Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по практическому заданию

«Модуль 1.1. Основы C#»

по учебной практике по программированию

Выполнила учащаяся Жданова С.О.

группы 3ПОИС23 29.09.2025

Проверил Алейников М.А.

29.09.2025

Орша, 2025

**Цель:** применить на практике ранее полученные знания в основах языка программирования C#.

**Задания (вариант №4)**

**Задание 1.**

Создайте программу, которая генерирует случайное число от 1 до 100 и предлагает пользователю угадать это число. При каждой попытке программа дает подсказку, больше или меньше введенное число загаданного.

**Задание 2.**

Напишите программу, которая находит среднее арифметическое из трех введенных пользователем чисел.

**Задание 3.**

Напишите программу, которая принимает на вход две строки и определяет, является ли вторая строка подстрокой первой строки.

**Задание 4.**

Напишите программу, которая создает массив из 10 целых чисел, заполняет его случайными числами и выводит сумму всех элементов массива.

**Задание 5.**

Напишите программу, которая выводит на экран числа от 1 до 100. Если число делится на 3, вместо него выводится «Fizz», если на 5 – «Buzz», а если на оба – «FizzBuzz».

**Ход выполнения работы**

**Выполнение практических заданий**

**Задание 1.**

Создайте программу, которая генерирует случайное число от 1 до 100 и предлагает пользователю угадать это число. При каждой попытке программа дает подсказку, больше или меньше введенное число загаданного.

Ниже представлен листинг 1.1 с кодом программы для генерации случайного числа от 1 до 100 и предложением пользователю угадать загаданное число.

Листинг 1.1. Игра – угадывание числа.

using System;

namespace GuessTheNumber

{

class Program

{

static void Main(string[] args)

{

// Создаём объект Random для генерации случайных чисел

Random random = new Random();

// Флаг для управления повторением игры

bool playAgain = true;

// Основной цикл игры, продолжается, пока игрок хочет играть

while (playAgain)

{

// Генерируем случайное число от 1 до 100

int secretNumber = random.Next(1, 101);

// Переменные для хранения догадки, количества попыток и флага угадывания

int guess;

int attempts = 0;

int maxAttempts = 10;

bool hasGuessed = false;

// Приветственное сообщение с правилами игры

Console.WriteLine("Я загадал число от 1 до 100. У тебя есть 10 попыток, чтобы угадать!");

// Цикл для обработки попыток угадывания

while (!hasGuessed && attempts < maxAttempts)

{

// Запрашиваем у игрока ввод числа

Console.Write($"Попытка {attempts + 1}. Введи своё число: ");

string input = Console.ReadLine();

attempts++;

// Проверяем, является ли ввод корректным числом

if (!int.TryParse(input, out guess))

{

Console.WriteLine("Пожалуйста, введи число!");

attempts--; // Не засчитываем попытку при некорректном вводе

continue;

}

// Сравниваем введённое число с загаданным

if (guess == secretNumber)

{

// Если число угадано, выводим сообщение и завершаем цикл

Console.WriteLine($"Поздравляю! Ты угадал число за {attempts} попыток!");

hasGuessed = true;

}

else if (guess < secretNumber)

{

// Подсказка, если введённое число меньше загаданного

Console.WriteLine("Моё число больше!");

}

else

{

// Подсказка, если введённое число больше загаданного

Console.WriteLine("Моё число меньше!");

}

}

// Если число не угадано за максимум попыток, выводим загаданное число

if (!hasGuessed)

{

Console.WriteLine($"Увы, попытки закончились! Загаданное число было {secretNumber}.");

}

// Запрашиваем, хочет ли игрок сыграть ещё раз

Console.Write("Хочешь сыграть ещё раз? (да/нет): ");

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

// Продолжаем игру, если ответ "да" или "yes"

playAgain = (answer == "да" || answer == "yes");

}

// Прощальное сообщение после завершения игры

Console.WriteLine("Спасибо за игру!");

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.1.1.

