IFRN
PROGRAMAÇÃO BÁSICA EM C#

Decisão

Prof. Gilbert Azevedo

Objetivos

- Utilizar as estruturas de controle de fluxo: if, if – else, switch
- Utilizar variáveis lógicas, operadores relacionais e lógicos na escrita de programas

Estruturas de Controle de Fluxo

- Estruturas de controle de fluxo
  - São estruturas utilizadas para controlar o fluxo de execução dos comandos em um algoritmo ou programa
- Estruturas condicionais
  - Permitem controlar a execução ou não de um comando ou bloco de comandos
- Estruturas de repetição
  - Permitem controlar a repetição de um comando ou bloco de comandos

Estruturas Condicionais

- Alternativa Simples (se, if)
  - Executa ou não um comando (ou bloco de comandos) de acordo com um teste realizado
- Alternativa Dupla (se – senão, if – else)
  - Executa um ou outro comando (ou bloco de comandos) de acordo com um teste realizado
- Alternativa Múltipla (caso, switch)
  - Executa um comando (ou bloco de comandos) de acordo com um valor específico de uma variável

Valores Lógicos

- As estruturas condicionais utilizam valores lógicos nos testes para determinar a execução dos comandos
  - Constantes: Verdadeiro (true) e Falso (false)
  - Variáveis Lógicas
  - Expressões com Operadores Relacionais e Lógicos

Variáveis Lógicas

- Variáveis lógicas ou booleanas são definidas com o tipo *bool*
  - A variável bool armazena os valores Verdadeiro (true) ou Falso (false)
  - O resultado de uma expressão com operadores relacionais e lógicos é um valor lógico.
  - O operador de igualdade (==), por exemplo, testa se dois valores são iguais.
  - `int i = 10;`
  - `bool a = true;   // a recebe true`
  - `bool b = i == 5; // b recebe falso`

Operadores Relacionais

- São operadores que comparam dois valores de tipos compatíveis e tem como resultado um valor lógico

Operação	Algoritmo	C#	Tipos
Igualdade	=	==	I,R,C,L,S
Diferença	<>	!=	I,R,C,L,S
Maior ou igual	>=	>=	I,R,C
Maior	>	>	I,R,C
Menor ou igual	<=	<=	I,R,C
Menor	<	<	I,R,C

Operadores Relacionais

- Valores inteiros, reais e caracteres podem ser comparados entre si
  - `int a = 1, b = 2; char c = '*'; double d = 5;`
  - `bool r1 = a == 1;` `// true`
  - `bool r2 = a == b;` `// false`
  - `bool r3 = b == c;` `// false`
  - `bool r4 = c == d;` `// false`

Operadores Relacionais

- Textos e lógicos podem ser comparados apenas com valores do mesmo tipo
  - `double d = 5;`
  - `string e = "teste"; bool f = false;`
  - `// bool r5 = d == e;    // ERRO`
  - `bool r6 = e == "TESTE"; // false`
  - `bool r7 = f == false;   // true`
  - `bool r8 = e != "TESTE"; // true`

Operadores Lógicos

- São operadores que realizam as operações lógicas de conjunção, disjunção (binários) e negação (unário)

Operação	Algoritmo	C#	Tipos
Conjunção	e	&&	L
Disjunção	ou	||	L
Negação	não	!	L

X	Y	X e Y
F	F	F
F	V	F
V	F	F
V	V	V

X	Y	X ou Y
F	F	F
F	V	V
V	F	V
V	V	V

X	não X
F	V
V	F

Operadores Lógicos

- Testar se um valor inteiro x está entre 1 e 10
  - `int x = 7;`
  - `bool r9 = 1 <= x && x <= 10;`
- Testar se foi aprovado em uma disciplina (média e frequência)
  - `int md = 75; double fr = 0.8;`
  - `bool r10 = md >= 60 && fr >= 0.75;`
- Testar se foi reprovado em uma disciplina
  - `bool r11 = !r10;`
  - `bool r12 = md < 60 || fr < 0.75;`

Precedência de Operadores

- Expressões são resolvidas de acordo com a precedência:
- Operadores unários: mais, menos, negação, incremento, decremento
  - `+ - ! ++ --`
- Aritméticos multiplicativos: multiplicação, divisão, resto
  - `* / %`
- Aritméticos aditivos: soma, subtração
  - `+ -`
- Relacionais: menor (ou igual), maior (ou igual), igualdade e diferença
  - `< <= > >= == !=`
- Lógicos: conjunção e disjunção
  - `&& ||`

Estrutura Condicional IF

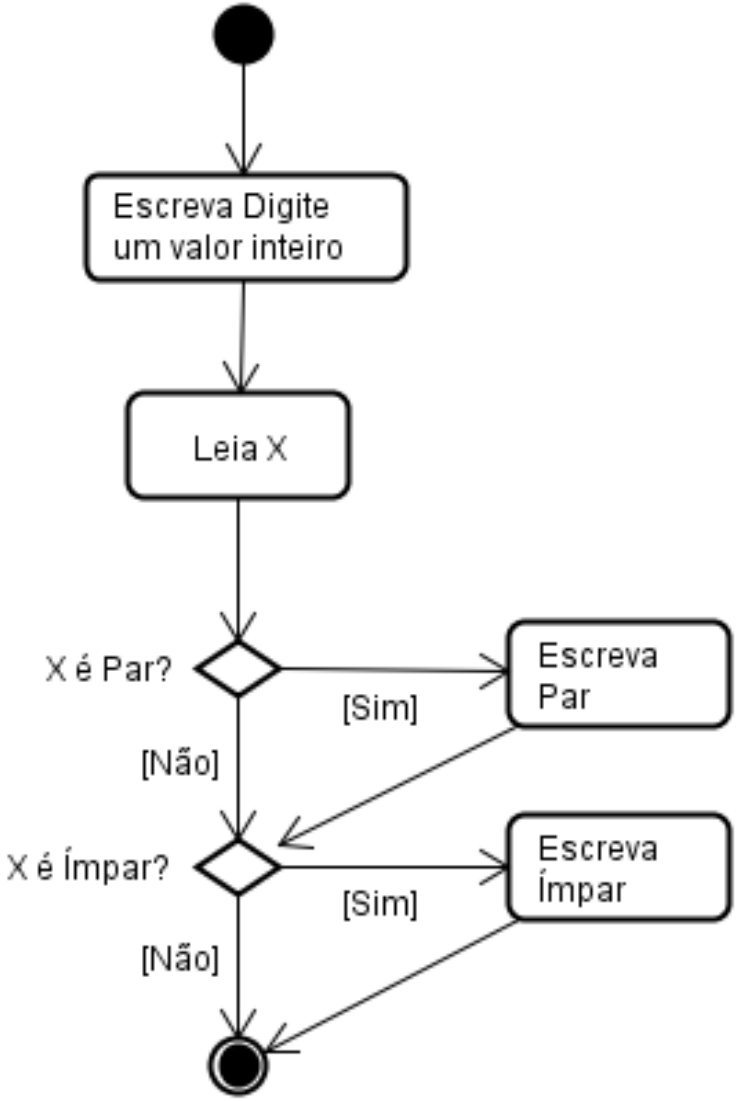
- A estrutura IF é usada para testar se um comando ou um bloco de comandos deve ser realizado
- Se o teste na estrutura for verdadeiro, os comandos são executados; caso contrário, nenhum comando é executado.

Algoritmo	C#
se teste então comando;	if (teste) comando;

se teste então
início
  comandos;
fim;
	if (teste)
	{
	  comandos;
	}

Diagrama do IF

- Algoritmo: Par ou Ímpar?
- Declaração de Variáveis
  - x : inteiro;
- Início
  - Escreva("Digite um valor inteiro");
  - Leia(x);
  - se x Mod 2 = 0 então Escreva("Par");
  - se x Mod 2 = 1 então Escreva("Ímpar");
- Fim.



