Машинная математика. Переменные и типы данных.

№ урока: 2 **Курс:** C# Starter

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение двоичной и шестнадцатеричной систем счисления.
Рассмотрение понятия переменной и типов данных.

Изучив материал данного занятия, учащийся сможет:

- Понимать двоичную и шестнадцатеричную системы счисления.
- Понимать, когда и какие типы использовать при создании переменной.
- Понимать отличие между вещественными и десятичными типами.

Содержание урока

1. Обзор устройства оперативной памяти ОЗУ (RAM). Адресация памяти.
2. Обзор позиционных, непозиционных, смешанных и алфавитных систем счисления.
3. Детальное рассмотрение двоичной и шестнадцатеричной систем счисления.
4. Понятие знака для двоичного числа. Двоичное дополнение.
5. Представление данных в ОЗУ с использованием двоичной и шестнадцатеричной систем счисления.
6. Представление бита, тетрады, байта, машинного слова, двойного машинного слова, учетверенного машинного слова, параграфа памяти.
7. Килобайт, мегабайт, гигабайт, терабайт.
8. Понятие вещественного числа. Точность числа.
9. Роль модуля FPU в обработке чисел с плавающей точкой.
10. Рассмотрение примера: создание переменной в C#
11. Правила именования локальных переменных в .Net (Code convention).
12. Рассмотрение примера: простые типы и их псевдонимы.
13. Рассмотрение примера: целые типы, вещественные типы, десятичный тип, логический тип, символьный и строковой типы. Суффиксы. Диапазоны допустимых значений чисел со знаком и без знака.
14. Рассмотрение примера: Значения по умолчанию для локальных переменных.

Резюме

- **ОЗУ (Оперативное запоминающее устройство)** – RAM (Random Access Memory, память с произвольным доступом) – энергозависимая память в которой хранятся данные и команды необходимые процессору для выполнения им операций.
- **CPU (Central processing unit – ЦПУ, центральное обрабатывающее устройство)** – исполнитель машинных инструкций (кода программ).
- **ALU (Arithmetic and logic unit – АЛУ, Арифметико-логическое устройство)** – блок процессора, который служит для выполнения арифметических и логических преобразований над данными.
- **Система счисления** – символический метод записи чисел.
- **Позиционная система счисления (позиционная нумерация)** – система счисления, в которой значение каждого числового знака (цифры) в записи числа зависит от его позиции (разряда).
- **Двоичная система счисления** – это позиционная система счисления с основанием 2. В этой системе счисления числа записываются с помощью двух символов (0 и 1).

