МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

#### Факультет информационных технологий и компьютерной безопасности

(факультет)

### Кафедра Систем автоматизированного проектирования и информационных систем

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине Программирование на платформе .NET Framework

Тема «Разработка пользовательских интерфейсов для программирования консольного ввода-вывода и основных операторов языка C# на платформе

.NET Framework»

Выполнили студенты группы ИСТ-222 Е.Г. Гладнева

Подпись, дата Инициалы, фамилия

А.Н. Грипунова

Подпись, дата Инициалы, фамилия

Е.В. Журавлёва

Подпись, дата Инициалы, фамилия

## Руководитель Э.И. Воробьев

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

2023

Лабораторная работа № 1

«Разработка пользовательских интерфейсов для программирования консольного ввода-вывода и основных операторов языка C# на платформе .NET Framework»

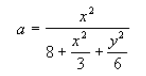
1 Цель работы

Целью лабораторной работы является изучение основ разработки консольного пользовательского интерфейса для программирования ввода-вывода и основных операторов языка C# на платформе .NET Framework.

2 Задание на лабораторную работу

Был выбран вариант №9.

Задание 1. Даны x, y, z . Вычислить a, b, если:



Задание 2. По длинам двух сторон треугольника и углу между ними найти длину третьей стороны и площадь треугольника.

Задание 3. Дана точка М(х, у). Проверить, принадлежит ли точка окружности единичного радиуса.

Задание 4. Создать структуру согласно предметной области варианта – Жилой дом. Добавить метод отображения данных структуры.

3 Ход выполнения

1. В первую очередь создадим интерактивное меню (рисунок 1), с помощью которого можно будет перейти к любому заданию. Для этого в основном классе Program напишем метод DrawMenu, который будет выводить список пунктов и выделять цветом текущий. Навигация по пунктам осуществляется с помощью бесконечного цикла вырисовывания меню и конструкции swith-case, принимающей в качестве выражения нажатие клавиши.

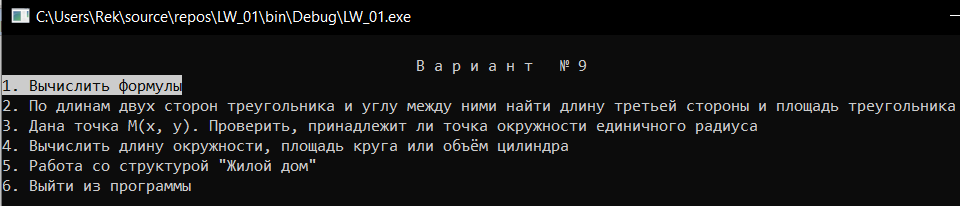


Рисунок 1 — Меню программы

2. Для выполнения первого задания потребуется метод GetDouble для получения числа двойной точности. Он поможет избежать ошибок при вводе символов и т.д., постоянно прося пользователя ввести корректные данные. Запрашиваем значение переменных *x*, *y*, *z*. Для математических функций был создан отдельный класс Calculator. Передаём значения переменных в методы func\_a и func\_b и выводим результаты на консоль (рисунок 2).

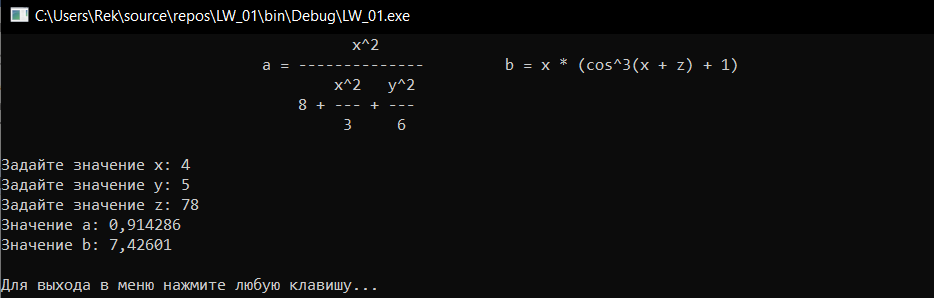


Рисунок 2 — Задание 1. Вычисление формул

3. Выполнение второго задания происходит похожим образом: получаем значения сторон треугольника и угла между ними с помощью GetDouble, но также проверяем, чтобы значение угла в градусах было от 0 до 360 (рисунок 3). Если данные корректны, подсчитываем третью сторону и площадь с помощью методов get\_third и get\_area соответственно, выводим на консоль (рисунок 4).

