

Programação em Java

Djonathan Luiz de Oliveira Quadras

14/04/2020

Contents

Bem vindo!	5
Sobre Programação	7
Sobre o Livro	7
Ferramentas Computacionais	7
1 Lógicas de Programação	9
1.1 Pseudocódigos	9
2 Princípios da Programação em Java	11
2.1 Classes	12
2.2 Objetos	12
2.3 Variáveis	12
2.4 Criando o primeiro código em Java	13
3 Operadores	15
3.1 Operações Matemáticas	15
3.2 Operadores Relacionais	16
3.3 Lógica Booleana e Operadores Lógicos	16
4 Estruturas de Decisão ou Seleção	19
4.1 If-Else	19
4.2 Operador Ternário	19
4.3 Switch	19
Exercícios	19

5 Estruturas de Repetição	21
5.1 For	21
5.2 While	21
5.3 Do-While	21
Exercícios	21

Bem vindo!

Bem vindo ao livro de programação em Java! Enjoy your visit! :)

Sobre Programação

Working on it! :)

Sobre o Livro

Working on it! :)

Ferramentas Computacionais

Working on it! :)

Chapter 1

Lógicas de Programação

Working on it! :)

1.1 Pseudocódigos

Working on it! :)

Chapter 2

Princípios da Programação em Java

A linguagem Java é considerada como sendo de baixo nível, ou seja, sua programação (e conseqüentemente sua leitura) não é trivial como se estivéssemos escrevendo um livro. Ela se apresenta como sendo uma linguagem muito burocrática e que necessita de diversos processos para que funcione corretamente. Um código em Java sempre deverá ser montado com uma estrutura semelhante a apresentada na figura abaixo.

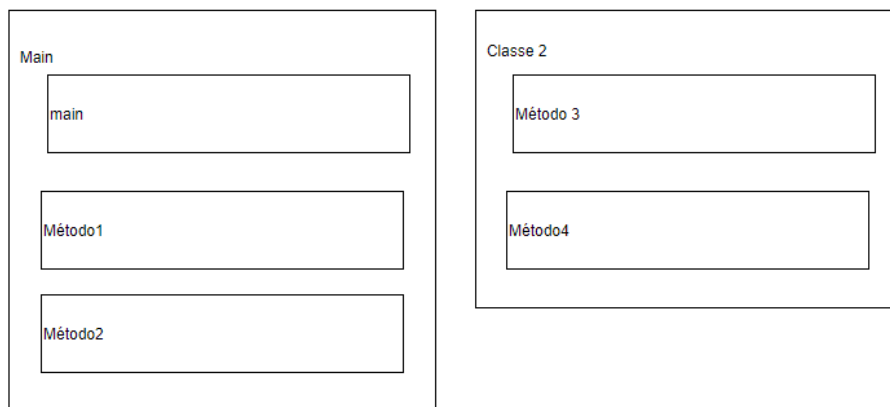


Figure 2.1: Classes

Um código é geralmente composto por Classes e Objetos. Esses pontos serão explicados nas subseções a seguir.

2.1 Classes

As classes de programação são receitas de um objeto, aonde têm características e comportamentos, permitindo assim armazenar propriedades e métodos dentro dela. Uma classe geralmente representa um substantivo, por exemplo uma pessoa, um lugar ou um sistema.

Vamos imaginar uma universidade. Ela é composta por diversas classes: alunos, professores, prédios, salas de aula. Note que uma classe pode ser parte de uma outra classe, ou seja, todas as classes apresentadas fazem parte de uma classe maior chamada de universidade. Mas afinal, o que diferencia uma classe da outra? A diferenciação, além do nome dado, está nos objetos que compõe a classe.

2.2 Objetos

Objetos são caracterizados por atributos e métodos e são características definidas pelas classes.

2.2.1 Atributos ou Propriedades

Atributos são as características de um objeto. Essas características também são conhecidas como variáveis. Utilizando o exemplo dos cães, temos alguns atributos, tais como: cor, peso, altura e nome.

2.2.2 Métodos

Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos. Utilizando o exemplo dos cães, temos alguns exemplos: latir, correr, pular.

2.3 Variáveis

Variáveis são uma forma de salvar informações no computador. As variáveis que nós armazenamos em um programa são acessadas pelo nome que nós atribuímos a elas.

Uma variável pode ter seu valor alterado ao longo do programa, ou seja, seu valor é variável. É apenas necessário informar ao computador qual o tipo de informação será armazenado e, então, dar um nome ao valor. Existem muitos tipos de dados que podem ser utilizados para definir uma variável. Os principais estão listados abaixo.

Table 2.1: Tipos de Variáveis

Tipo	Descrição
int	Números Inteiros
double	Números Decimais
boolean	Armazena apenas 'Verdadeiro' ou 'Falso'
String	Frases
char	Caracteres

Existem diversos outros tipos de variáveis, que serão apresentados com maior detalhe nas seções seguintes.

2.4 Criando o primeiro código em Java

A primeira etapa para a criação de um código em Java é a criação de uma classe. Por padrão, a classe principal do seu programa se chama “Main”.

```
public class Main{  
    //Oi! Eu sou uma classe!  
}
```

Perceba alguns pontos importantes:

- O código inicia com a palavra “*public*”. Isso indica que sua classe pode ser encontrada por outras classes que estiverem salvas no mesmo diretório. Você pode alterar por “*private*” ou “*protected*”;
- Há a presença de chaves { } no código. Isso é uma burocracia da linguagem. A classe é correspondente ao que está dentro das chaves. Tudo o que estiver fora delas não será considerado como parte do código;
- