**Roteiro de aula – comando FOR (C#)**

**1 - Suponha que você deseja imprimir a palavra “Algoritmo” 5 vezes. Você pode fazer o seguinte programa.**

// Exemplo01

class MainClass {

public static void Main(string[] args) {

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

}

}

O problema de fazer um programa como o exemplo acima, é que cada comando a ser repetido deve ser digitado pelo programador. E se quiséssemos imprimir a palavra “Algoritmo” 1000 vezes? Nesse caso seriam necessários 1000 comandos Console.WriteLine(“Algoritmo”).

// Exemplo02

class MainClass {

public static void Main(string[] args) {

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

// 998 System.out.println's depois

Console.WriteLine("Algoritmo");

Console.WriteLine("Algoritmo");

}

}

Fica evidente, pelo exemplo acima, que o ideal é utilizar um comando que repita os comandos desejados. O comando **for** é um dos comandos da linguagem Java que executa repetições. A principal utilidade do comando **for** é repetir o bloco de comandos interno uma determinada quantidade de vezes. Se utilizarmos o comando **for**, podemos facilmente imprimir a palavra “Algoritmo” quantas vezes desejarmos.

// Exemplo03

class MainClass {

public static void Main(string[] args) {

for(int aux = 1; aux <= 1000; aux++)

Console.WriteLine("Algoritmo");

}

}

No exemplo acima, o comando Console.WriteLine(“Algoritmo”) foi digitado uma única vez. Entretanto, o comando **for** repetirá o comando Console.WriteLine 1000 vezes, conforme definido pelo intervalo de repetição (de 1 ate 1000).

O comando **for** tem a seguinte sintaxe (modo de escrever).

**for ( inicialização ; condição ; atualização ) {**

**// bloco de comandos a serem repetidos**

**}**

**Observações:**

**for ( int aux = valorinicial ; aux <= valorfinal ; aux++ ) {**

**// bloco de comandos a serem repetidos**

**}**

1. Em uma forma mais simples de utilização, usamos o comando **for** para repetir o bloco de comando uma quantidade **pré-determinada** de vezes
2. Para isso usamos uma variável de controle (ou variável auxiliar), que se sugere (nesse caso) ser do tipo int
3. O **<valorinicial>** não precisa ser, necessariamente, o número 1. Pode-se começar a repetição com qualquer outro valor. Ex: ***for (int x = 19; x <= 26; x++)***. Nesse caso serão realizadas 8 repetições.
4. A quantidade de repetições é dada pelo **<valorfinal> - <valorinicial> + 1**. No exemplo acima ***for (int x = 19; x <= 26; x++)***, 26 – 19 + 1 = 8

**Exercício**: faça um programa que imprima seu nome 15 vezes iniciando a variável auxiliar a partir de 1.

**Resposta**

for(int aux = 1; aux <= 15; aux += 1) {

Console.WriteLine(“Jéssica Cristina Araújo Lima de Melo”);

}

**Exercício**: altere o programa que imprima seu nome 15 vezes iniciando a variável auxiliar a partir de 117.

**Resposta**

for(int aux = 117; aux <= 131; aux += 1) {

Console.WriteLine(“Jéssica Cristina Araújo Lima de Melo”);

}

**2 – Embora seja menos comum, é possível utilizar qualquer tipo de variável para controlar a repetição no comando for. Veja dois exemplos em que a variável de controle é do tipo float e double.**

// Exemplo04

class MainClass {

public static void Main(string[] args) {

for(float aux = 1; aux <= 5; aux = aux + 1)

Console.WriteLine("Algoritmo");

}

}

// Exemplo05

class MainClass {

public static void Main(string[] args) {

for(double aux = 1.5; aux <= 3.14; aux = aux + 0.1)

Console.WriteLine("Algoritmo");

}

}

Quantas vezes a palavra “Algoritmo” foi impressa no Exemplo04? E no Exemplo05?

Resposta

No exemplo04 foi impresso 5 vezes a palavra “Algoritmo”.

No exemplo04 foi impresso 17 vezes a palavra “Algoritmo”.