|  |  |
| --- | --- |
|  |  |
| Рисунок 3 — Проверка угла в задании 2 | Рисунок 4 — Корректное выполнение задания 2 |

4. В задании 3 к классу Calculator добавляется новый метод check\_point, в который передаются координаты точки. Фактически он высчитывает гипотенузу от центра окружности: если она больше 1, значит точка лежит за пределами единичной окружности (рисунок 5).

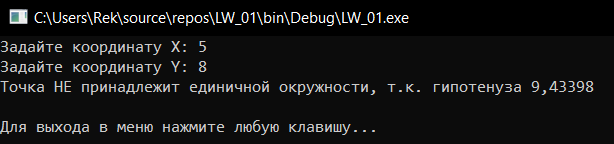


Рисунок 5 — Выполнение задания 3

5. Для выполнения следующего задания добавляем новый метод GetChar, действующий таким же образом, как и GetDouble, но для символов. Предлагаем пользователю выбрать объект для вычислений и проверяем символ на корректность. Если всё хорошо, направляем программу с помощью switch-case на вычисление требуемого объекта (длина окружности, площадь круга, объём цилиндра) (рисунок 6). При необходимости запрашиваем дополнительные данные (радиус, высота).

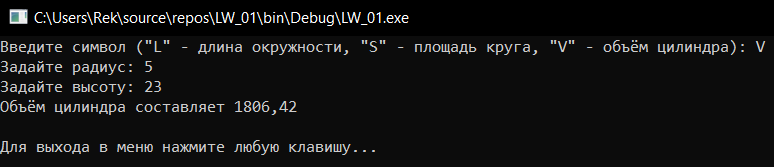


Рисунок 6 — Выполнение задания 4

6. Последним заданием является работа со структурой. В нашем представлении жилой дом (структура House) имеет следующие свойства: улица, номер дома, количество этажей, количество комнат, наличие двора. Для заполнения целочисленных свойств добавим очередной метод GetInt. Предлагаем пользователю заполнить перечисленные поля, после чего выводим информацию одной строкой с помощью метода info (рисунок 7).

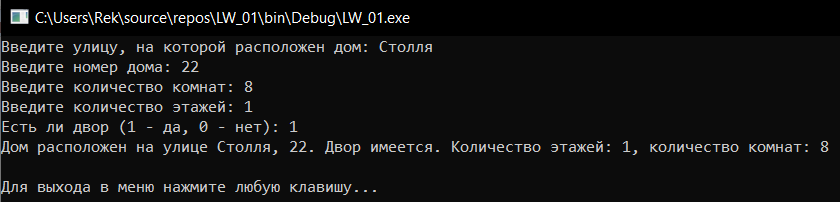


Рисунок 7 — Выполнение задания со структурой

Вывод

В результате выполнения лабораторной работы мы изучили основы разработки консольного пользовательского интерфейса для программирования ввода-вывода и основные операторы языка C# на платформе .NET Framework.

Приложение А

(обязательное)

Листинг программного кода языка C#

using System;

namespace LW\_01

{

internal class Program

{

static void Main(string[] args)

{

Calculator calc = new Calculator();

string[] items = new string[] { "1. Вычислить формулы",

"2. По длинам двух сторон треугольника и углу между ними найти длину третьей стороны и площадь треугольника",

"3. Дана точка М(х, у). Проверить, принадлежит ли точка окружности единичного радиуса",

"4. Вычислить длину окружности, площадь круга или объём цилиндра",

"5. Работа со структурой \"Жилой дом\"",

"6. Выйти из программы"};

int index = 0;

while (true)

{

DrawMenu(items, index);

switch (Console.ReadKey(true).Key)

