МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии

Департамент цифровых, робототехнических систем и электроники

Отчет к практической работе № 3

Тема: **«Управление потоком выполнения в программе»**

Дисциплина: **«Алгоритмы и структуры данных»**

|  |  |  |
| --- | --- | --- |
|  |  | **Выполнил:** |
|  |  | Студент группы ПИЖ-б-о-23-1, направление подготовки: 09.03.04 «Программная инженерия»  Панчешный Александр Алексеевич  **Проверил:**  Доцент ДЦРСИЭ  Николаев Евгений Иванович |

Ставрополь 2025

**Лабораторная работа 3.** **Управление потоком выполнения в программе**

**Цель работы:** изучить операторы, позволяющие организовывать непоследовательное выполнение программного кода.

**Ход работы**

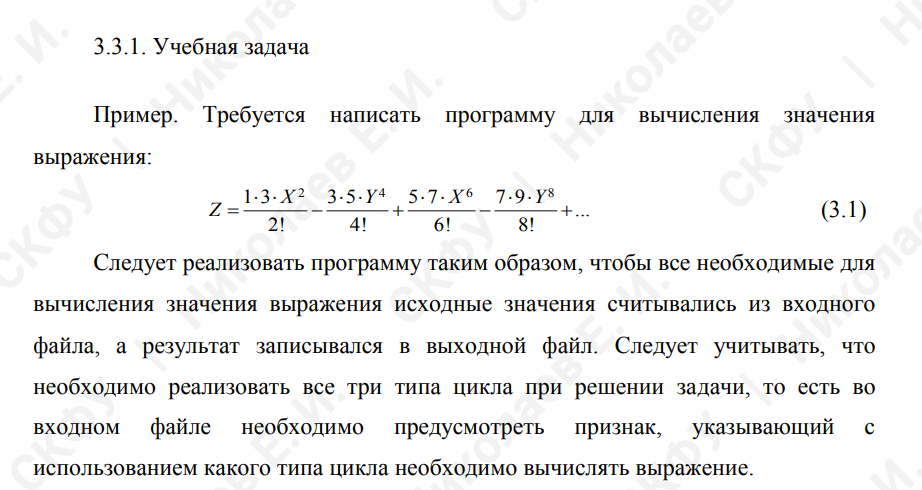
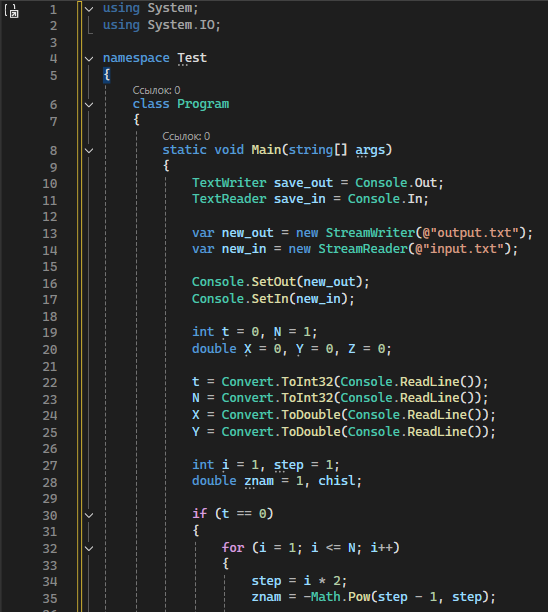


Рисунок 1. Учебное задание



Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2. Решение учебной задачи

**I. Задание и исходный код индивид. задания**

**Задание:**



**Исходный код:**

using System;

using System.IO;

public class Program

{

public static void Main(string[] args)

{

// Чтение входных данных из файла "input.txt"

double Y, X, Z = 0;

int t, n = 4; // n - количество членов ряда (можно изменить)

try

{

using (StreamReader reader = new StreamReader("input.txt"))

{

t = int.Parse(reader.ReadLine());

Y = double.Parse(reader.ReadLine());

X = double.Parse(reader.ReadLine());

