

Console.WriteLine("Измененная матрица:");

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

{

for (int j = 0; j < this.n; j++)

Console.Write(this.matrix[i][j] + " ");

Console.WriteLine();

}

}

}

1. Класс Calculator

using System;

public class Calculator

{

private string[] expression;

private string[] result;

private double answer;

public Calculator()

{

Console.WriteLine("Программа может вычислять выражения с +, -, /, \*. Введите выражение:");

}

public void InputExpression()

{

string expressionLine = Console.ReadLine().Replace('.', ',');

if (expressionLine.IndexOf("/0") !=-1)

{

Console.WriteLine("Ошибка! На ноль делить нельзя.");

return;

}

if(expressionLine.Length==0)

{

Console.WriteLine("Ошибка!Пустое выражение.");

return;

}

string[] arithmetic = expressionLine.Split(new char[] { '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', ',' }, StringSplitOptions.RemoveEmptyEntries);

string[] expr = expressionLine.Split(new char[] { ' ', '+', '-', '/', '\*' }, StringSplitOptions.RemoveEmptyEntries);

this.expression = new string[expr.Length + expr.Length];

for (int i = 0, j = 0; i < expr.Length; i++)

{

expression[j] = expr[i];

j += 2;

}

if (arithmetic.Length == 0)

return;

for (int i = 0, j = 1; i < arithmetic.Length; i++)

{

expression[j] = arithmetic[i];

j += 2;

}

this.CalculateSmth();

}

public void CalculateSmth()

{

result = new string[this.expression.Length];

MultiplAndDivision();

PlusAndMinus();

Console.WriteLine(answer.ToString());

}

private void MultiplAndDivision()

{

for (int i = 0, j = 0; i < this.expression.Length - 1; i++)

{

if (expression[i].CompareTo("\*") != 0 && expression[i].CompareTo("/") != 0)

{

this.result[j] = this.expression[i];

j++;

}

else if (expression[i].CompareTo("\*") == 0)

{

this.answer = (double.Parse(this.result[j - 1]) \* double.Parse(this.expression[i + 1]));

this.result[j - 1] = answer.ToString();

i++;

}

else

{

answer = double.Parse(this.result[j - 1]) / double.Parse(this.expression[i + 1]);

this.result[j - 1] = answer.ToString();

i++;

}

}

}

private void PlusAndMinus()

{

for (int j = 0; j < this.result.Length - 1; j++)

{

if (result[j] == null)

break;

if (result[j].CompareTo("+") == 0)

answer = (double.Parse(this.result[j - 1]) + double.Parse(this.result[j + 1]));

this.result[j + 1] = answer.ToString();

j++;

}

if (result[j].CompareTo("-") == 0)

{

answer = double.Parse(this.result[j - 1]) - double.Parse(this.result[j + 1]);

this.result[j + 1] = answer.ToString();

j++;

}

}

}

}

1. **Выводы:**

В ходе выполнения данной лабораторной работы мною были освоены основные средства и синтаксические конструкции языка C# путем написания консольного приложения. При этом была разработана программная реализация 11-ти задач, предложенных вариантом задания. Решение общей задачи состояло из реализаций нескольких подзадач, предполагающих выполнение различных операций.

Для написания программы были использованы стандартные методы и свойства стандартных классов, а также разработан ряд собственных классов. Реализация каждой отдельной подзадачи была заключена в особый класс, содержащий метод, производящий решение подзадачи и выводящий результат на консоль, а так же дополнительные методы.

Диалог пользователя с программой в консоли был реализован посредством главного меню, каждый пункт которого содержал подзадачу. После выполнения подзадачи пользователю предлагался диалог о повторе, позволяющий продолжить работу с данной подзадачей или вернуться в меню, в самом же меню так же осуществлен диалог для выхода из программы.

В ходе выполнения работы было рассмотрено большинство основных средств и конструкции языка C#.