**Присваивание**

Операции присвоения устанавливают значение. В операциях присвоения участвуют два операнда, причем левый операнд может представлять только модифицируемое именованное выражение, например, переменную

Как и во многих других языках программирования, в C# имеется базовая операция присваивания =, которая присвоивает значение правого операнда левому операнду:

int number = 23;

Также можно выполнять множественно присвоение сразу нескольких переменным одновременно:

int a, b, c;

a = b = c = 34;

Стоит отметить, что операции присвоения имеют низкий приоритет. И вначале будет вычисляться значение правого операнда и только потом будет идти присвоение этого значения левому операнду. Например:

int a, b, c;

a = b = c = 34 \* 2 / 4; // 17

**Вывод результатов**

Для вывода информации на консоль мы уже использовали встроенный метод Console.WriteLine. То есть, если мы хотим вывести некоторую информацию на консоль, то нам надо передать ее в метод Console.WriteLine:

string hello = "Привет мир";

Console.WriteLine(hello);

Console.WriteLine("Добро пожаловать в C#!");

Console.WriteLine("Пока мир...");

Console.WriteLine(24.5);

Нередко возникает необходимость вывести на консоль в одной строке значения сразу нескольких переменных. В этом случае мы можем использовать прием, который называется интерполяцией:

string name = "Tom";

int age = 34;

double height = 1.7;

Console.WriteLine($"Имя: {name} Возраст: {age} Рост: {height}м");

**Ввод данных**

Кроме вывода информации на консоль мы можем получать информацию с консоли. Для этого предназначен метод Console.ReadLine(). Он позволяет получить введенную строку.

Console.Write("Введите свое имя: ");

string? name = Console.ReadLine();

Console.WriteLine($"Привет {name}");

В данном случае все, что вводит пользователь, с помощью метода Console.ReadLine() передается в переменную name.

Особенностью метода Console.ReadLine() является то, что он может считать информацию с консоли только в виде строки. Кроме того, возможная ситуация, когда для метода Console.ReadLine не окажется доступных для считывания строк, то есть когда ему нечего считывать, он возвращаает значение null, то есть, грубо говоря, фактически отсутствие значения. И чтобы отразить эту ситуацию мы определяем переменную name, в которую получаем ввод с консоли, как переменную типа string?. Здесь string указывает, что переменная может хранить значения типа string, то есть строки. А знак вопроса ? указывает, что переменная также может хранить значение null, то есть по сути не иметь никакого значения. Далее мы более подробно разберем null и как с ним работать.

Однако, может возникнуть вопрос, как нам быть, если, допустим, мы хотим ввести возраст в переменную типа int или другую информацию в переменные типа double или decimal? По умолчанию платформа .NET предоставляет ряд методов, которые позволяют преобразовать различные значения к типам int, double и т.д. Некоторые из этих методов:

Convert.ToInt32() (преобразует к типу int)

Convert.ToDouble() (преобразует к типу double)

Convert.ToDecimal() (преобразует к типу decimal)

Пример ввода значений:

Console.Write("Введите имя: ");

string? name = Console.ReadLine();

Console.Write("Введите возраст: ");

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

Console.Write("Введите рост: ");

double height = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите размер зарплаты: ");

decimal salary = Convert.ToDecimal(Console.ReadLine());

Console.WriteLine($"Имя: {name} Возраст: {age} Рост: {height}м Зарплата: {salary}$")