



CIÊNCIAS DA COMPUTAÇÃO

FACULDADE DE CIÊNCIAS | UNIVERSIDADE AGOSTINHO NETO



FUNDAMENTOS DE PROGRAMAÇÃO

Docente:

✓ Lufialuiso Sampaio Velho, MSc.

Monitor

✓ António Malengue



Cap. IV

Estruturas de Repetição

CAP. IV - Estruturas de Repetição

- ❑ As estruturas de repetição são usadas para executar de forma repetida um conjunto de instruções.
- ❑ Usam condição que define o número de vezes que as instruções serão executadas.
- ❑ Em Java as estruturas de repetição são:
 - `for()`
 - `while()`
 - `do while()`

CAP. IV - Estruturas de Repetição – for()

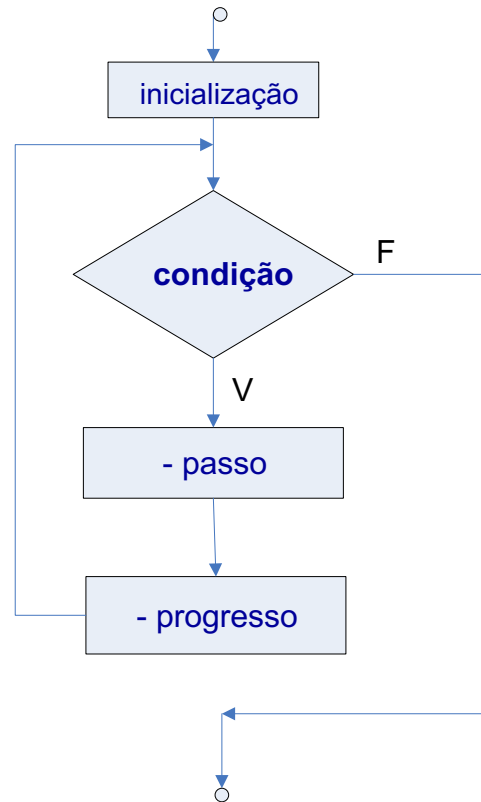
- ❑ A estrutura **for** é utilizada na maioria das vezes quando sabemos o número de vezes que se deseja repetir as instruções.

SINTAXE - JAVA	Pseudocódigo
<pre>for(inicialização; condição; incremento){ <blocos de instruções> }</pre>	<pre>para V de vi ate vf passo p faça <blocos de instruções> fimpara</pre>

- **Inicialização** – expressão de atribuição sobre a variável de controlo que define o valor inicial do ciclo.
- **Condição** – expressão relacional que define o número de iteração do ciclo.
- **Incremento** – define como a variável de controlo do ciclo varia em cada iteração do ciclo.
- As três secções anterior são separadas por pontos-e-vírgulas.

CAP. IV - Estruturas de Repetição – for()

FLUXOGRAMA



CAP. IV - Estruturas de Repetição – for()

- ❑ EX: Programa em Java que imprime a todos números inteiros de 1 à 50.

Resolução - Pseudocódigo	Resolução - JAVA
<pre>inicio Para v de 1 ate 50 passo 1 escreva("\n Numero: ", v) fimpara fimalgoritmo</pre>	<pre>public class Fundamentos { public static void main(String[] args) { for (int i = 1; i <= 50; i++) { System.out.println("Numero: " + i); } } }</pre>

CAP. IV - Estruturas de Repetição – while()

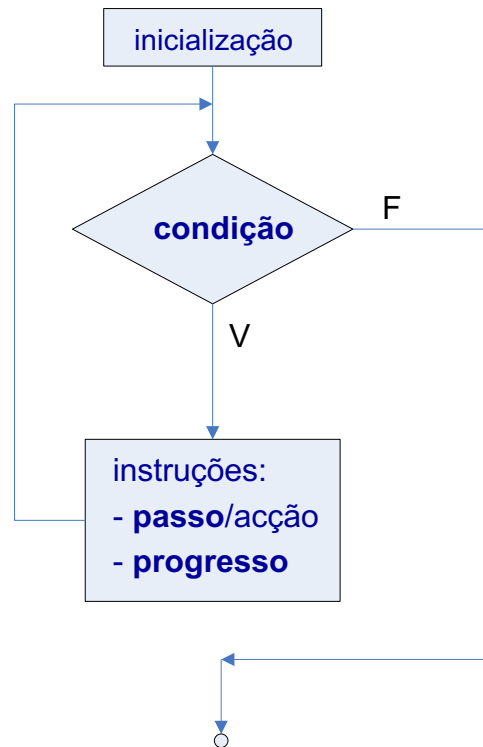
- ❑ **While** é uma estrutura de repetição com verificação no início. Pode ser utilizada quando não sabemos o número de vezes que as instruções serão repetidas, desde que a condição do while seja verdadeira.

SINTAXE - JAVA	Pseudocódigo
<pre><inicialização> while(condição){ <blocos de instruções> <incremento> }</pre>	<pre><inicialização> enquanto (condição) faça <bloco de instruções> <incremento> fimenquanto</pre>

- **Inicialização** – expressão de atribuição sobre a variável de controle que define o valor inicial do ciclo.
- **Condição** – expressão relacional que define a número de iteração do ciclo.
- **Incremento** – define como a variavel de controle do ciclo varia em cada iteração do ciclo.

CAP. IV - Estruturas de Repetição – while()

FLUXOGRAMA



CAP. IV - Estruturas de Repetição – while()

- ❑ EX: Programa em Java que imprime a todos números inteiros de 1 à 50.