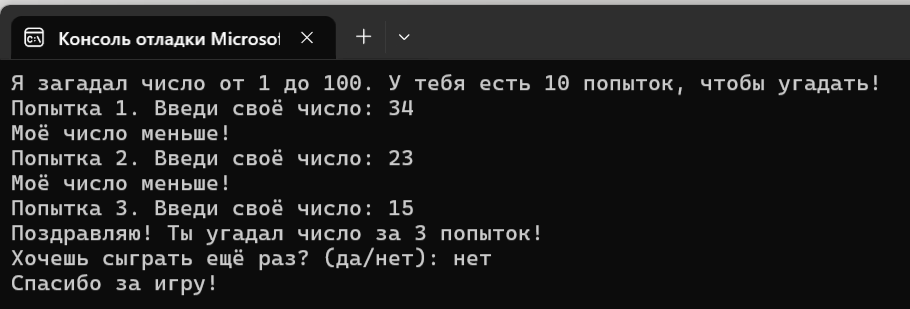


Рисунок 1.1.1 – Результат игры – угадывание числа

**Задание 2.**

Напишите программу, которая находит среднее арифметическое из трех введенных пользователем чисел.

Ниже представлен листинг 1.2 с кодом программы для нахождения среднего арифметического из трех введенных чисел.

Листинг 1.2. Код программы для нахождения среднего арифметического числа (сумма чисел, делённая на 3).

using System;

namespace AverageCalculator

{

class Program

{

static void Main(string[] args)

{

// Выводим приглашение для ввода трёх чисел

Console.WriteLine("Введите три числа для вычисления среднего арифметического.");

// Запрашиваем первое число от пользователя

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

string input1 = Console.ReadLine(); // Считываем введённую строку

// Проверяем, является ли ввод корректным числом

if (!double.TryParse(input1, out double number1)) // Преобразуем строку в число типа double

{

Console.WriteLine("Ошибка: введите корректное число!"); // Сообщение об ошибке, если ввод некорректен

return; // Завершаем программу, если введено не число

}

// Запрашиваем второе число от пользователя

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

string input2 = Console.ReadLine(); // Считываем вторую строку

// Проверяем корректность второго ввода

if (!double.TryParse(input2, out double number2)) // Преобразуем строку в число

{

Console.WriteLine("Ошибка: введите корректное число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Запрашиваем третье число от пользователя

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

string input3 = Console.ReadLine(); // Считываем третью строку

// Проверяем корректность третьего ввода

if (!double.TryParse(input3, out double number3)) // Преобразуем строку в число

{

Console.WriteLine("Ошибка: введите корректное число!"); // Сообщение об ошибке

return; // Завершаем программу при некорректном вводе

}

// Вычисляем среднее арифметическое: сумма трёх чисел, делённая на 3

double average = (number1 + number2 + number3) / 3;

// Выводим результат, форматируя число до двух знаков после запятой

Console.WriteLine($"Среднее арифметическое чисел {number1}, {number2}, {number3} равно {average:F2}");

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.1.2.

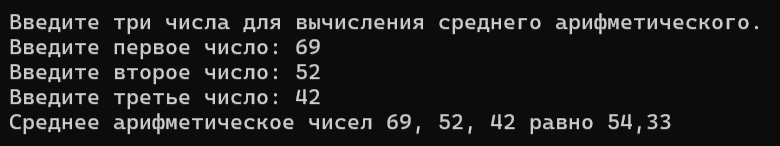


Рисунок 1.1.2 – Результат нахождения среднего арифмитического числа

**Задание 3.**

Напишите программу, которая принимает на вход две строки и определяет, является ли вторая строка подстрокой первой строки.

Ниже представлен листинг 1.3 с кодом программы для ввода двух строк и проверкой, является ли вторая строка подстрокой первой строки.