**3 – Digite e execute o programa abaixo:**

// Exemplo06

using System;

class MainClass {

public static void Main(string[] args) {

int numero;

for(int i = 1; i <= 3; i++) {

Console.Write("Digite um numero: ");

numero = int.Parse(Console.ReadLine());

}

}

}

Agora altere o programa acima para que ele fique da seguinte forma:

// Exemplo07

using System;

class MainClass {

public static void Main(string[] args) {

int numero;

for(int i = 1; i <= 3; i++) {

Console.Write("Digite o "+i+"o numero: ");

numero = int.Parse(Console.ReadLine());

}

}

}

Que diferença você notou entre a execução do exemplo06 e exemplo07? De onde vem os números que compõem as mensagens “Digite o 1o numero”?

**Resposta**

A diferença que consegui perceber é que as mensagens dentro do “Console.Write” estão descritas de formas diferentes.

A mensagem “Digite o 1o numero” vem da variável de controle “i”, dentro do comando “for”.

**4 – Além do comando “for” ser utilizado simplesmente para repetir uma quantidade pré-determinada de vezes, uma de suas características mais interessantes é que sua variável de controle (ou variável auxiliar - aquela variável utilizada para contar a quantidade de repetições) também pode ser utilizada para outros fins (cálculos, impressões, etc). Digite e execute os exemplos abaixo.**

// Exemplo08

class MainClass {

public static void Main(string[] args) {

for(int i = 1; i <= 15; i++)

Console.WriteLine(i);

}

}

// Exemplo09

class MainClass {

public static void Main(string[] args) {

for(int i = 1497; i <= 1511; i++)

Console.WriteLine(i);

}

}

Quantos números foram impressos no exemplo08? E no exemplo09? Nos dois exemplos as variáveis auxiliares, além de contarem a quantidade de repetições, também serviram para imprimir os números dentro do intervalo informado (1..15 e 1497..1511).

**Resposta**

Foi impresso no exemplo08 e 09 15 números.

**Exercício:** faça um programa que imprima os números de 1 até 100.

**Resposta**

for (int a = 0; a <= 100; a++)

    Console.WriteLine(a);

**5 – Digite e execute o programa abaixo:**

// Exemplo10

class MainClass {

public static void Main(string[] args) {

for(int i = 1; i <= 100;  **i = i + 7** )

Console.WriteLine(i);

}

}

A alteração no valor do incremento da variável auxiliar (note que usamos **i = i + 7** ao invés de **i++** ou **i = i + 1**) no comando **for** para alterou o seu comportamento? O que você notou?

**Resposta**

Notei que passou a incrementar não apenas de 1 em 1, mas sim de 7 em 7 cada vez que há uma nova iteração.

Ao invés de incremento da variável de controle, podemos usar também o decremento. Veja:

// Exemplo11

class MainClass {

public static void Main(string[] args) {

for(int i = 100; i >= 1; i = i - 5)

Console.WriteLine(i);

}

}

O que está errado no programa abaixo?

// Exemplo12

class MainClass {

public static void Main(string[] args) {

for(int i = 1; i <= 100; i = i - 5)

Console.WriteLine(i);

}

}

**Resposta**

A variável foi declarada com o valor inicial de 1, e o valor final de 100.

Isso gerará um conflito no algoritmo, pois o valor será sempre negativo, gerando assim, valores negativos infinitos, não sairá do loop nunca.

Vamos fazer agora um programa em que você determina o intervalo a ser impresso e o valor do incremento/decremento.

// Exemplo13

using System;

class MainClass {

public static void Main(string[] args) {

// Declaração das variáveis

int vi, vf, p;

// Leitura dos valores a serem usados no comando "for"

Console.Write("Informe o valor inicial do intervalo : ");

vi = int.Parse(Console.ReadLine());

Console.Write("Informe o valor final do intervalo : ");

vf = int.Parse(Console.ReadLine());

Console.Write("Informe o valor do passo : ");

p = int.Parse(Console.ReadLine());

for(int num = vi; num <= vf; num = num + p)

Console.WriteLine(num);

}

}

Agora teste seu programa com os seguintes valores:

a – valor inicial = 13, valor final = 147, passo = 11

b – valor inicial = 1987, valor final = 1, passo = -2

c – valor inicial = 100, valor final =2, passo = 2

Algum desses testes deu errado? Se sim, qual foi o problema que ocasionou o erro?

