1. **Что такое функции и зачем они нужны?**

* **Функция** — это блок кода, который выполняет конкретную задачу и может быть вызван по имени.
* **Преимущества:**
  + Повторное использование кода.
  + Упрощение отладки.
  + Улучшение структуры программы.

2. **Синтаксис функций в C#**

* **Объявление функции:**

модификатор\_доступа возвращаемый\_тип ИмяМетода(параметры)

{

// Тело метода

}

* **Примеры:**
  + Метод **без возвращаемого значения** (void):

static void PrintHello()

{

Console.WriteLine("Hello!");

}

* + Метод **с возвращаемым значением** и параметрами:

static int Add(int a, int b)

{

return a + b;

}

3. **Ключевое слово**static

* В консольных приложениях C# методы, вызываемые из Main, должны быть статическими (static), если они находятся в том же классе.
* Пример вызова методов:

static void Main(string[] args)

{

PrintHello(); // Вызов метода

int result = Add(5, 3); // result = 8

Console.WriteLine(result);

}

4. **Параметры и аргументы**

* **Параметры** — переменные, указанные в объявлении метода.
* **Аргументы** — конкретные значения, передаваемые в метод при вызове.
* Пример:

static void Greet(string name, int age)

{

Console.WriteLine($"Привет, {name}! Тебе {age} лет.");

}

// Вызов:

Greet("Анна", 25); // Аргументы: "Анна" и 25

5. **Область видимости переменных**

* **Локальные переменные** — объявлены внутри метода и доступны только в нём.
* **Параметры метода** — действуют как локальные переменные.
* **Глобальные переменные** (поля класса) — объявляются внутри класса, но вне методов, и доступны всем методам класса.

**Пример:**

class Program

{

static int globalValue = 10; // Глобальная переменная

static void Main()

{

int localValue = 5; // Локальная переменная Main

Console.WriteLine(localValue + globalValue); // 15

}

static void AnotherMethod()

{

// Console.WriteLine(localValue); // Ошибка: переменная не существует здесь!

Console.WriteLine(globalValue); // Работает

}

}

**Практическое задание**

**Задача 1: Калькулятор**

Создайте методы для основных арифметических операций и вызовите их:

static int Add(int a, int b) => a + b;

static int Subtract(int a, int b) => a - b;

static int Multiply(int a, int b) => a \* b;

static double Divide(int a, int b) => (double)a / b;

// В Main:

Console.WriteLine(Add(10, 5)); // 15

Console.WriteLine(Subtract(10, 5)); // 5

Console.WriteLine(Divide(10, 3)); // 3.333...

**Задача 2: Проверка на чётность**

Вынесите проверку чётности числа в отдельный метод:

static bool IsEven(int number)

{

return number % 2 == 0;

}

// В Main:

Console.Write("Введите число: ");

int num = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(IsEven(num) ? "Чётное" : "Нечётное");

**Задача 3: Модифицированная игра «Угадай число»**

Реорганизуйте код из прошлого урока, используя функции:

static int GenerateSecretNumber()

{

Random random = new Random();

return random.Next(1, 11);

}

static int GetUserGuess()

{

Console.Write("Угадайте число (1-10): ");

return Convert.ToInt32(Console.ReadLine());

}

// В Main:

int secretNumber = GenerateSecretNumber();

int attempts = 0;

int guess;

do

{

guess = GetUserGuess();

attempts++;

GiveHint(guess, secretNumber);

} while (guess != secretNumber);

Console.WriteLine($"Угадано за {attempts} попыток!");

**Задача 4 (дополнительно): Рекурсивный факториал**

Напишите метод, вычисляющий факториал с использованием рекурсии:

static int Factorial(int n)

{

if (n == 0)

return 1;

else

return n \* Factorial(n - 1);

}

// В Main:

Console.WriteLine(Factorial(5)); // 120