

# Programando em Java

---

Programando em Java.....	1
Exercícios Comando de Controle de Fluxo.....	2
Primeiro Exercício.....	2
Segundo Exercício .....	2
Terceiro Exercício .....	2
Quarto Exercício .....	3
Quinto Exercício .....	3
Sexto Exercício.....	3
Sétimo Exercício .....	4
Oitavo Exercício.....	4

# Exercícios Comando de Controle de Fluxo

---

É muito importante que você faça os exercícios propostos antes de passar para a próxima seção, porque os exercícios lhe ajudarão a fixar a matéria e é neste momento que muitas dúvidas surgirão.

Antes de ver a resposta, tente fazer até conseguir. Alguns exercícios propostos são padrões para exercitar a lógica, não vale buscar ajuda no Google, hein! ☹

Não jogue a toalha facilmente. ☺

No projeto Exercicio, crie o pacote cap4. Neste pacote você vai criar os exercícios referentes ao que aprendemos sobre este tema na vídeoaula ControleDeFluxo.

## Primeiro Exercício

- 1) Faça uma classe que informe se os alunos foram aprovados, reprovados ou estão em recuperação, de acordo com as seguintes regras:
  - a) A média é calculada a partir de três notas;
  - b) Estas notas devem ser passadas por meio do método *nextDouble()* da classe *Scanner* (veja a vídeoaula ClasseScanner disponível nesta seção).
  - c) Para média maior ou igual a sete, aluno aprovado;
  - d) Para média maior ou igual a cinco e menor ou igual à sete, aluno em recuperação;
  - e) Para média menor que cinco, aluno reprovado.
  - f) A média e a situação do aluno (aprovado, reprovado ou recuperação) devem ser apresentadas no console do Eclipse.
  - g) Sugestão para o nome da classe: *CalculoMediaEscolar*

## Segundo Exercício

- 2) Crie uma classe que calcule o IMC (índice de massa corporal) de uma pessoa.
  - a) O cálculo do IMC é feito dividindo o peso (em quilogramas) pela altura (em metros) ao quadrado.
  - b) A altura e peso devem ser passadas por meio do método *nextDouble()* da classe *Scanner* (veja a vídeoaula ClasseScanner disponível nesta seção).
  - c) IMC menor ou igual a 18,5: abaixo do peso normal
  - d) IMC maior que 18,5 e IMC menor ou igual a 25: peso normal
  - e) IMC maior que 25 e IMC menor ou igual a 30: acima do peso normal.
  - f) IMC maior que 30: obesidade
  - g) O valor do IMC e a situação devem ser apresentados no console do Eclipse.
  - h) Sugestão para o nome da classe: *CalculoIMC*.

## Terceiro Exercício

- 3) Crie uma classe que calcule o Imposto de renda a pagar de acordo com as seguintes regras:

- a) A fórmula para calcular o IR a deduzir dos salários é:

$$IR = \text{salario} * \text{alíquota} - \text{dedução}$$

- b) A tabela abaixo apresenta as faixas salariais e respectivas alíquotas e dedução:

Faixas salariais	Alíquota %	Parcela a deduzir do imposto
Até 1.903,98	Isento	-
De 1.903,99 até 2.826,65	7,5	142,80
De 2.826,66 até 3.751,05	15,0	354,80
De 3.751,06 até 4.664,68	22,5	636,13
Acima de 4.664,68	27,5	869,36

Fonte: <https://www.tabeladoirrf.com.br/tabela-irrf-2016.html>

- c) O valor do salário deve ser passado por meio do método *nextDouble()* da classe *Scanner* (veja a vídeoaula ClasseScanner disponível nesta seção).
- d) Deverá ser apresentado no console do Eclipse o valor do IR a recolher. Por exemplo, para um salário de R\$ 4.000,00, a saída deverá ser:

```
Salário: 4000.0
Valor do IR a recolher: 230.64
```

#### Quarto Exercício

- 1) Crie uma classe que mostre todos os números pares no intervalo de 1 a 100.
- 2) Sugestão para o nome da classe: MostraNumerosPares.

#### Quinto Exercício

- 1) Crie uma classe que verifique se um número é par ou ímpar.
- 2) Use o método *nextInt()* da classe *Scanner* para pegar o número digitado no console do Eclipse.
- 3) Deve ser dada uma saída dizendo se o número é par ou ímpar.
- 4) Sugestão para o nome da classe: VerificaNumero.