Exemplo de IF em C#

```
using System;
class MainClass {
  public static void Main (string[] args) {
    Console.WriteLine("Digite um valor inteiro");
    int x = int.Parse(Console.ReadLine());
    if (x % 2 == 0) Console.WriteLine("Par");
    if (x % 2 == 1) Console.WriteLine("Ímpar");
  }
}
```

Estrutura Condicional IF – ELSE

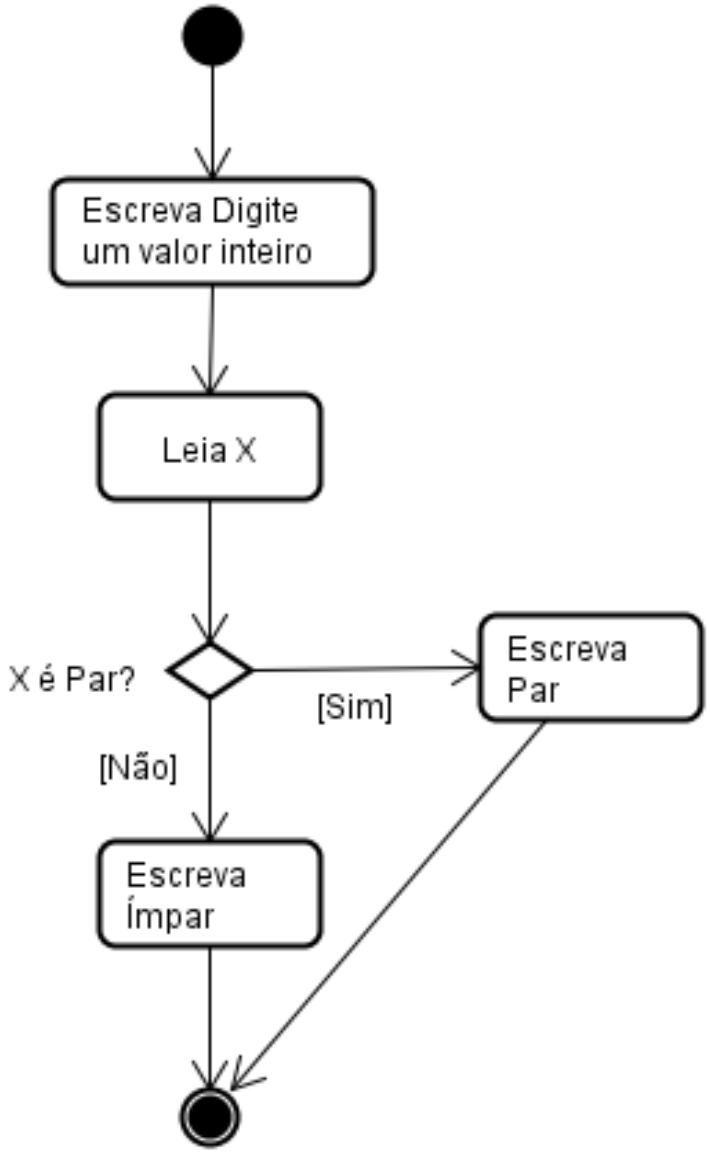
- A estrutura IF – ELSE é usada para realizar um comando ou um bloco de comandos se um teste for verdadeiro e outro comando ou bloco de comandos se o mesmo teste for falso

Algoritmo	C#
se teste então comando1; senão comando2;	if (teste) comando1; else comando2;

se teste então início
    comandos1;
fim;
senão início
    comandos2;
fim;
	if (teste) {
	    comandos1;
	}
	else {
	    comandos2;
	}

