

INE5603 Introdução à POO

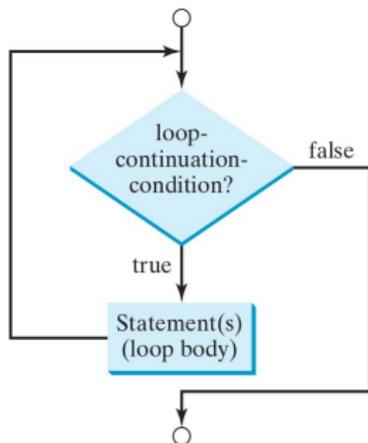
Prof. A. G. Silva

18 de setembro de 2017

Estruturas de repetição

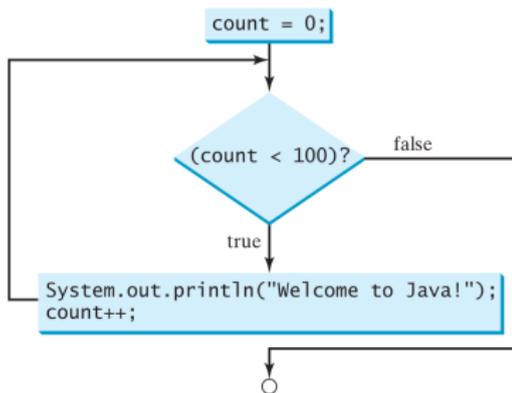
while

while



```
while (loop-continuation-condition) {  
    // Loop body  
    Statement(s);  
}
```

while – exemplo 1



```
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java!");
    count++;
}
```

loop-continuação-condição

corpo do loop

while – exemplo 2

```
int sum = 0, i = 1;
while (i < 10) {
    sum = sum + i;
    i++;
}
System.out.println("sum is " + sum); // sum is 45
```

while – exemplo 3

RepeatAdditionQuiz.java

```
1  import java.util.Scanner;
2
3  public class RepeatAdditionQuiz {
4      public static void main(String[] args) {
5          int number1 = (int)(Math.random() * 10);
6          int number2 = (int)(Math.random() * 10);
7
8          // Create a Scanner
9          Scanner input = new Scanner(System.in);
10
11         System.out.print(
12             "What is " + number1 + " + " + number2 + "? ";
13         int answer = input.nextInt();
14
15         while (number1 + number2 != answer) {
16             System.out.print("Wrong answer. Try again. What is "
17                 + number1 + " + " + number2 + "? ");
18             answer = input.nextInt();
19         }
20
21         System.out.println("You got it!");
22     }
23 }
```

```
What is 5 + 9? 12 ↵Enter
Wrong answer. Try again. What is 5 + 9? 34 ↵Enter
Wrong answer. Try again. What is 5 + 9? 14 ↵Enter
You got it!
```

while – exemplo 4

GuessNumber.java

```
1  import java.util.Scanner;
2
3  public class GuessNumber {
4      public static void main(String[] args) {
5          // Generate a random number to be guessed
6          int number = (int)(Math.random() * 101);
7
8          Scanner input = new Scanner(System.in);
9          System.out.println("Guess a magic number between 0 and 100");
10
11         int guess = -1;
12         while (guess != number) {
13             // Prompt the user to guess the number
14             System.out.print("\nEnter your guess: ");
15             guess = input.nextInt();
16
17             if (guess == number)
18                 System.out.println("Yes, the number is " + number);
19             else if (guess > number)
20                 System.out.println("Your guess is too high");
21             else
22                 System.out.println("Your guess is too low");
23         } // End of loop
24     }
25 }
```

while – exemplo 5

MaxNumber.java

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

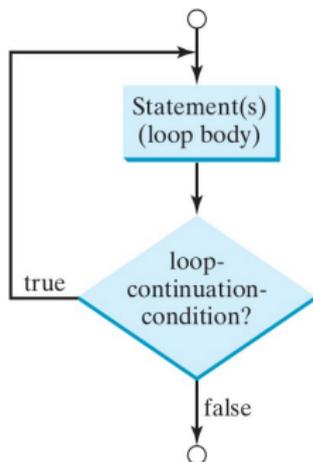
        int number, max;
        number = input.nextInt();
        max = number;

        while (number != 0) {
            number = input.nextInt();
            if (number > max)
                max = number;
        }