{

case ConsoleKey.DownArrow:

if (index < items.Length - 1)

index++;

break;

case ConsoleKey.UpArrow:

if (index > 0)

index--;

break;

case ConsoleKey.Enter:

Console.Clear();

switch (index)

{

case 0:

{

double x, y, z;

DrawFormulas();

Console.Write("Задайте значение x: ");

x = GetDouble();

Console.Write("Задайте значение y: ");

y = GetDouble();

Console.Write("Задайте значение z: ");

z = GetDouble();

Console.WriteLine("Значение a: {0:g6}", calc.func\_a(x, y));

Console.WriteLine("Значение b: {0:g6}", calc.func\_b(x, z));

break;

}

case 1:

{

double first, second, angle;

Console.Write("Задайте значение первой стороны: ");

first = GetDouble();

Console.Write("Задайте значение второй стороны: ");

second = GetDouble();

Console.Write("Задайте значение угла в градусах: ");

angle = GetDouble();

if (angle < 0 || angle > 360)

{

Console.WriteLine("Угол не может быть меньше 0 или больше 360");

}

else

{

Console.WriteLine("Значение третьей стороны: {0:g6}", calc.get\_third(first, second, angle));

Console.WriteLine("Площадь треугольника: {0:g6}", calc.get\_area(first, second, angle));

}

break;

}

case 2:

{

double x, y, z;

Console.Write("Задайте координату X: ");

x = GetDouble();

Console.Write("Задайте координату Y: ");

y = GetDouble();

z = calc.check\_point(x, y);

if (z <= 1)

Console.WriteLine("Точка принадлежит единичной окружности, т.к. гипотенуза {0:g6}", z);

else

Console.WriteLine("Точка НЕ принадлежит единичной окружности, т.к. гипотенуза {0:g6}", z);

break;

}

case 3:

{

char sym;

Console.Write("Введите символ (\"L\" - длина окружности, \"S\" - площадь круга, \"V\" - объём цилиндра): ");

sym = GetChar();

if (sym != 'L' && sym != 'S' && sym != 'V')

Console.WriteLine("Нет такого варианта, попробуйте ещё раз");

else

{

switch (sym)

{

case 'L':

{

double r;

Console.Write("Задайте радиус: ");

r = GetDouble();

Console.WriteLine("Длина окружности составляет {0:g6}", calc.len\_circle(r));

break;

}

case 'S':

{

double r;

Console.Write("Задайте радиус: ");

r = GetDouble();

Console.WriteLine("Площадь круга составляет {0:g6}", calc.area\_circle(r));

break;

}

case 'V':

{

double r, h;

Console.Write("Задайте радиус: ");

r = GetDouble();

Console.Write("Задайте высоту: ");

h = GetDouble();

Console.WriteLine("Объём цилиндра составляет {0:g6}", calc.vol\_cylinder(r, h));

break;

}

}

}

break;

}

case 4:

{

House h = new House();

int choice;

Console.Write("Введите улицу, на которой расположен дом: ");

h.street = Console.ReadLine();

Console.Write("Введите номер дома: ");

h.number = GetInt();

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

h.count\_rooms = GetInt();

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

h.count\_floors = GetInt();

Console.Write("Есть ли двор (1 - да, 0 - нет): ");

choice = GetInt();

if (choice != 1 && choice != 0)

{

Console.Write("Нет такого варианта");

break;

}

h.yard = Convert.ToBoolean(choice);

h.info();

break;

}

case 5:

return;

}

Console.WriteLine("\nДля выхода в меню нажмите любую клавишу...");

Console.ReadKey();

Console.Clear();

break;

}

}

}

private static void DrawMenu(string[] items, int index)

{

string variant = "В а р и а н т № 9";

int centerX = (Console.WindowWidth / 2) - (variant.Length / 2) - 5;

Console.SetCursorPosition(centerX, 1);

Console.WriteLine(variant);

Console.SetCursorPosition(0, 3);

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

{

if (i == index)

{

Console.BackgroundColor = Console.ForegroundColor;

Console.ForegroundColor = ConsoleColor.Black;

}

Console.WriteLine(items[i]);

Console.ResetColor();

