Linguagens de Programação

Introdução

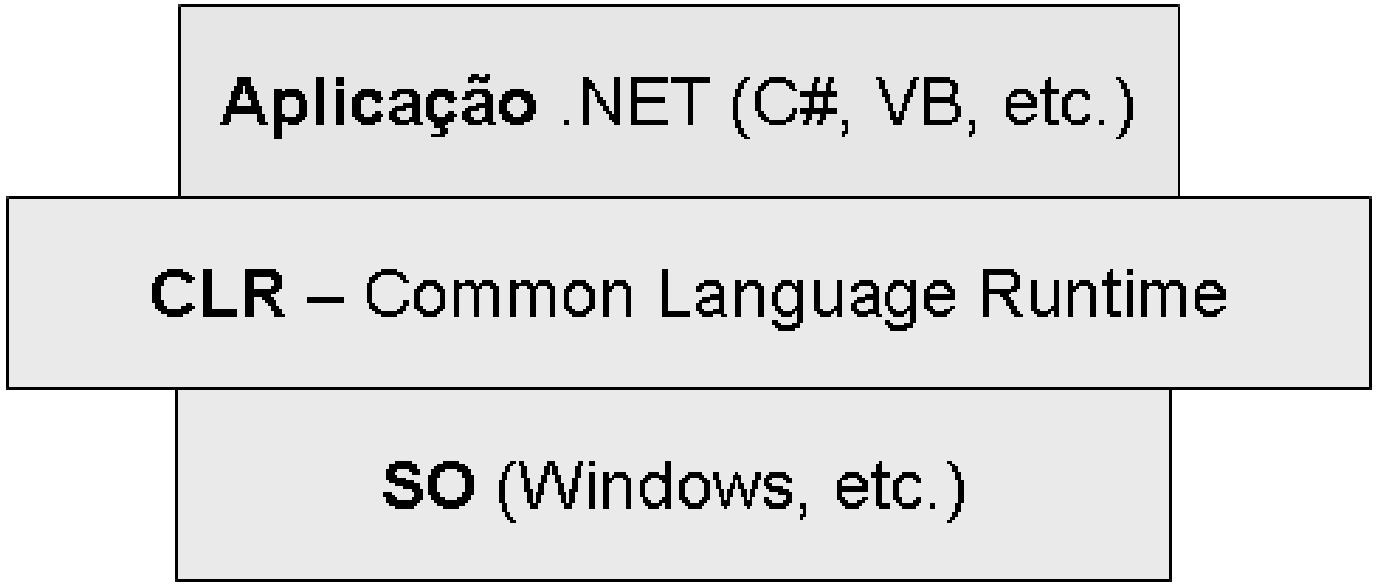
José Martins
Escola Superior de Tecnologia
Instituto Politécnico do Cávado e do Ave
jmartins@ipca.pt

Adaptado de: António Araújo

Linguagem C#

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- Linguagem de Programação Orientada a Objectos
- Foi criada para ser suportada pela arquitectura .NET



Linguagem C#

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- Um programa em C# tem diferentes fases:
- O código da aplicação é compilado e convertido para MSIL (Microsoft Intermediate Language)
- Depois uma estrutura denominada por CLR (Common Language Runtime) gera as intruções máquina de forma a que o processador as execute

Linguagem C#

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- Para criar um programa em C# usaremos um ambiente de desenvolvimento:
- Visual Studio Community: https://www.visualstudio.com/downloads/
- Ambiente de Trabalho com muitas funcionalidades e complexidade
- Ao trabalhar com este programa convém arranjar um método de trabalho que nos permita “andar” com os nossos trabalhos independentemente do nosso local de trabalho (casa, escola, ...)