Diagrama do IF – ELSE

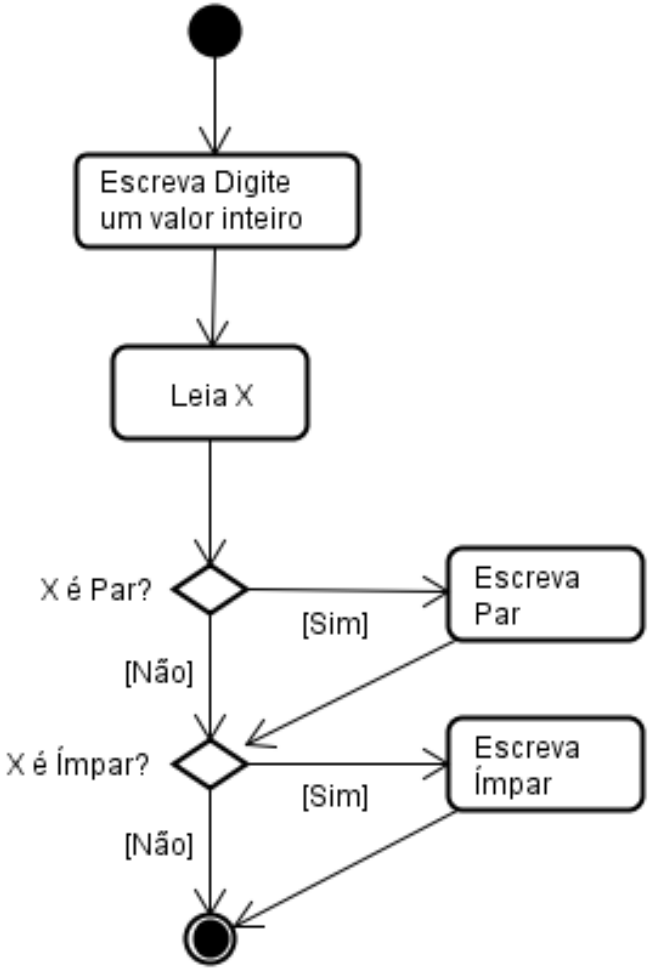
- Algoritmo: Par ou Ímpar?
- Declaração de Variáveis
  - x : inteiro;
- Início
  - Escreva("Digite um valor inteiro");
  - Leia(x);
  - se x Mod 2 = 0
  - então Escreva("Par");
  - senão
  - Escreva("Ímpar");
- Fim.