        System.out.println("max is " + max);
        System.out.println("number " + number);
    }
}
```

do-while

do-while



```
do {  
    // Loop body;  
    Statement(s);  
} while (loop-continuation-condition);
```

do-while – exemplo 1

TestDowhile.java

```
1 import java.util.Scanner;
2
3 public class TestDowhile {
4     /** Main method */
5     public static void main(String[] args) {
6         int data;
7         int sum = 0;
8
9         // Create a Scanner
10        Scanner input = new Scanner(System.in);
11
12        // Keep reading data until the input is 0
13        do {
14            // Read the next data
15            System.out.print(
16                "Enter an integer (the input ends if it is 0): ";
17            data = input.nextInt();
18
19            sum += data;
20        } while (data != 0);
21
22        System.out.println("The sum is " + sum);
23    }
24 }
```

```
Enter an integer (the input ends if it is 0): 3 --> Enter
Enter an integer (the input ends if it is 0): 5 --> Enter
Enter an integer (the input ends if it is 0): 6 --> Enter
Enter an integer (the input ends if it is 0): 0 --> Enter
The sum is 14
```

for

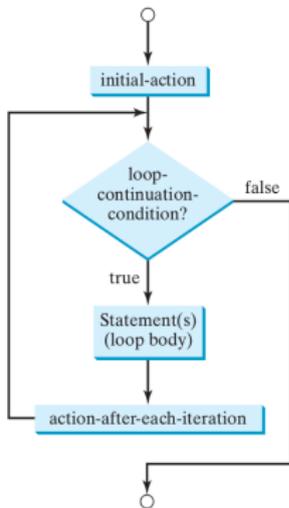
for

```
i = initialValue; // Initialize loop control variable
while (i < endValue)
    // Loop body
    ...
    i++; // Adjust loop control variable
}
```



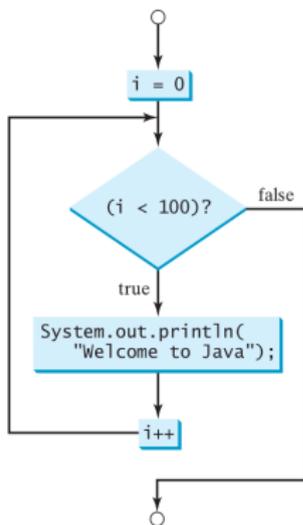
```
for (i = initialValue; i < endValue; i++)
    // Loop body
    ...
}
```

for



```
for (initial-action; loop-continuation-condition;  
      action-after-each-iteration) {  
    // Loop body;  
    Statement(s);  
}
```

for



```
int i;  
for (i = 0; i < 100; i++) {  
    System.out.println("Welcome to Java!");  
}
```

for

```
for ( ; ; ) {  
    // Do something  
}
```

Equivalent

```
for ( ; true; ) {  
    // Do something  
}
```

Equivalent

```
while (true) {  
    // Do something  
}
```

```
while (loop-continuation-condition) {  
    // Loop body  
}
```

Equivalent

```
for ( ; loop-continuation-condition; ) {  
    // Loop body  
}
```

```
for (initial-action;  
     loop-continuation-condition;  
     action-after-each-iteration) {  
    // Loop body;  
}
```

Equivalent

```
initial-action;  
while (loop-continuation-condition) {  
    // Loop body;  
    action-after-each-iteration;  
}
```

loops – cuidados

Error

```
for (int i = 0; i < 10; i++);  
{  
    System.out.println("i is " + i);  
}
```

Empty body

```
for (int i = 0; i < 10; i++) { };  
{  
    System.out.println("i is " + i);  
}
```

Error

```
int i = 0;  
while (i < 10);  
{  
    System.out.println("i is " + i);  
    i++;  
}
```

Empty body

```
int i = 0;  
while (i < 10) { };  
{  
    System.out.println("i is " + i);  
    i++;  
}
```

```
int i = 0;  
do {  
    System.out.println("i is " + i);  
    i++;  
} while (i < 10);
```

Correct

loops aninhados