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- **Шестнадцатеричная система счисления** – позиционная система счисления по целочисленному основанию 16. Обычно в качестве шестнадцатеричных цифр используются десятичные цифры от 0 до 9 и латинские буквы от A до F, то есть (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).
- Дополнительный код (two's complement, или twos-complement) – способ представления знаковых (положительных и отрицательных) целых чисел.
- **Бит (Bit - binary digit)** – единица измерения информации – один двоичный разряд в двоичной системе счисления. Впервые слово bit, было использовано Клодом Шенноном для логарифмической единицы информации в 1948 г. В вычислительной технике, слово «бит» часто применяется в значении «двоичный разряд».
- **Тетрада** (от греч. tetrás, родительный падеж tetrádos – четвёрка), совокупность 4 бит.
- **Байт (byte)** – единица хранения и обработки цифровой информации. В настольных вычислительных системах байт считается равным восьми битам, в этом случае он может принимать одно из 256 (2^8) различных значений.
- **Разрядность процессора** – способность одновременно обрабатывать какое-то количество бит. Часто, разрядностью компьютера называют разрядность его машинного слова.
- **Машинное слово** – машинно-зависимая и платформенно зависимая величина, измеряемая в битах или байтах (тритах или трайтах – машина Сетунь-70), равная разрядности регистров процессора и/или разрядности шины данных.
- Для 32-битных процессоров x86: исторически машинным словом считается 16 бит, реально – 32 бита. Это правило распространяется на двойные слова (32 бита – 64 бита), учетверенные слова (64 бита – 128 бит) и параграф (128 бит – 256 бит).
- Килобайт (KB): 1 KB = 1024B= 2^{10} B, где B - байт
- Мегабайт (MB): 1 MB = 1024 KB=1024^2B= 2^{20} B= 1048576 B
- Гигабайт (GB): 1 GB = 1024 MB=1024^3B= 2^{30} B= 1 073 741 824 B
- Терабайт (TB): 1 TB = 1024 GB=1024^4B= 2^{40} B = 1 099 511 627 776 B
- **Вещественное число или действительное число** (от лат. realis – действительный) – в информатике – тип данных, содержащий числа, записанные с десятичной точкой и/или с десятичным порядком.
- **Плавающая запятая** – форма представления действительных чисел, в которой число хранится в форме мантиссы и показателя степени. При этом число с плавающей запятой имеет фиксированную относительную точность и изменяющуюся абсолютную. Наиболее часто используемое представление утверждено в стандарте IEEE 754. Реализация математических операций с числами с плавающей запятой в вычислительных системах может быть, как аппаратная, так и программная.
- **Число одинарной точности (англ. Single precision)** – компьютерный формат представления чисел, занимающий в памяти одну ячейку (машинное слово; в случае 32-битного компьютера – 32 бита или 4 байта). Как правило, обозначает формат числа с плавающей точкой стандарта IEEE 754.Числа одинарной точности с плавающей точкой обеспечивают относительную точность 7-8 десятичных цифр и масштабы в диапазоне от 10^{-38} до примерно 10^{38}. Числа одинарной/двойной/расширенной точности (32, 64 и 80 бит) поддерживаются на аппаратном уровне сопроцессором (FPU).
- **Число двойной точности (англ. Double precision)** – компьютерный формат представления чисел, занимающий в памяти две последовательных ячейки (компьютерных слова; в случае 32-битного компьютера – 64 бита или 8 байт). Как правило, обозначает формат числа с плавающей запятой стандарта IEEE 754. Числа двойной точности с плавающей точкой обеспечивают относительную точность около 16 десятичных цифр и масштабы в диапазоне от 10^{-308} до примерно 10^{308}. В компьютерах, которые имеют 64-разрядные с плавающей точкой арифметические единицы, большинство численных вычислений осуществляется в двойной точности с плавающей точкой, поскольку использование чисел одинарной точности обеспечивает почти такую же производительность.
- **Математический сопроцессор** – сопроцессор для расширения командного множества центрального процессора и обеспечивающий его функциональностью модуля операций с плавающей запятой, для процессоров, не имеющих интегрированного модуля.
- **Модуль операций с плавающей запятой** (или с плавающей точкой; англ. floating point unit (FPU)) – часть процессора для выполнения математических операций над вещественными числами.