Resolução - Pseudocódigo	Resolução - JAVA
<pre>inicio i <- 1 enquanto (i <= 50) faca escreva("\n Numero: ", i) i <- i + 1 fimenquanto Fimalgoritmo</pre>	<pre>public class Fundamentos { public static void main(String[] args) { int i = 1; //inicialização while(i <= 50) { //condição System.out.println("Numero: " + i); i++; //incremento } } }</pre>

CAP. IV - Estruturas de Repetição – do..while()

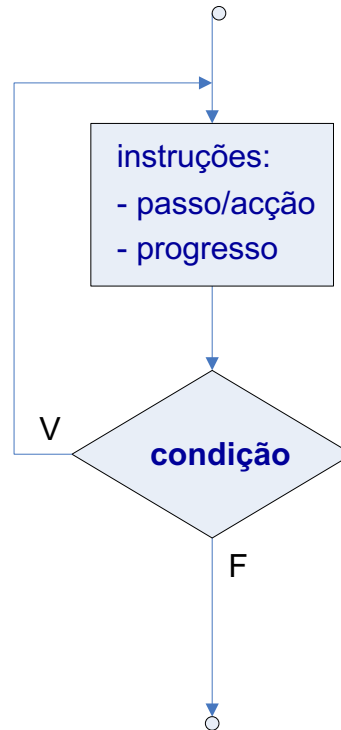
- ❑ **do...while** é uma estrutura de repetição com verificação feita no fim. Ela é utilizada maioritariamente na solução de problemas que executam as instruções pelo menos uma vez. Ex: Criação de um MENU de opções

SINTAXE - JAVA	Pseudocódigo
<pre><inicialização> do { <blocos de instruções> <incremento> }while(condição);</pre>	<pre><inicialização> faca <blocos de instruções> <incremento> Enquanto(condição)</pre>

- **Inicialização** – expressão de atribuição sobre a variável de controlo que define o valor inicial do ciclo.
- **Condição** – expressão relacional que define a número de iteração do ciclo.
- **Incremento** – define como a variavel de controlo do ciclo varia em cada iteração do ciclo.

CAP. IV - Estruturas de Repetição – do..while()

FLUXOGRAMA



CAP. IV - Estruturas de Repetição – do..while()

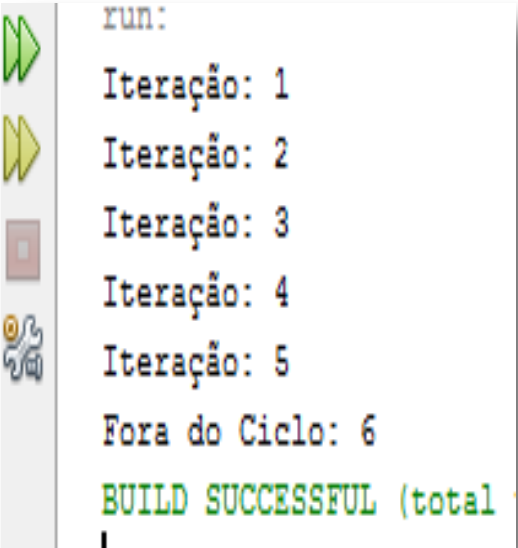
- ❑ EX: Programa em Java que imprime a todos números inteiros de 1 à 50.

Resolução - Pseudocódigo	Resolução - JAVA
<pre>inicio i <- 1 faca escreva("\n Numero: ", i) i <- i + 1 enquanto (i <= 50) Fimalgoritmo</pre>	<pre>public class Fundamentos { public static void main(String[] args) { int i = 1; //inicialização do{ System.out.println("Numero: " + i); i++; //incremento }while(i<=50); //condição } }</pre>

CAP. IV - Estruturas de Repetição

❑ Comando Break


- O comando break é utilizado nas estruturas de repetição e na estrutura switch para ocasionar uma saída imediata desta estrutura. A execução continua na primeira instrução que vem após as estruturas de repetição.

Exemplo	Resultado
<pre>public class Fundamentos { public static void main(String[] args) { int i; for (i = 1; i <= 12; i++) { if (i == 6) break; System.out.println("Iteração: " + i); } System.out.println("Fora do Ciclo: " + i); } }</pre>	

CAP. IV - Estruturas de Repetição

❑ Comando Continue

- O comando continue é utilizado nas estrutura de repetição para ocasionar um salto nas instruções restantes no corpo do ciclo e prosseguir para a próxima iteração do ciclo.

Exemplo	Resultado
<pre>public class Fundamentos { public static void main(String[] args) { int i; for (i = 1; i <= 12; i++) { if (i == 6) continue; System.out.println("Iteração: " + i); } System.out.println("Fora do Ciclo: " + i); } }</pre>	 <pre>run: Iteração: 1 Iteração: 2 Iteração: 3 Iteração: 4 Iteração: 5 Iteração: 7 Iteração: 8 Iteração: 9 Iteração: 10 Iteração: 11 Iteração: 12 Fora do Ciclo: 13 BUILD SUCCESSFUL (total</pre>

CAP. IV - Estruturas de Repetição – Exercícios

1. Crie um programa em Java que recebe uma quantidade de infinita de números inteiros, só termina quando digitar um número Negativo. O programa deverá imprimir a quantidade de números lidos menores que 8.
2. Crie um programa em Java que recebe dois (2) números inteiros e imprime todos os números no intervalo do primeiro e o segundo. Considere o segundo maior do que o primeiro.
3. Crie um programa em Java que imprime a média aritmética de vários valores positivos múltiplos de 2, lidos a partir do teclado. O programa termina de ler caso seja inserido um valor negativo.

Até a próxima Aula