Листинг 1.3. Код программы для проверки строки на подстроку.

using System;

namespace SubstringChecker

{

class Program

{

static void Main(string[] args)

{

// Выводим приветственное сообщение с инструкцией

Console.WriteLine("Программа проверяет, является ли вторая строка подстрокой первой (без учёта регистра).");

// Запускаем бесконечный цикл для многократной проверки строк

while (true)

{

// Запрашиваем первую строку от пользователя

Console.Write("Введите первую строку: ");

string firstString = Console.ReadLine(); // Считываем первую строку

// Проверяем, не является ли первая строка пустой или null

if (string.IsNullOrEmpty(firstString))

{

Console.WriteLine("Ошибка: первая строка не может быть пустой!"); // Сообщение об ошибке

continue; // Возвращаемся к началу цикла для нового ввода

}

// Запрашиваем вторую строку от пользователя

Console.Write("Введите вторую строку: ");

string secondString = Console.ReadLine(); // Считываем вторую строку

// Проверяем, не является ли вторая строка пустой или null

if (string.IsNullOrEmpty(secondString))

{

Console.WriteLine("Ошибка: вторая строка не может быть пустой!"); // Сообщение об ошибке

continue; // Возвращаемся к началу цикла для нового ввода

}

// Проверяем, является ли вторая строка подстрокой первой, игнорируя регистр

bool isSubstring = firstString.ToLower().Contains(secondString.ToLower()); // Приводим обе строки к нижнему регистру

// Находим позицию подстроки (если она есть)

int position = firstString.ToLower().IndexOf(secondString.ToLower()); // Возвращает индекс первого вхождения или -1, если не найдено

// Выводим результат проверки

if (isSubstring)

{

// Выводим сообщение с позицией подстроки

Console.WriteLine($"Да, строка \"{secondString}\" является подстрокой строки \"{firstString}\".");

Console.WriteLine($"Она начинается с позиции {position + 1} (нумерация с 1).");

}

else

{

// Выводим сообщение, если подстрока не найдена

Console.WriteLine($"Нет, строка \"{secondString}\" не является подстрокой строки \"{firstString}\".");

}

// Спрашиваем, хочет ли пользователь продолжить

Console.Write("Хотите проверить ещё раз? (да/нет): ");

string continueAnswer = Console.ReadLine().ToLower(); // Считываем ответ и приводим к нижнему регистру

if (continueAnswer != "да" && continueAnswer != "yes") // Проверяем, хочет ли пользователь выйти

{

break; // Выходим из цикла, если ответ не "да" или "yes"

}

Console.WriteLine(); // Пустая строка для удобства чтения

}

// Выводим прощальное сообщение при завершении программы

Console.WriteLine("Спасибо за использование программы!");

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.1.3.

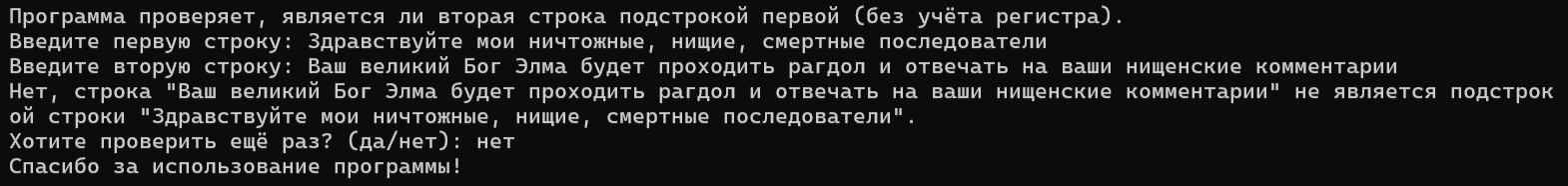


Рисунок 1.1.3 – Результат проверки строки на подстроку

**Задание 4.**

Напишите программу, которая создает массив из 10 целых чисел, заполняет его случайными числами и выводит сумму всех элементов массива.

Ниже представлен листинг 1.4 с кодом программы для создания массива из 10 целых случайных чисел и вычислением суммы всех элементов массива.