}

Console.WriteLine();

}

private static void DrawFormulas()

{

int centerX = (Console.WindowWidth / 2) - ("a = -------------- b = x \* (cos^3(x + z) + 1)".Length / 2) - 5;

int startY = Console.CursorTop;

Console.SetCursorPosition(centerX, startY++);

Console.WriteLine(" x^2");

Console.SetCursorPosition(centerX, startY++);

Console.WriteLine("a = -------------- b = x \* (cos^3(x + z) + 1)");

Console.SetCursorPosition(centerX, startY++);

Console.WriteLine(" x^2 y^2");

Console.SetCursorPosition(centerX, startY++);

Console.WriteLine(" 8 + --- + ---");

Console.SetCursorPosition(centerX, startY++);

Console.WriteLine(" 3 6");

startY++;

Console.SetCursorPosition(0, startY++);

}

private static double GetDouble()

{

int cur\_y = Console.CursorTop;

int cur\_x = Console.CursorLeft;

double res;

while (!double.TryParse(Console.ReadLine(), out res))

{

Console.SetCursorPosition(cur\_x, cur\_y);

Console.WriteLine("Не удалось получить значение, попробуйте ещё раз. Нажмите любую клавишу...");

Console.ReadKey();

Console.SetCursorPosition(cur\_x, cur\_y);

Console.Write(new string(' ', Console.WindowWidth));

Console.SetCursorPosition(cur\_x, cur\_y);

}

return res;

}

private static char GetChar()

{

int cur\_y = Console.CursorTop;

int cur\_x = Console.CursorLeft;

char res;

while (!char.TryParse(Console.ReadLine(), out res))

{

Console.SetCursorPosition(cur\_x, cur\_y);

Console.WriteLine("Не удалось получить символ, попробуйте ещё раз. Нажмите любую клавишу...");

Console.ReadKey();

Console.SetCursorPosition(cur\_x, cur\_y);

Console.Write(new string(' ', Console.WindowWidth));

Console.SetCursorPosition(cur\_x, cur\_y);

}

return res;

}

private static int GetInt()

{

int cur\_y = Console.CursorTop;

int cur\_x = Console.CursorLeft;

int res;

while (!int.TryParse(Console.ReadLine(), out res))

{

Console.SetCursorPosition(cur\_x, cur\_y);

Console.WriteLine("Не удалось получить число, попробуйте ещё раз. Нажмите любую клавишу...");

Console.ReadKey();

Console.SetCursorPosition(cur\_x, cur\_y);

Console.Write(new string(' ', Console.WindowWidth));

Console.SetCursorPosition(cur\_x, cur\_y);

}

return res;

}

}

struct House

{

public string street;

public int number;

public int count\_floors;

public int count\_rooms;

public bool yard;

public void info()

{

Console.Write("Дом расположен на улице {0}, {1}.", street, number);

if (yard)

Console.Write(" Двор имеется. ");

else

Console.Write(" Двора нету. ");

Console.Write("Количество этажей: {0}, количество комнат: {1}\n", count\_floors, count\_rooms);

}

}

public class Calculator

{

public double func\_a(double x, double y) => Math.Pow(x, 2) / (8 + Math.Pow(x, 2) / 3 + Math.Pow(y, 2) / 6);

public double func\_b(double x, double z) => x \* (Math.Pow(Math.Cos(z + x), 3) + 1);

public double get\_third(double a, double b, double ang) => Math.Sqrt(Math.Pow(a, 2) + Math.Pow(b, 2) - 2 \* a \* b \* Math.Cos(ang \* Math.PI / 180));

public double get\_area(double a, double b, double ang) => 0.5 \* a \* b \* Math.Sin(ang \* Math.PI / 180);

public double check\_point(double a, double y) => Math.Sqrt(Math.Pow(a, 2) + Math.Pow(y, 2));

public double len\_circle(double r) => 2 \* Math.PI \* r;

public double area\_circle(double r) => Math.PI \* Math.Pow(r, 2);

public double vol\_cylinder(double r, double h) => Math.PI \* Math.Pow(r, 2) \* h;

}

}