#### Sexto Exercício

Um número é considerado *primo* quando é divisível por ele mesmo e por um.

- 1) Crie uma classe que verifique se um número é primo.
- 2) Use o método *nextInt()* da classe *Scanner* para pegar o número digitado no console do Eclipse.
- 3) Quando o número passado for primo, o seguinte deve aparecer no console do Eclipse:

```
Digite um número: 13
13 é um número primo. É divisível por 2 números.
```

- 4) Quando o número passado não for primo, o seguinte deve aparecer no console do Eclipse:

```
Digite um número: 14
14 não é um número primo. É divisível por 4 números.
```

- 5) Sugestão para o nome da classe: NumerosPrimos.

### Sétimo Exercício

- 1) Usando o comando *for*, crie uma classe que mostre as tabuadas do 1 ao 10.
- 2) Sugestão para o nome da classe: Tabuada

### Oitavo Exercício

- 1) Crie uma classe que calcule o frete de produtos comprados em uma empresa na Internet, conforme a seguir:
  - a) A regra de negócios para cobrança de frete é:

Categoria de Produto	Código da Categoria	Custo em R\$
Livros	1	3,50
Celulares	2	4,50
Tablets	3	5,50
Games	4	6,50
Eletrônicos	5	7,50
Outros	6	10,00

- b) As informações *Código da Categoria* e *Quantidade* devem ser passadas no console do Eclipse. Para isso utilize a classe Scanner e seu método `nextDouble()`;
- c) Quando não houver mais *Código de Categoria* a ser inserido, digite *0* para finalizar.
- d) Após digitar *0*, deve ser apresentado uma saída com o valor de frete por categoria e o total a pagar do frete. Abaixo, um exemplo de saída, caso o usuário tenha comprado dois celulares e três games:

```
Informe o número do produto (1-6) (Digite 0 para parar): 2
Digite a quantidade vendida: 2
Informe o número do produto (1-6) (Digite 0 para parar): 4
Digite a quantidade vendida: 3
Informe o número do produto (1-6) (Digite 0 para parar): 0
Frete de Livros: $0,00
Frete de Celulares: $9,00
Frete de Tablets: $0,00
Frete de Games: $19,50
Frete de Eletrônicos: $0,00
Frete de outros produto: $0,00
-----
Total a pagar de frete: 28.5
```

- e) Sugestão para o nome da classe: `CalculoFrete.java`

Se você não souber por onde começar, leia algumas dicas abaixo para um possível caminho de solução deste exercício.

- f) Você vai precisar de dez variáveis, sendo:
  - Uma do tipo Scanner que deverá pegar o que for digitado como Código de Categoria e Quantidade.

- Sete do tipo *double*, sendo que seis vão guardar a quantidade comprada de cada item e uma guardará o valor total do frete.
- Uma tipo *int* que deverá guardar a quantidade digitada no console do Eclipse.
- Uma do tipo inteiro que deve guardar o número de identificação da categoria atual (inicialize esta variável com um número diferente de 0). Exemplo:

```
int codigoCategoria = 1; // número de identificação de produto atual
```

- Posteriormente esta variável deve guardar o que for passado no console do Eclipse:

```
codigoCategoria = input.nextInt();
```

- g) Use o comando *while* para determinar quando o *loop* deve ser parado e o resultado apresentado. Ou seja, *enquanto* o *codigoCategoria* for diferente de 0, a seguinte saída deve aparecer no console do Eclipse:

```
Informe o número da categoria do produto (1-6) (Digite 0 para parar):
```

- h) No escopo do *while*, utilize o comando *if-else* para garantir que os códigos das categorias digitados estejam entre 1 e 6.

```
codigoCategoria >= 1 && codigoCategoria <= 6
```

- i) Se for digitado um valor de *codigoCategoria* fora do intervalo (1-6) e diferente de 0, a seguinte mensagem deve aparecer no console do Eclipse:

```
Digite um número entre 1 e 6 para determinar a categoria do produto ou 0 para parar
```

- j) Dentro do escopo do *if-else*, atribua à variável *quantidade* o conteúdo método *nextInt()* da classe *Scanner*:

```
quantidade = input.nextInt();
```

- k) Ainda no escopo do *if-else*, utilize o comando *switch-case* para determinar o preço de frete de cada produto.
- l) Após o fim do escopo do comando *while*, calcule o total do frete e apresente na tela todos os valores.