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка чтения файла: {ex.Message}");

return;

}

// Выбор цикла на основе значения t

if (t == 0) // Цикл for

{

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

{

Z += Math.Pow(Y, 2 \* i) \* Math.Pow(X, 2 \* i - 1) / (2 \* i);

}

}

else if (t == 1) // Цикл while

{

int i = 1;

while (i <= n)

{

Z += Math.Pow(Y, 2 \* i) \* Math.Pow(X, 2 \* i - 1) / (2 \* i);

i++;

}

}

else if (t == 2) // Цикл do-while

{

int i = 1;

do

{

Z += Math.Pow(Y, 2 \* i) \* Math.Pow(X, 2 \* i - 1) / (2 \* i);

i++;

} while (i <= n);

}

else

{

Console.WriteLine("Неверное значение t (должно быть 0, 1 или 2).");

return;

}

// Запись результата в файл "output.txt"

try

{

using (StreamWriter writer = new StreamWriter("output.txt"))

{

writer.WriteLine(Z);

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка записи файла: {ex.Message}");

return;

}

Console.WriteLine($"Результат записан в файл output.txt: Z = {Z}");

}

}

**II. Контрольные вопросы**

**1. Опишите, каким образом оформляется комментарий в языке C#.**

В языке C# комментарии оформляются следующим образом:

* **Однострочные комментарии:** Используются для добавления комментариев на одну строку. Начинаются с двойного слеша //. Все символы после // до конца строки игнорируются компилятором.

int x = 10; // Это комментарий в конце строки

* **Многострочные комментарии:** Используются для добавления комментариев на несколько строк. Начинаются с /\* и заканчиваются \*/. Все символы между /\* и \*/ игнорируются компилятором.

/\*

Это многострочный комментарий.

Он может занимать несколько строк.

\*/

* **XML-комментарии (документирующие комментарии):** Используются для создания документации к коду. Начинаются с тройного слеша ///. Они позволяют добавлять структурированную информацию о типах, методах и других элементах кода.

**2. Какое ключевое (зарезервированное) слово в условном операторе является обязательным?**

В условном операторе if в C# обязательным ключевым словом является if. Ключевое слово else является необязательным и используется для выполнения блока кода, если условие в if не выполняется.

**3. Дан фрагмент кода:**

double x = 2, y = 3, z = 4, res;

Boolean yes = true;

if (!yes)

{

res = Math.Pow(x, y) \* z;

}

else

{

res = Math.Pow(z, x) \* y;

}

**Вопрос:** Чему равно значение переменной res после выполнения данного фрагмента?

**Ответ:**

1. Переменной yes присваивается значение true.
2. В условном операторе if проверяется условие !yes, что означает "не yes". Так как yes равно true, то !yes равно false.
3. Условие if не выполняется, поэтому выполняется блок кода в else.
4. В блоке else вычисляется res = Math.Pow(z, x) \* y.
5. Math.Pow(z, x) вычисляет z в степени x, то есть 4^2 = 16.
6. Затем результат умножается на y: 16 \* 3 = 48.
7. Значение 48 присваивается переменной res.

**4. Дан фрагмент кода:**

int counter = 10;

const int N = 10;

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

{

counter += i;

}

**Вопрос:** Чему равно значение переменной counter после выполнения данного фрагмента?

**Ответ:**

1. Переменной counter присваивается начальное значение 10.
2. Цикл for выполняется N раз, где N равно 10.
3. В каждой итерации цикла к counter добавляется значение переменной i.
4. Сумма чисел от 0 до 9 равна (0 + 9) \* 10 / 2 = 45.
5. После выполнения цикла значение counter будет равно 10 + 45 = 55.

**5. Дан фрагмент кода:**

int seo = 0;

const int N = 5;

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

{

seo += 2;

}

**Вопрос:** Чему равно значение переменной seo после выполнения данного фрагмента?

**Ответ:**

1. Переменной seo присваивается начальное значение 0.
2. Цикл for выполняется N - 1 раз, где N равно 5. То есть цикл выполняется 4 раза (при i от 1 до 4 включительно).
3. В каждой итерации цикла к seo добавляется значение 2.
4. После выполнения цикла значение seo будет равно 0 + 2 \* 4 = 8.