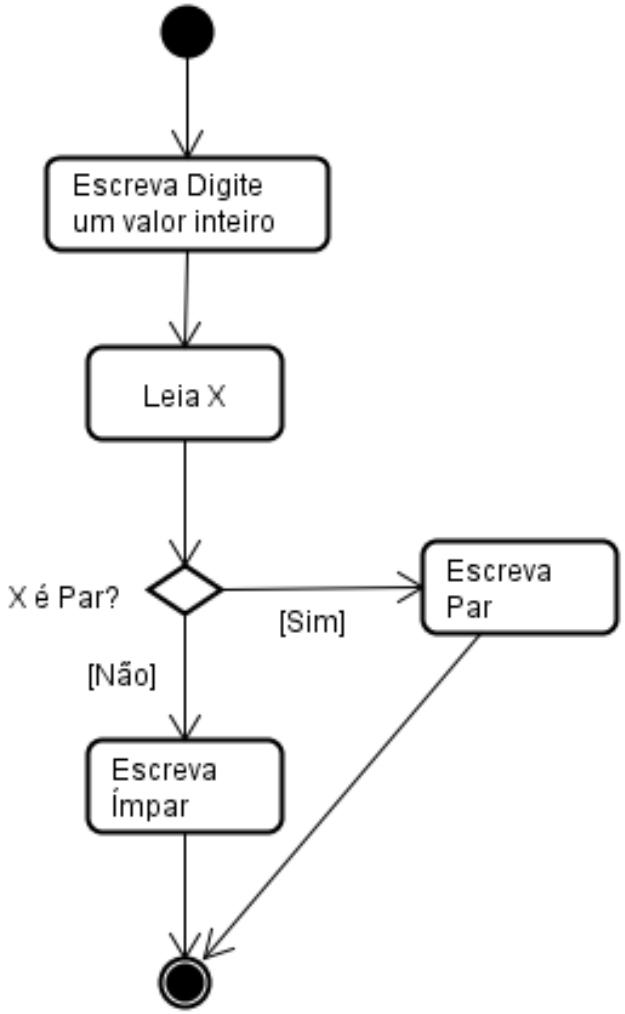


Exemplo de IF – ELSE em C#

```
using System;
class MainClass {
  public static void Main (string[] args) {
    Console.WriteLine("Digite um valor inteiro");
    int x = int.Parse(Console.ReadLine());
    if (x % 2 == 0) Console.WriteLine("Par");
    else Console.WriteLine("Ímpar");
  }
}
```

Comparação: IF e IF – ELSE





Operador Condicional Ternário

- Em alguma situações, o operador condicional ternário (?) pode substituir uma estrutura condicional IF – ELSE
- O operador ? têm três operandos
  - Uma expressão lógica
  - Um valor que é retornado se a expressão for verdadeira
  - Um valor que é retornado se a expressão for false
- Sintaxe
  - resultado = ExpressãoLógica ? valor-1 : valor-2;

Exemplo de Operador ? em C#

```
using System;
class MainClass {
  public static void Main (string[] args) {
    Console.WriteLine("Digite um valor inteiro");
    int x = int.Parse(Console.ReadLine());
    Console.WriteLine(x % 2 == 0 ? "Par" : "Ímpar");
  }
}
```

Estrutura Condicional Switch

- A estrutura Switch é usada para testar vários valores de uma variável e realizar comandos específicos para cada valor
- As variáveis de controle podem ser inteiros, caracteres ou strings
- Os comandos para cada valor devem ser finalizados com um break

