

Tempo estimado de leitura:
5 min

Módulo #2

Guias de estudo

1

Introdução

Nesse módulo você aprendeu sobre a sintaxe básica do Java. Você estudou sobre **variáveis**, **tipos primitivos**, **operadores lógicos** e **estruturas de controle**.

Neste guia, você encontrará um pequeno resumo do conteúdo trabalhado no módulo e exemplos que ilustram o que você já aprendeu. Vamos lá?

Variáveis

Variáveis nada mais são do que etiquetas que guardam algum valor. Elas podem ser de qualquer tipo:

```
int idade = 25;  
String nome = "Lucas";
```

No exemplo acima, temos **duas variáveis**. A variável "**idade**" armazena um número inteiro e a variável "**nome**", um texto (ou cadeia de caracteres).

2



Tipos primitivos

Para criar uma variável (ou declarar uma variável) é necessário informar de qual tipo ela será. Se você declara que uma variável é do tipo **inteiro** ("int") não poderá atribuir um texto ou um valor decimal a ela.

A linguagem Java possui 8 tipos primitivos de variáveis: "**byte**", "**short**", "**int**", "**long**", "**float**", "**double**", "**boolean**" e "**char**". Você pode aprender mais sobre esses tipos nessa documentação oficial da Oracle:

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

Operadores aritméticos

O Java possui os seguintes operadores aritméticos:

- +: Adição
- : Subtração
- *: multiplicação
- /: Divisão
- ?: Módulo (*permite obter o resto de uma divisão*)

Experimente usar esses operadores com esse programa que permite um usuário saber a soma de dois números inteiros:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("Calculadora de números");

        Scanner scanner = new Scanner(System.in);
        System.out.print("Informe o primeiro número: ");
        int numA = scanner.nextInt();

        System.out.print("Informe o segundo número: ");
        int numB = scanner.nextInt();

        int soma = numA + numB;

        System.out.println("A soma deles é: " + soma);
    }
}
```



Operadores relacionais

Usamos operadores relacionais para **comparar o valor entre variáveis**. O Java possui os seguintes operadores relacionais:

- `==`: Igual a
- `!=`: Diferente de
- `>`: Maior que
- `>=`: Maior ou igual a
- `<`: Menor que
- `<=`: Menor ou igual a

Você pode aprender mais sobre os operadores do Java nessa documentação oficial:

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>

Experimente usar operadores relacionais com o programa ao lado:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("Calculadora de números");

        Scanner scanner = new Scanner(System.in);
        System.out.print("Informe o primeiro número: ");
        int numA = scanner.nextInt();

        System.out.print("Informe o segundo número: ");
        int numB = scanner.nextInt();

        boolean comparacao = numA >= numB;
        System.out.println("O primeiro número é maior ou igual do
que o segundo? " + comparacao);
    }
}
```



Operadores lógicos

O Java fornece os seguintes operadores lógicos:

&& - E

|| - OU

?: - Ternário

Os operadores lógicos são geralmente usados em conjunto com estruturas de controle.

Estruturas de controle

if-then-else:

Estrutura responsável por decidir a execução de um bloco de código com base em uma condição.

No exemplo a seguir, uma mensagem informando que o produto está caro será exibida caso o preço informado pelo usuário do programa seja maior do que 5. Caso contrário, será exibida uma mensagem informando que o produto está barato.

```
System.out.println("Preços");
```

```
Scanner scanner = new Scanner(System.in);  
System.out.print("Informe o preço do produto: ");  
int preco = scanner.nextInt();
```

```
if (preco > 5) {  
    System.out.println("O produto está muito caro.");  
} else {  
    System.out.println("O produto está em conta");  
}
```

while:

A estrutura de controle "while" permite que um bloco de código continue sendo executado enquanto uma condição for verdadeira:

```
import java.util.Scanner;
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        boolean continuar = true;  
  
        while (continuar) {  
            System.out.println("Vamos aprender Java?");  
            continuar = scanner.nextBoolean();  
        }  
    }  
}
```

No programa anterior a mensagem "*Vamos aprender Java?*" será exibida enquanto o usuário estiver digitando "**true**" no console. Tente você mesmo.

for:

A estrutura de controle "for" permite que um bloco de código seja executado até que uma condição seja atingida.

No programa abaixo, a mensagem "Um olá do Java" será exibido 10 vezes.

```
public class Main {  
    public static void main(String[] args) {  
        for (int i = 0; i < 10; i++) {  
            System.out.println("Um olá do Java");  
        }  
    }  
}
```

É importante entender as informações que devem ser passadas para o "for":

for (inicialização da variável; condição a ser atingida; incremento da variável inicializada)

switch-case:

Semelhante ao "if-then-else", "switch-case" permite que um ou mais blocos de código sejam executados caso uma condição seja verdadeira. Note que é importante colocar o comando "**break**" no final de cada bloco de um "**case**". Caso contrário, todos os blocos restantes serão executados.

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Qual o seu nome: ");  
        String nome = scanner.next();  
  
        switch (nome) {  
            case "Lucas": {  
                System.out.println("Seja bem-vindo, Lucas");  
                break;  
            }  
            case "Luana": {  
                System.out.println("Olá, Luana");  
                break;  
            }  
            default: {  
                System.out.println("Não te conheço");  
            }  
        }  
    }  
}
```