Visão geral conceitual do .NET Framework e C#

Objetivo

- Apresentar os elementos fundamentais da linguagem C#
- Enfatizar conceitos comuns ao *framework* Microsoft . NET

.NET Framework

- O .NET Framework é um componente essencial do Windows que oferece suporte à criação e execução da próxima geração de aplicativos e serviços XML da Web.
- Uma característica central da Plataforma .NET é aderir aos padrões da Internet sem abrir mão de procedimentos já consagrados no Windows.
- Para isso conta com o Visual Studio.NET, suíte de programação definida pela Microsoft como "especialmente voltada para a rápida construção e integração de Web Services".
- O produto incorpora as linguagens Visual Basic, Visual C++ e Visual C# ("CSharp), todas com sobrenome .NET. Linguagens tradicionais, as duas primeiras sofreram ajustes para a nova plataforma, enquanto o C# começa do zero.

- O .NET Framework foi criado para atender os seguintes objetivos:
  - Para fornecer um ambiente de programação orientada a objetos consistente, se o código objeto for armazenado e executado localmente, mas distribuído pela Internet ou executado remotamente.
  - Para fornecer um ambiente da execução de código que minimiza conflitos de implantação e versionamento de software.
  - Para fornecer um ambiente da execução que promova a execução segura do código, incluindo o código criado por terceiros: desconhecidos ou semi-confiáveis.
  - Para fornecer um ambiente de execução que elimina os problemas de desempenho dos ambientes interpretados ou com scripts.
  - Para tornar a experiência do desenvolvedor consistente, através dos diversos tipos de aplicativos, como aplicativos baseados no Windows e aplicativos baseados na Web.
  - Para criar todas as comunicações nas indústrias padrão, para garantir que códigos baseados no .NET Framework possam se integrar a qualquer outro código.

.NET Framework

- possui dois componentes principais:
  - a *common language runtime* (CLR)
  - a biblioteca de classes do .NET Framework

CLR

- é o alicerce do .NET Framework
- agente que dirige o código no tempo de execução, fornecendo serviços principais como gerenciamento de memória, gerenciamento de segmento e arquitetura de comunicação remota, enquanto forçam, também, a segurança de tipos estritos e outras formas de precisão de código que promovem segurança e robustez
- o conceito de gerenciamento de código é um princípio fundamental do Runtime.
- o código que visa o Runtime é conhecido como código gerenciado, enquanto o código que não visa o Runtime é conhecido como código não gerenciado.

Biblioteca de classes

- é uma coleção orientada a objeto extensa de tipos reutilizáveis, que você pode usar para desenvolver aplicativos, desde os tradicionais por linha de comando (CLI) ou aplicativos por interface gráfica (GUI), até aplicativos com base nas inovações mais recentes fornecidas pelo ASP.NET, como Web Forms e Serviços XML da Web

A CLR é capaz de executar, atualmente, mais de 20 diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem.

- Estas são:
- APL
- Boo
- Clarion
- COBOL
- Component Pascal
- C#
- C++
- Eiffel
- Forth
- Fortran
- Haskell
- Java
- JScript
- J#
- Lua
- Mercury
- Mondrian
- Oberon
- Object Pascal / Delphi Language
- Oz
- Pascal
- Perl
- PowerBuilder
- PowerShell
- Python
- RPG
- Ruby
- Scheme
- SmallTalk
- Standard ML
- Visual Basic

Visual Studio .NET

- Três pacotes diferentes:
- Enterprise Architect: mais completo e inclui, além das três linguagens, ferramentas para depuração e modelagem, desenvolvimento em grupos e todos os servidores do Windows
- Enterprise Developer: mais simples, não tem, por exemplo, os recursos de modelagem. Mais voltada para o programador individual
- Professional: não traz servidores nem itens de trabalho em grupo

Termos da Plataforma

- **MSIL** - Microsoft Intermediate Language. Quando se compila uma aplicação .NET, ela é convertida para uma linguagem intermediária, a MSIL, um conjunto de instruções independentes de CPU. Na hora de executar o programa, um novo compilador, chamado Just-in-time (JIT) Compiler, o converte para o código nativo, ou seja, específico para o processador da máquina.
- **IDE COMPARTILHADO** - Ambiente integrado de programação (Integrated Development Environment) do Visual Studio.NET. Diferentes linguagens usam o mesmo editor de código e depurador e compilam executáveis na linguagem MSIL.

C# Language

- C# é uma elegante e segura linguagem orientada à objeto que permite aos desenvolvedores criar uma variedade de aplicativos seguros e robustos que são executados no .NET Framework
- Pode ser usado para criar aplicações cliente tradicional do Windows, XML Web Services, componentes distribuídos, cliente-servidor de aplicações, bases de dados, aplicativos, etc.

C# Language

- Criada junto com a arquitetura . NET, foi criada praticamente do zero para funcionar na nova plataforma
- A maior parte das classes do .NET Framework foram desenvolvidas em C#

C# Language

- A sintaxe do C # será instantaneamente reconhecível para alguém familiarizado com C, C ++ ou Java
- Gestão automática de memória (*garbage collection)*
  - Elimina problemas com fugas de memória e apontadores inválidos
  - Mas possibilita trabalhar diretamente com apontadores

Primeiro programa

-

A cláusula using define quais bibliotecas de classe o programa irá usar.

Classe padrão criada na criação de um projeto que identifica o ponto inicial de execução de uma aplicação

```
using System;
class HelloWorld
{
   static void main() {
        Console.WriteLine(“Hello, world”);
   }
}
```

Comando que permite escrever informações no Console de saída do programa, que nesse caso é o monitor por padrão

“Hello, World em C#”