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- **Переменная (Variable)** – это область памяти, которая хранит в себе некоторое значение, которое можно изменить.
- **Инициализация переменной** – это первое присвоение ей значения. Все последующие присвоения новых значений этой переменной, не считаются инициализацией.
- **Правило**: **При создании переменной обязательно указать ее тип, а при дальнейшем ее использовании, тип указывать не нужно.**
- В языке C# нет базовых типов – все типы реализуются как классы библиотеки NET Framework, но в нем есть набор встроенных типов, которые рассматриваются как **псевдонимы (алиасы–aliases,** короткие и упрощенные формы записи**)** типов из пространства имен System.
- **Псевдоним** – альтернативный тип, который можно использовать вместо системных типов. Например, `decimal` – это псевдоним типа `Decimal`, а тип `int` – псевдоним типа `Int32`.
- В случае использования псевдонимов, как типов для переменных можно не использовать подключение пространства имен `System` (т.е. не писать `using System;`)
- **C# является строго типизированным языком**. Каждая переменная должна иметь четко определенный тип.
- При создании переменной, используйте название-псевдоним, когда это возможно, а не полное имя типа.
- Джеффри Рихтер никогда не использует псевдонимов типов, считая, что их использование приводит к запутанному коду (это не рекомендация, а мнение Джеффри).
- **Идентификатор** – последовательность символов, которые используется для именования членов, таких как переменные, методы, параметры, а также множество
  других программных конструкций, которые будут рассмотрены позже. Иными словами, **идентификатор** переменной – **это имя** этой переменной.
- Спецификация языка C# рекомендует придерживаться определенных правил (casing conventions) при создании идентификаторов (выбора имен для ваших переменных, методов и т.д).
- К таким правилам можно отнести:
  Pascal casing – каждое слово в идентификаторе начинается с большой буквы;
  Camel casing – каждое слово, исключая первое, в идентификаторе начинается с большой буквы;
  Uppercase – идентификатор состоит из букв, написанных в верхнем регистре (все буквы большие)
- Технически, имена переменных могут начинаться со знака «_» – нижнее подчеркивание и с любого алфавитного символа. (Имена не могут начинаться с цифр и других символов.)
- Для именования локальных переменных в C#, рекомендуется использовать **соглашение Camel Casing**. Чтобы выделить слова в идентификаторе, первые буквы каждого слова (кроме первого) сделайте заглавными. Например: `myAge`, `myName`.
- Язык C# чувствительный к регистру (case sensitivity). Например, `MyName` и `myName` – это разные имена.
- Не используйте символы подчеркивания, дефисы и любые другие неалфавитно-цифровые символы для разделения слов в идентификаторе.
- Не используйте венгерскую нотацию. Суть **венгерской нотации** сводится к тому, что имена идентификаторов предваряются заранее оговорёнными префиксами, состоящими из одного или нескольких символов. Например: `string sClientName; int iSize;`
- Имена переменных должны быть понятны и передавать смысл каждого элемента.
- В редких случаях, если у идентификатора нет точного семантического значения, используйте общие названия. Например: `value`, `item`.
- Если вы используете псевдонимы для переменных, и ваше приложение будет выполняться с применением других .NET поддерживаемых языков, желательно ограничиться лишь CLS-совместимыми типами, к которым можно отнести все типы за исключением беззнаковых целочисленных типов и `sbyte`.
- Символы Юникода – это 16-разрядные символы, которые используются для представления большинства известных письменных языков мира.

Закрепление материала

- Что такое переменная?

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- Где и для чего используются переменные?
- Назовите основные типы данных.
- Какие типы данных подходят для хранения значений чисел с плавающей запятой?
- В каком формате должны задаваться значения для строковых переменных?

Дополнительное задание

Задание
Используя Visual Studio, создайте проект по шаблону Windows Forms. Разместите на форме 8 кнопок с названиями целых типов. В обработчиках события нажатия на каждую кнопку напишите следующий код – `MessageBox.Show("сюда можно вписать текст");` и выполните программу.
После успешного выполнения программы, удалите строки – `"сюда можно вписать текст"` из каждого обработчика и вместо них укажите диапазон допустимых значений для соответствующих типов. Запустите приложение.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные акронимы и понятия, рассмотренные на уроке.
Выучите все типы данных, рассмотренные на уроке и диапазоны значений типов: `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`.
Запомните значения по умолчанию для всех простых типов данных.

Задание 2
Используя Visual Studio, создайте проект по шаблону Console Application.
Дано значение числа **pi**, которое равно 3,141592653 и значение числа Эйлера **e**, которое равно 2,7182818284590452. Создайте две переменные, присвойте им значения числа **pi** и числа **e** и выведите их на экран без потери точности.

Задание 3
Используя Visual Studio, создайте проект по шаблону Console Application.
Создайте три строковые переменные и присвойте им значения:

```
"\nмоя строка 1"
"\tмоя строка 2"
"\aмоя строка 3"
```

Выведите значение каждой переменной на экран. Какие отличия вы увидели. Сделайте выводы.

Задание 4
Зайдите на сайт MSDN.
Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Типы значений (Справочник C#)
http://msdn.microsoft.com/ru-ru/library/s1ax56ch.aspx

MSDN: Переменные и константы.
http://msdn.microsoft.com/ru-ru/library/wew5ytx4.aspx

MSDN: Справочные таблицы по типам (Справочник по C#)
http://msdn.microsoft.com/ru-ru/library/1dhd7f2x.aspx

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com