**6. Что будет выведено в консоль в результате выполнения следующей программы?**

static void Main(string[] args)

{

int a = 0, b = 5, c = 0;

for (; a < b;)

{

a += 5;

b += 2;

c++;

}

Console.Write(c);

Console.ReadKey();

}

**Ответ:**

1. Инициализируем переменные: a = 0, b = 5, c = 0.
2. Цикл for без инициализации и инкремента выполняется, пока a < b.
3. Внутри цикла:
   * a становится 5, b становится 7, c становится 1.
   * a становится 10, b становится 9, c становится 2.
4. На следующей итерации условие a < b (10 < 9) не выполняется, цикл завершается.
5. В консоль выводится значение c, которое равно 2.

**7. Что будет выведено в консоль в результате выполнения следующей программы?**

static void Main(string[] args)

{

int a = 0, b = 5, c = 0;

while (a < b)

{

a += 5;

b += 2;

c++;

}

Console.Write(c);

Console.ReadKey();

}

**Ответ:**

1. Инициализируем переменные: a = 0, b = 5, c = 0.
2. Цикл while выполняется, пока a < b.
3. Внутри цикла:
   * a становится 5, b становится 7, c становится 1.
   * a становится 10, b становится 9, c становится 2.
4. На следующей итерации условие a < b (10 < 9) не выполняется, цикл завершается.
5. В консоль выводится значение c, которое равно 2.

**8. Что будет выведено в консоль в результате выполнения следующей программы?**

static void Main(string[] args)

{

int a = 3, z = 0;

Boolean \_stop = false;

while (!\_stop)

{

z += a;

if (z > 100)

{

\_stop = true;

}

}

Console.Write(z);

Console.ReadKey();

}

**Ответ:**

1. Инициализируем переменные: a = 3, z = 0, \_stop = false.
2. Цикл while выполняется, пока !\_stop истинно (то есть пока \_stop ложно).
3. Внутри цикла:
   * z увеличивается на a (3).
   * Если z становится больше 100, \_stop становится true, и цикл завершается.
4. Выполняем цикл, пока z не станет больше 100:
   * z = 0 + 3 = 3
   * z = 3 + 3 = 6
   * ...
   * z = 99 + 3 = 102
5. Когда z становится 102, \_stop становится true, и цикл завершается.
6. В консоль выводится значение z, которое равно 102.

**9. Что будет выведено в консоль в результате выполнения следующей программы?**

static void Main(string[] args)

{

int a = 3, z = 0;

do

{

z += a;

} while (z-- != 0);

Console.Write(z);

Console.ReadKey();

}

**Ответ:**

Программа ничего не выведет, представляет собой бесконечный цикл while.

**10. Дан листинг программы:**

static void Main(string[] args)

{

int a = 3;

int stop = 1;

while (stop < 10)

{

if (a % 2 == 0)

{

a++;

}

else

{

continue;

}

stop++;

}

Console.Write(a);

Console.ReadKey();

}

**Вопрос:** Что будет выведено в консоль?

**Ответ:**

1. Инициализируем переменные: a = 3, stop = 1.
2. Цикл while выполняется, пока stop < 10.
3. Внутри цикла:
   * Если a четное (a % 2 == 0), то a увеличивается на 1.
   * Если a нечетное, то выполняется continue, и оставшаяся часть текущей итерации цикла пропускается.
   * stop увеличивается на 1.
4. На первой итерации a = 3 (нечетное), поэтому выполняется continue. stop становится 2.
5. На второй итерации a = 3 (нечетное), поэтому выполняется continue. stop становится 3.
6. Это продолжается, пока stop не станет 10.
7. Цикл while выполняется 9 раз, но a никогда не становится четным, поэтому a не увеличивается.
8. В консоль выводится значение a, которое остается равным 3.