Exemplo de um programa em C#

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

```
using System;
namespace Teste{
   class Exemplo
   {
      public static void Main(String[] args)
      {
         Console.WriteLine("Hello World!");
      }
   }
}
```

Explicação C#

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

using System

- indica que o espaço de nomes (System) vai ser usado pelo programa, o que implica de todas as definições contidas neste espaço (módulo) vão ser importadas, ou antes, podem ser usadas no programa
- O espaço de nomes System define classes de uso comum (e.g. Console)

namespace Teste

- namespace é usado para evitar o conflito de nomes. Um conjunto de classes relacionadas deve ser colocado no mesmo namespace

Explicação C#

class Exemplo

- conjunto de dados e métodos que descrevem uma entidade. Num programa C# deve existir pelo menos uma classe que contém um método Main()

Main()

- Primeiro método a utilizar na execução na execução do programa. Num mesmo namespace podem existir mais do que um método Main()

Documentar o código

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- É importante documentar o código que se cria
- Ajuda a que outros programadores percebam para que serve um determinado pedaço de código
- Ajuda a perceber o propósito dos componentes de uma classe (variáveis, métodos, ...)
- Ajuda a organizar as ideias sobre um programa e permite identificar melhor os elementos de um problema

Comentários

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- O Visual Studio permite documentar o código num ficheiro à parte em XML
- Usa-se “//” para iniciar o comentário
- Existem várias tags que permitem organizar melhor os comentários
- ///<summary>
- ///<example>
- ///<code>
- Ou ainda várias linhas com /* …… */

Variáveis

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- Uma variável é uma área de memória, identificada por um nome, que pode armazenar valores de um determinado tipo
- O tipo da variável permite determinar os valores que esta pode conter e as operações em que pode participar
- C# é uma linguagem fortemente tipada (todas as variáveis têm obrigatoriamente um tipo)

Variáveis

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- Tipos numéricos:

Int

Float

Double

Tipo	Faixa de Valores	Tamanho
sbyte	-128 até 127	8 bits
byte (sem sinal)	0 até 255	8 bits
char (unicode)	U+0000 até U+ffff	16 bits
short	-32.768 até 32.767	16 bits
ushort (sem sinal)	0 até 65.535	16 bits
int	-2.147.483.648 até 2.147.483.647	32 bits
uint (sem sinal)	0 até 4.294.967.295	32 bits
long	-9.223.372.036.854.775.808 até 9.223.372.036.854.775.807	64 bits
ulong (sem sinal)	0 até 18.446.744.073.709.551.615	64 bits

Tipo	Faixa de Valores	Precisão	Tamanho
float	$1.5x10^{-45}$ até $3.4x10^{38}$	7 dígitos	32 bits
double	5.0×10^{-324} até 1.7×10^{308}	15-16 dígitos	64 bits

Tipo	Faixa de Valores	Precisão	Tamanho
decimal	$1.0x10^{-28}$ até $7.9x10^{28}$	28 dígitos	128 bits

Variáveis

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Tipo lógico ou Boolean

- O tipo boolean é usado para operações lógicas
- Os valores possíveis são true e false

Tipo char

- representa caracteres

Tipo string

- representa uma sequência de caracteres

Declaração e Atribuição

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

A declaração de uma variável serve para indicar o tipo de dados associado a essa variável

- int x; //esta declaração expressa que x é um inteiro

Atribuição é quando se associa um valor a uma variável

- int x = 10; // a variável x passa a ter o valor 10
- o símbolo "=" é o operador de atribuição
- uma atribuição também pode ser feita com o mesmo valor

  x = x + 5; //à variável x são acrescentados 5 unidades

Constantes

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Uma constante é uma variável mas cujo conteúdo não pode ser alterado

- const int x = 10; //isto quer dizer que x terá sempre o valor 10

As variáveis por seu lado podem mudar de valor sempre que o programador entender

- int x = 10; //primeiro x tem o valor 10
- x = 20; //depois passa a ter o valor 20

Expressões e Operadores

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Uma expressão é uma combinação de operandos e operadores:

- int a, b, c;
- a = (b + c) / c;

Nota:

- a, b e c são os operandos
- + e / são os operadores
- Parênteses são usados para alterar a ordem de operações (tal como numa calculadora)

Expressões e Operadores

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Categoria	Operadores	Associatividade
Unário	`+ - !`	esquerda
Multiplicativo	`* / %`	esquerda
Aditivo	`+ -`	esquerda
Relacional	`< = > >= <= == is`	esquerda
Igualdade	`== !=`	esquerda
'and' (bool)	`&&`	esquerda
'ou' (bool)	`||`	esquerda

Expressões e Operadores

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Operador	Significado
=	Atribuição simples
+=	Atribuição com adição
-=	Atribuição com subtração
*=	Atribuição c/ multiplicação
/=	Atribuição c/ divisão
%=	Atribuição do resto

Expressões e Operadores

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Operador	Significado	Exemplo de uso
>	Maior que	a>b, 4>3, c>3
<	Menor que	2*a>b*2
!=	Diferente de	a+b!=c
==	Igual a	a==0
<=	Menor ou igual a	b*a<=0
>=	Maior ou igual a	a>=b

Expressões de Controlo

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

- **Expressões condicionais**
- **Expressões de seleção alternativa**
- **Expressões de repetição com base numa condição**
- **Expressões de repetição com base num contador**

Expressões de Controlo

Expressões condicionais (IF)

o bloco de instruções é executado se a condição lógica for verdadeira

```
        if(<condição>)
{
        <bloco de instruções>
}
```

Expressões de Controlo

Expressões condicionais (IF-THEN-ELSE)

se a condição lógica for verdadeiro é executado o primeiro bloco de instruções, se não é o segundo

```
    if(<condição>){
    <primeiro bloco de instruções>
}
    else{
    <segundo bloco de instruções>
}
```

Expressões de Controlo

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Expressões de seleção alternativa (switch)

É executado um bloco de instruções de acordo com o valor de uma variável

```
switch(<variável>)
{
        case <valor1>: <bloco 1>
                                        break;
        case <valor2>: <bloco 2>
                                        break;
}
```

Expressões de Controlo

Expressões de repetição com base numa condição (while)

É executado um bloco de instruções de acordo com uma condição desde que essa condição se mantenha verdadeira

```
while(<condição>)
{
        <bloco de instruções>
}
```

Expressões de Controlo

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Expressões de repetição com base numa condição (do-while)

Ao contrário do while, onde a condição é testada primeiro, aqui, executa-se primeiro o código e só depois é que se testa se é para continuar ou não:

```
do
{
        <bloco de instruções>
}
while(<condição>);
```

Expressões de Controlo

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Expressões de repetição com base num contador (for)

- É executado um bloco de instruções de acordo com o incremento de uma variável enquanto a condição for verdadeira
- Primeiro são executadas as operações sobre a variável de controlo e só depois é executado o bloco de instruções

```
for(<inicialização>; <condição>; <atualização>)
{
    <bloco de instruções>
}
```

Expressões de Controlo

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Quebras de Ciclo - break

- Em qualquer bloco de instruções numa expressão de repetição, o comando break pode ser usado para terminar o ciclo
- O uso deste tipo de controlo é pouco aconselhado devido à má estruturação do código a que ele leva

Interatividade

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Escrever na consola:

Console.Write/WriteLine(<string>, [params])

Console.Write(“escreve na consola”)

Console.WriteLine(“escreve e muda de linha”)

```
int x = 2;
string nome = “abc”;
Console.Write(“Hello World”); //simples escrita de string
Console.Write(“valor:” + x + “palavra: ” + nome)
```

Interatividade

ESCOLA TÉCNICA SUPERIOR PROFISSIONAL IPCA

Leitura de valores da consola

```
string linha = Console.ReadLine();
int x = Int.Parse(linha);
string nome = “abc” + linha;
```