Арифметические операторы

№ урока: 12 **Курс:** Процедурное программирование на языке C#

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Данный урок поможет вам понять то, как используются бинарные и унарные арифметические операторы в коде. Вы рассмотрите вопросы приоритета различных операторов, понятие и виды инкремента и декремента, а также понятие «нечисла».

Изучив материал данного занятия, учащийся сможет:

- Понимать и уметь работать с бинарными и унарными математическими операторами.
- Понимать порядок и последовательность вычислений с использованием инкремента и декремента.
- Понимать, что такое «нечисло» и какие действия могут привести к появлению результата вычисления в виде «нечисла».

Содержание урока

1. Бинарные и унарные операторы
2. Инкремент и декремент
3. «Нечисла»
4. Составное присвоение
5. Приоритеты операторов

Резюме

- **Арифметические операторы** — это знаки действий, которые необходимо выполнить над числовыми значениями.
  Арифметические операторы бывают двух видов: унарные и бинарные.
- **Бинарные арифметические операторы** — это операторы умножения, деления, получения остатка от деления, сложения и вычитания. Они работают с двумя операндами, а именно с левым операндом и с правым операндом.
- **Операнд** – это то, над чем производится операция.
- **Унарный** (единственный, однокомпонентный) **арифметический оператор** — это такой оператор, который работает только с одним операндом.
- **Оператор инкремента** – это унарный оператор. **Инкремент** – переводится как приращение.
- **Оператор инкремента (++)** увеличивает свой операнд на 1. Оператор инкремента может находиться как перед операндом, так и после него: ++variable или variable++.
- **Префиксная операция инкремента** – результатом выполнения этой операции является использование значения операнда после его увеличения. ++variable
- **Постфиксная операция инкремента** – результатом выполнения этой операции является использование значения операнда перед его увеличением. variable++.
- **Оператор декремента (--)** уменьшает свой операнд на 1. Оператор декремента может находиться как перед операндом, так и после него: --variable или variable--.
- **Префиксная операция декремента** – результатом выполнения этой операции является использования значения операнда после его декремента. --variable

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Постфиксная операция декремента** – результатом этой операции является использование значения операнда до его декремента. variable--
- **NaN**, расшифровывается как Not-a-Number, и переводится как «нечисло» - одно из особых состояний числа с плавающей запятой. Используется во многих математических библиотеках и математических сопроцессорах. Такое состояние может возникнуть в определенных случаях, например, когда предыдущая математическая операция завершилась с неопределённым результатом или если в ячейку памяти попало не удовлетворяющее условиям число.
- К операциям, приводящим к появлению NaN в качестве ответа, относятся:
    - все математические операции, содержащие NaN в качестве одного из операндов;
    - деление нуля на ноль;
    - деление бесконечности на бесконечность;
    - умножение нуля на бесконечность;
    - сложение бесконечности с бесконечностью противоположного знака;
    - вычисление квадратного корня отрицательного числа;
    - логарифмирование отрицательного числа.
- Для вычисления степени числа можно использовать метод **Math**.Pow(double x, double y). При этом x - число, возводимое в степень, а y – показатель степени.
- Для вычисления квадратных корней - метод S-Q-R-T (расшифровывается Square root - квадратный корень) - **Math**.Sqrt(double x);
- Для вычисления логарифмов - метод **Math**.Log(double x, double b). x - число, логарифм которого требуется найти, b - основание логарифма.
- **Операция составного присваивания** состоит из простой операции присваивания, скомбинированной с какой-либо другой бинарной арифметической операцией. При составном присваивании вначале выполняется действие, описанное бинарной операцией, а затем результат присваивается левому операнду. Выражение составного присваивания со сложением, например имеет вид: <выражение1> += <выражение2>.
- Префиксный инкремент и префиксный декремент, несмотря на то что это, по сути операции сложения и вычитания единицы, имеют более высокий приоритет, чем операции умножения и деления.
- Постфиксный инкремент и декремент, имеют самый низкий приоритет, даже ниже, чем операции сложения и вычитания.
- Инкремент и декремент нужны для удобства представления циклических конструкций, но никак, не для использования их в арифметических выражениях. Поэтому, лучше просто, прибавить единицу привычным способом.

Закрепление материала

- Что такое инкремент и декремент?
- Что такое операция составного присвоения?
- Для чего предназначен инкремент и декремент?
- Что такое «нечисло»?
- Какие операции могут привести к получению «нечисла»?

Самостоятельная деятельность учащегося

- Задание 1.

  Ознакомьтесь с дополнительными материалами к уроку.

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Задание 2.

  Имеется 3 переменных типа int x = 10, y = 12, и z = 3;
  Выполните, рассчитайте и выведите на экран результат следующих операций для этих переменных:
  x += y - x++ * z;
  z = --x - y * 5;
  y /= x + 5 % z;
  z = x++ + y * 5;
  x = y - x++ * z.
  Внимательно ознакомьтесь с материалом по последней ссылке в рекомендуемых ресурсах.

Рекомендуемые ресурсы

Арифметические операторы (справочник по C#)
https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/arithmetic-operators

Оператор инкремента ++
https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/arithmetic-operators#increment-operator-

Оператор декремента
https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/arithmetic-operators#decrement-operator---

Not-a-Number, «нечисло»
https://ru.wikipedia.org/wiki/NaN

Приоритет и ассоциативность операторов
https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/arithmetic-operators#operator-precedence-and-associativity

x += x++ ВЫ НЕ ДОЛЖНЫ ПИСАТЬ ТАКОЙ КОД!
https://habr.com/ru/post/188888/

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com