Algoritmo	C#

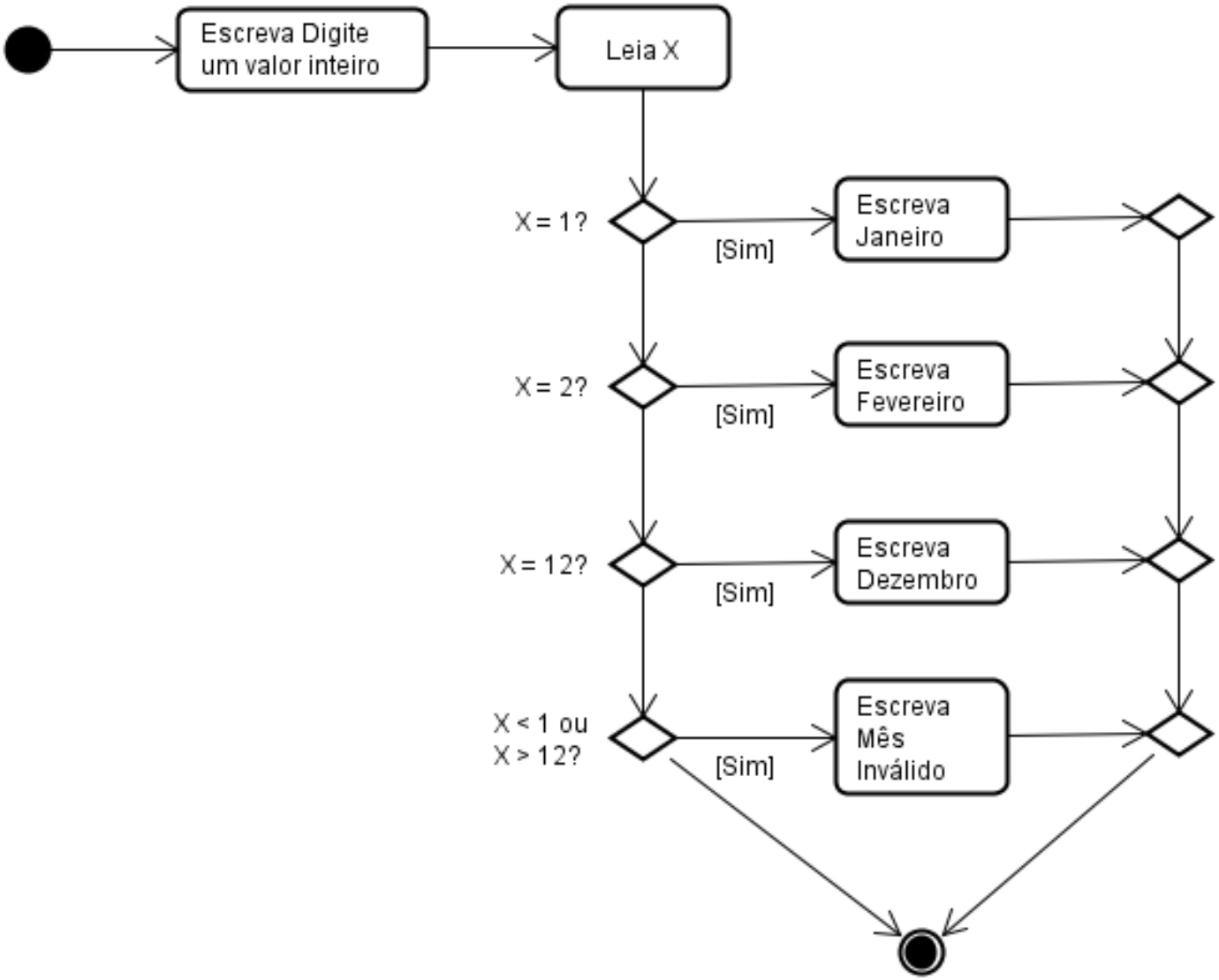
caso Variável seja
  <valor1> : comandos1;
  <valor2> : comandos2;
  ...
  <valorN> : comandosN;
  senão: comandosX;
fim;
	switch (Variável) {
	  case <valor1> : comandos1; break;
	  case <valor2> : comandos2; break;
	  ...
	  case <valorN> : comandosN; break;
	  default : comandosX; break;
	}

Exemplo de Switch

- Algoritmo para ler um nº do mês e mostrar o nome correspondente

- Declaração de Variáveis
  - x : inteiro;
- Início
  - Escreva("Digite o nº do mês");
  - Leia(x);
  - caso x seja
  - 1 : Escreva(“Janeiro”);
  - 2 : Escreva(“Fevereiro”);
  - 3 : Escreva(“Março”);
  - 4 : Escreva(“Abril”);
  - .....
  - 12 : Escreva(“Dezembro”);
  - senão Escreva(“Mês inválido”);
  - fim;
- Fim.

Diagrama do Switch



Exemplo de Switch em C#

```
class MainClass {
  public static void Main (string[] args) {
    Console.WriteLine("Digite um valor inteiro entre 1 e 12");
    int x = int.Parse(Console.ReadLine());
    switch (x) {
      case 1: Console.WriteLine("Janeiro"); break;
      case 2: Console.WriteLine("Fevereiro"); break;
      ........
      case 12: Console.WriteLine("Dezembro"); break;
      default: Console.WriteLine("Mês inválido"); break;
    }
  }
}
```

Referências

- Microsoft Visual C# 2010 – Passo a passo, John Sharp, Bookman, 2010
- Operadores em C#
  - https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/
- Instruções (Guia de Programação em C#)
  - https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/statements-expressions-operators/statements

Fim

- Tarefa
  - Questionário
- Próxima Aula
  - Estruturas de Repetição