

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

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

На этом уроке будет рассмотрено понятие инфраструктуры .NET и ее составные части. Вы рассмотрите такую классификацию языков программирования как деление их на компилируемые, интерпретируемые и байт-код языки, познакомитесь с понятиями «виртуальная машина» и «управляемый код», а также рассмотрите за счет чего организуется кроссплатформенность .NET Core.

Изучив материал данного занятия, учащийся сможет:

- Понимать суть понятия «инфраструктура .NET».
- Понимать особенности и различия компилируемых, интерпретируемых и байт-код языков.
- Понимать смысл понятия «виртуальная машина» вообще и в применении к платформе .NET.

Содержание урока

1. Компилируемые, интерпретируемые и байт-код языки
2. Виртуальная машина
3. FCL
4. Управляемый код

Резюме

- Инфраструктура .NET = CLR + FCL.
- **Компилируемые языки** работают так:
 - Разработчик пишет программу в виде текстового файла с исходным кодом программы.
 - Файл с написанной программистом программой передается специальной программе, которая называется компилятор.
 - Компилятор берет исходный код программы, который программист писал английскими буквами и преобразовывает этот исходный код в машинный код. Машинный код представляет собой числа, которые являются инструкциями для центрального процессора. Соответственно, компилятор в результате своей работы, генерирует файл с расширением EXE, в котором находятся машинные инструкции для центрального процессора.
 - Когда пользователь дважды кликнет по исполняемому файлу, который имеет расширение EXE, то содержимое этого файла, при помощи

операционной системы, загружается в оперативную память и процессор начинает читать и выполнять те инструкции, которые находятся в этом файле.

Преимуществом кода на компилируемом языке является скорость работы.
Недостатком – зависимость от аппаратной платформы.

- **Интерпретируемые языки.**

- Файл с исходным кодом передается специальной программе, которая называется интерпретатор.
- Интерпретатор не переводит сразу всю программу в машинные коды понятные процессору и не генерирует исполняемого файла. Интерпретатор, берёт из исходного кода программы, по одной отдельной строке, переводит эту отдельную строчку на машинный язык.
- Далее интерпретатор, уже переведенную в машинный код строчку передает процессору, для выполнения.
- Затем берет следующую строку. И так далее. Интерпретатор, занимается переводом исходного кода программы на машинный язык маленькими порциями.

Главное **преимущество** интерпретируемого принципа работы программы в том, что все такие программы имеют открытый исходный код и в них можно быстро вносить изменения. После внесенных изменений не надо эти программы перекомпилировать. **Недостаток** заключается в том, что сам процесс интерпретации требует затрат процессорных ресурсов.

- **Виртуальная машина** — программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы. В упрощенном понимании виртуальная машина — это программа, которая анализирует байт-код, сгенерированный компилятором CSC.EXE и генерирует на основе этого анализа машинный код для процессора.
- **CLR** (Common Language Runtime — общезыковая исполняющая среда) — виртуальная машина, исполняющая среда для байт-кода, полученного в результате компиляции из языков VB, C#, MC++ , CIL.
- **Байт-код** — стандартное промежуточное представление программы, в которое может быть переведена компьютерная программа автоматическими средствами. По сравнению с исходным кодом, удобным для создания и чтения человеком, байт-код — это компактное представление программы, уже прошедшей этап компиляции. В нём в явном виде закодированы типы, области видимости и другие конструкции. С технической точки зрения байт-код представляет собой машинно-независимый код низкого уровня, сгенерированный из исходного кода.
- **Байт-код языки**, к которым относится и язык C#. Такие языки еще называют интерпретируемо-компилируемые.

- Файл с исходным кодом написанной программистом программы передается специальной программе-компилятору. В нашем случае, это компилятор языка C#.
- Компилятор языка C#, переводит исходный код программы, не на машинный язык, а в специальный байт-код. Этот байт-код, очень похож на машинный код, но центральный процессор, этот байт-код не понимает. Байт-код, компилятор сохраняет в исполняемом файле с расширением EXE. Этот байт-код предназначен для того, чтобы его анализировала специальная программа, которая называется виртуальной машиной.
- Виртуальная машина CLR, делает практически то же, что и обычный интерпретатор. CLR, читает исполняемый файл с байт-кодом, выбирает из этого файла кусочки этого байт-кода, интерпретирует (то есть переводит его в машинный код) и передает полученный машинный код, центральному процессору для выполнения.

Преимущество байт-код языков над компилируемыми языками в большей эффективности и портируемости, то есть один и тот же байт-код может исполняться на разных платформах и архитектурах, для которых реализован интерпретатор.

Преимуществом над интерпретируемыми языками является то, что байт-код обычно менее абстрактен и более компактен, чем исходный код. Эффективность интерпретации байт-кода обычно выше, чем чистая интерпретация исходного кода. Кроме того, интерпретатор байт-кода зачастую проще интерпретатора исходного кода и его проще перенести (портировать) на другую аппаратную платформу.

- Имя файла-компилятора - csc.exe.
- **FCL** или BCL. Base Class Library, или так называемая .NET FCL (полностью - Framework Class Library), сокращённо BCL — стандартная библиотека классов платформы «.NET Framework». FCL или BCL – это набор тех классов, которые разработаны заранее компанией Microsoft, чтобы разработчикам не приходилось каждый раз заново создавать часто-используемую функциональность.
- Файл FCL именуется mscorlib.dll.
- Указание **Native** в имени вызываемого метода обозначает, что внутри метода имеется не байт-код, а настоящий машинный код, который выполняет процессор. Можно сказать, что многие классы, которые входят в FCL – это просто удобные, высокоуровневые обертки над так называемым API операционной системы Windows.
- **API** расшифровывается как application programming interfaces - интерфейсы программирования приложений.
- **Windows API** – это большой набор разных функций, написанных на компилируемых языках. Именно эти функции выполняют всю полезную работу по взаимодействию с компьютерным железом.

- **Управляемый код** (по-английски. managed code) — термин, введенный фирмой Microsoft, для обозначения кода программы, исполняемой под управлением виртуальной машины .NET — Common Language Runtime или Mono. При этом машинный код (который выполняет центральный процессор) называется неуправляемым кодом (по-английски unmanaged code).
- Байт-код для виртуальной машины называется управляемым кодом, а машинный код для процессора, – неуправляемый код.

Закрепление материала

- Что такое инфраструктура .NET?
- В чем отличие компилируемых, интерпретируемых и байт-код языков?
- Что такое виртуальная машина?
- С кодами, скомпилированными из каких языков может работать CLR?
- Что такое байт-код?
- К какой платформе относится C#?

Самостоятельная деятельность учащегося

- Задание

Ознакомьтесь с дополнительными материалами к уроку.

Рекомендуемые ресурсы

Обзор книги "Инфраструктура программных проектов» Кржиштоф Цвалина, Брэд Абрамс

<https://itvdn.com/ru/channel/video/infrastructure-software-projects>

https://ru.wikipedia.org/wiki/Управляемый_код

https://ru.wikipedia.org/wiki/Windows_API

https://ru.wikipedia.org/wiki/Виртуальная_машина

<https://ru.wikipedia.org/wiki/Байт-код>