**Resposta**

**B** e **C** estão erradas, o valor inicial está maior que o valor final, não entrando no for, pois 1 não e <= a 1987, assim como, 2 não é <= a 100.

**6–Agora vamos utilizar a variável de controle/auxiliar do comando “for” para executarmos programas mais interessantes. Digite e execute o programa abaixo:**

// Exemplo14

using System;

class MainClass {

public static void Main(string[] args) {

// Declaração das variáveis

int lado;

double area;

for(int aux = 1; aux <= 5; aux++) {

Console.Write($"Informe o lado do {aux}o quadrado: ");

lado = int.Parse(Console.ReadLine());

area = Math.Pow(lado, 2);

Console.Write($"Lado = {lado} – área = {area}\n\n");

}

}

}

O programa acima calcula a área de 5 quadrados, cujos lados deverão ser informados pelo usuário. Mas imagine que você já sabe que deseja calcular a área dos quadrados com lados = 1, 2, 3, 4 e 5. Podemos fazer a variável auxiliar variar entre 1 e 5 e utilizar os valores dela como o lado do quadrado.

// Exemplo15

class MainClass {

public static void Main(string[] args) {

// Declaração das variáveis

int lado;

double area;

for(int aux = 1; aux <= 5; aux++) {

lado = aux;

area = Math.Pow(lado, 2);

Console.Write($"Lado = {lado} – área = {area}\n\n");

}

}

}

Ou então, fazer a própria variável lado variar de 1 até 5.

// Exemplo16

class MainClass {

public static void Main(string[] args) {

// Declaração das variáveis

double area;

for(int lado = 1; lado <= 5; lado++) {

area = Math.Pow(lado, 2);

Console.Write($"Lado = {lado} – área = {area}\n\n");

}

}

}

**Outro exemplo**. Agora vamos fazer um programa que calcule a tabuada de um determinado número digitado pelo usuário. Por exemplo, a tabuada de 7 é composta pelo número 7 multiplicado pelos números entre 1 e 10. Assim, podemos calcular a tabuada de um número da seguinte forma:

// Exemplo17

using System;

class MainClass {

public static void Main(string[] args) {

int num;

Console.Write("Digite um numero: ");

num = int.Parse(Console.ReadLine());

for(int mult = 1; mult <= 10; mult++) {

int result = num \* mult;

Console.WriteLine(num+" x "+mult+" = "+result);

}

}

}

**Como último exemplo, vamos modificar o programa acima para calcular a tabuada dos números entre 1 e 10. Ou seja, ao invés de ler um número e calcular sua tabuada, vamos calcular a tabuada de todos os números entre 1 e 10.**

// Exemplo18

class MainClass {

public static void Main(string[] args) {

for(int num = 1; num <= 10; num++) {

for(int mult = 1; mult <= 10; mult++) {

int result = num \* mult;

Console.WriteLine(num+" x "+mult+" = "+result);

}

Console.WriteLine();

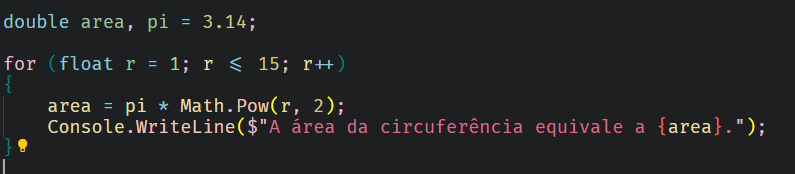
}

}

}

**Exercício**: faça um programa que calcule a área das circunferências com raio variando entre 1 e 15.

**Resposta**



**Exercício**: faça um programa que calcule a área das circunferências com raio variando entre x e y, sendo x e y informados pelo usuário.

**Resposta**