```
public class Test {  
    public static void main(String[] args) {  
        for (int i = 1; i < 5; i++) {  
            int j = 0;  
            while (j < i) {  
                System.out.print(j + " ");  
                j++;  
            }  
        }  
    }  
}
```

(a)

```
public class Test {  
    public static void main(String[] args) {  
        int i = 0;  
        while (i < 5) {  
            for (int j = i; j > 1; j--)  
                System.out.print(j + " ");  
            System.out.println("****");  
            i++;  
        }  
    }  
}
```

(b)

```
public class Test {  
    public static void main(String[] args) {  
        int i = 5;  
        while (i >= 1) {  
            int num = 1;  
            for (int j = 1; j <= i; j++) {  
                System.out.print(num + "xxx");  
                num *= 2;  
            }  
  
            System.out.println();  
            i--;  
        }  
    }  
}
```

(c)

```
public class Test {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            int num = 1;  
            for (int j = 1; j <= i; j++) {  
                System.out.print(num + "G");  
                num += 2;  
            }  
  
            System.out.println();  
            i++;  
        } while (i <= 5);  
    }  
}
```

(d)

Comandos de repetição – while

- Exemplo do MDC:

```
public int mdc(int n, int m) {  
    while (n != m)  
        if (n > m)    n = n - m;  
        else          m = m - n;  
}
```

Comandos de repetição – for

- Exemplo do fatorial:

```
public long fornecaFatorial(int n) {  
    long fat = 1;  
    for (int i=1; i <= n; i++)  
        fat = fat * i;  
    return fat;  
}
```

Observações

- Não fornecer, no corpo de uma instrução `while`, uma ação que conseqüentemente faz com que a condição torne-se falsa, normalmente resulta em um erro de lógica chamado **loop infinito** (a repetição pressegue indefinidamente)
- Escolher um valor de sentinela que também seja um valor legítimo de dados é um erro de lógica. Por exemplo: digitar -1 para sair, quando -1 poderia ser um número válido em um somatório
- Muitos programas podem ser logicamente divididos em três fases: uma de inicialização das variáveis; uma de processamento com inserção dos valores dos dados e ajuste das variáveis; e uma de término que calcula e insere os resultados finais

Observações (cont...)

- Ao realizar a divisão por uma expressão cujo valor possa ser zero (p. ex., no cálculo de média, é necessário somar e dividir pela quantidade), é importante testar e tratar o denominador (p. ex., imprimindo uma mensagem de erro) em vez de permitir a ocorrência do erro
- Omitir as chaves que delimitam um bloco pode levar a erros de lógica, como *loops infinitos*. Para evitar esse problema, alguns programadores incluem o corpo de cada instrução de controle entre chaves mesmo que seja uma única instrução
- A experiência tem mostrado que a parte mais difícil de resolver um problema em um computador é desenvolver o algoritmo para a solução. Uma vez que um algoritmo foi especificado, produzir um programa em Java (ou qualquer outra linguagem) que execute tal algoritmo é relativamente simples

Comando break

- O comando **break**, ao ser executado, causa uma saída imediata de um processo de repetição definido por um **while**, **do-while** ou **for**, desviando o fluxo de execução para o primeiro comando após o laço repetitivo.
- Exemplo:

```
int soma = 0;
int n = umaInterface.leiaValor();
for (int cont=2; cont < n; cont++) {
    if (n % cont == 0)
        break;
    soma = soma + cont;
}
```

Comando continue

- O comando **continue**, ao ser executado, causa um salto para o início da próxima iteração, ignorando o restante dos comandos dentro do laço. No exemplo a seguir, quando a variável *i* assume valor 7, hífen são exibidos (7 não é exibido, pois "System.out.println(i);" não é executado neste caso) e o laço inicia com o próximo valor de *i* (ou seja, $i \leftarrow 8$).
- Exemplo:

```
for (int i=0; i<10; i++) {  
    if (i==7) {  
        System.out.println("--");  
        continue;  
    }  
    System.out.println(i);  
}
```

Referência

- Y. Daniel Liang. Introduction to Java Programming – Comprehensive Version. 10th Edition. Pearson. 2015.