Листинг 1.4. Код программы для создания массива и вычисления суммы всех элементов массива.

using System;

namespace ArraySum

{

class Program

{

static void Main(string[] args)

{

// Выводим сообщение о назначении программы

Console.WriteLine("Программа создаёт массив из 10 случайных целых чисел и вычисляет их сумму.");

// Создаём объект для генерации случайных чисел

Random random = new Random();

// Создаём массив из 10 целых чисел

int[] numbers = new int[10];

// Заполняем массив случайными числами от 1 до 100

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

{

numbers[i] = random.Next(1, 101); // Генерируем число от 1 до 100 (101 не включается)

}

// Выводим элементы массива

Console.Write("Созданный массив: ");

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

{

Console.Write(numbers[i]);

if (i < numbers.Length - 1)

{

Console.Write(", "); // Добавляем запятую между элементами, кроме последнего

}

}

Console.WriteLine(); // Переходим на новую строку после вывода массива

// Вычисляем сумму элементов массива

int sum = 0;

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

{

sum += numbers[i]; // Добавляем каждый элемент к сумме

}

// Выводим сумму элементов

Console.WriteLine($"Сумма всех элементов массива: {sum}");

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.1.4.

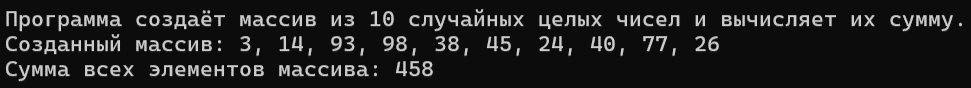


Рисунок 1.1.4 – Результат создания массива и вычисления суммы всех чисел

**Задание 5.**

Напишите программу, которая выводит на экран числа от 1 до 100. Если число делится на 3, вместо него выводится «Fizz», если на 5 – «Buzz», а если на оба – «FizzBuzz».

Ниже представлен листинг 1.5 с кодом программы для вывода чисел от 1 до 100 на экран со своими особенностями.

Листинг 1.5. Код программы для вывода чисел.

using System;

namespace FizzBuzz

{

class Program

{

static void Main(string[] args) // Точка входа в программу

{

// Выводим сообщение о назначении программы

Console.WriteLine("Программа выводит числа от 1 до 100, заменяя числа, делящиеся на 3, на 'Fizz', на 5 — на 'Buzz', на оба — на 'FizzBuzz'.");

// Цикл от 1 до 100

for (int i = 1; i <= 100; i++)

{

// Проверяем, делится ли число одновременно на 3 и на 5 (на 15)

if (i % 3 == 0 && i % 5 == 0)

{

Console.WriteLine("FizzBuzz"); // Выводим 'FizzBuzz' для чисел, делящихся на 15

}

// Проверяем, делится ли число только на 3

else if (i % 3 == 0)

{

Console.WriteLine("Fizz"); // Выводим 'Fizz' для чисел, делящихся на 3

}

// Проверяем, делится ли число только на 5

else if (i % 5 == 0)

{

Console.WriteLine("Buzz"); // Выводим 'Buzz' для чисел, делящихся на 5

}

// Если число не делится ни на 3, ни на 5, выводим само число

else

{

Console.WriteLine(i); // Выводим число

}

}

}

}

}

Далее запускаем написанный ранее код в среде разработке для получения результата, результат представлен на рисунке 1.1.5.

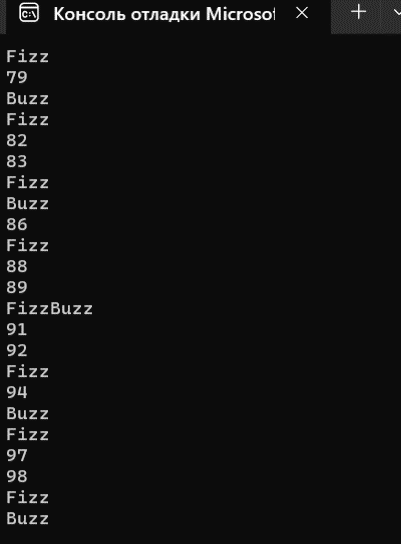


Рисунок 1.1.5 – Результат вывода чисел со своими особенностями

**Вывод:** в ходе выполнения практического задания были применены ранее полученные знания в основах языка программирования C# на практике.