**Методичка C# Урок 2: Целые Числа.**

**Операции с целыми числами**

В прошлом уроке мы изучали переменные и типы данных, а также узнали, как тип данных влияет на поведение операций. В этом уроке мы познакомимся с операциями над целыми числами.

**В этом уроке вы:**

1. Познакомитесь с двумя типами операций с целыми числами.

2. Узнаете, что такое бинарные операторы и как с ними работать.

3. Поймете, как в коде выставлять приоритет операций.

4. Изучите особенности использования унарных операторов.

5. Узнаете, как изменять переменную через составное присваивание.

6. Потренируете навык вывода целых чисел, решая практические задачи.

**Типы операций**

Все операции делятся на два типа:

- Унарные: операции с одним числом.

- Бинарные: операции с двумя числами.

Пример: В выражении `2 + 3 + 5` операция «+» является бинарной, так как участвуют два числа.

**Бинарные операторы**

Вот список бинарных арифметических операций:

1. Сложение: `a + b`

2. Вычитание: `a - b`

3. Умножение: `a \* b`

4. Деление:

- Целочисленное деление: `a / b`

- Остаток от деления: `a % b`

Операции `+`, `-`, `\*` работают так же, как в математике. Рассмотрим программу:

```csharp

static void Main()

{

int a = 5;

int b = 2;

Console.WriteLine(a + b); // 7

Console.WriteLine(a - b); // 3

Console.WriteLine(a \* b); // 10

}

```

**Деление в C#**

Деление в C# бывает двух видов:

1. Целочисленное деление: `a / b` (отбрасывает дробную часть).

2. Остаток от деления: `a % b` (возвращает остаток).

Пример целочисленного деления:

```csharp

static void Main()

{

Console.WriteLine(12 / 3); // 4

Console.WriteLine(12 / 5); // 2

Console.WriteLine(12 / 7); // 1

}

```

**Запомните:** Результат целочисленного деления меньшего числа на большее будет равен 0.

**Остаток от деления**

Оператор `%` возвращает остаток от деления двух целых чисел. Рассмотрим примеры:

```csharp

static void Main()

{

Console.WriteLine(12 % 3); // 0

Console.WriteLine(12 % 5); // 2

Console.WriteLine(12 % 14); // 12

}

```

**Запомните:** Остаток от деления `n % m` при `n < m` всегда равен `n`.

**Приоритет операций**

В C# порядок выполнения операций аналогичен математике. Рассмотрим код:

```csharp

static void Main()

{

int num1 = 2 + 2 \* 2; // 6

Console.WriteLine(num1);

}

```

**Запомните:** Сначала выполняется умножение или деление, затем сложение и вычитание. Для изменения порядка операций используйте скобки.

**Унарные операторы**

Унарные арифметические операторы:

- Инкрементация: `++` (увеличивает на 1).

- Декрементация: `--` (уменьшает на 1).

Пример:

```csharp

int a = 0;

int b = 5;

a++; // 1

b--; // 4

```

**Составное присваивание**

Составное присваивание позволяет изменять значение переменной, используя более компактный синтаксис.

Вот несколько примеров:

```csharp

int a = 10;

a += 5; // a = a + 5; (15)

a -= 3; // a = a - 3; (12)

a \*= 5; // a = a \* 5; (60)

a /= 3; // a = a / 3; (20)

a %= 6; // a = a % 6; (2)

```

**Запомните:** Составное присваивание помогает избежать дублирования имени переменной.

Давайте еще раз приведем таблицу сокращения составного присваивания:

| **Полное составное присваивание** | **Сокращенное присваивание** |
| --- | --- |
| a = a + 5; | a += 5; |
| a = a - 5; | a -= 5; |
| a = a \* 5; | a \*= 5; |
| a = a / 5; | a /= 5; |
| a = a % 5; | a %= 5; |

**Примечания:**

1. Минус может использоваться для обозначения отрицательных чисел. Однако операции `+`, `\*`, `/` и `%` всегда требуют два операнда.

```csharp

int num1 = -6; // Унарный минус

int num2 = 17 - 7; // Бинарный минус

```

1. Если хотите добавить результат математического выражения к тексту, оберните его в круглые скобки:

```csharp

static void Main()

{

int number = 5;

Console.WriteLine("Число на 1 больше чем 5 равно " + (number + 1)); // Число на 1 больше чем 5 равно 6

}

```

Без скобок:

```csharp

static void Main()

{

int number = 5;

Console.WriteLine("Число на 1 большее, чем 5 равно " + number + 1); // Число на 1 больше чем 5 равно 51

}

```

**Задачки:**

Теперь давайте проверим ваши знания с помощью задачек!

1. \*\*Что покажет приведенный ниже фрагмент кода?\*\*

```csharp

int s = 0;

int k = 30;

int d = k - 5;

k = 2 \* d;

s = k - 50;

Console.WriteLine(s);

```

Ответ: 0

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int s = 0;

int k = 30;

int d = k / 4;

k = 2 \* d;

s = k / 3;

Console.WriteLine(s);

```

\*\*Ответ\*\*: 4

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int s = 0;

int k = 30;

int d = k / 4;

k = 5 \* d;

s = k % 3;

Console.WriteLine(s);

```

Ответ: 2

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int rub = 30;

int kop = 40;

int count = 4;

int oneCost = rub \* 100 + kop;

int totalCost = oneCost \* count;

Console.WriteLine(totalCost / 100);

```

Ответ: 121

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int rub = 30;

int kop = 40;

int count = 4;

int oneCost = rub \* 100 + kop;

int totalCost = oneCost \* count;

Console.WriteLine(totalCost % 100);

```

Ответ: 60

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int rub = 30;

int kop = 40;

int count = 1;

int oneCost = rub \* 100 + kop;

int totalCost = oneCost \* count;

Console.WriteLine(totalCost % 100);

```

Ответ: 40

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int a = 10;

a -= 3;

a \*= 5;

a /= 3;

Console.WriteLine(a);

```

Ответ: 11

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int a = 2;

int b = 3;

Console.WriteLine("a + b = " + a + b);

```

Выберите один вариант из списка:

* 1. a + b = 3
  2. Будет ошибка
  3. a + b = 32
  4. a + b = 23
  5. a + b = 2
  6. a + b = 5

Ответ: 4) `a + b = 23`

1. Что покажет приведенный ниже фрагмент кода?

```csharp

int a = 2;

int b = 3;

Console.WriteLine("a + b = " + (a + b));

```

Выберите один вариант из списка

1. a + b = 23
2. Будет ошибка
3. a + b = 3
4. a + b = 5
5. a + b = 2
6. a + b = 32

Ответ: 4) `a + b = 5`

Эти задачи помогут вам закрепить знания о составном присваивании и операциях с переменными в C#. Если у вас есть дополнительные вопросы или вам нужна помощь с другими темами, не стесняйтесь спрашивать! 😊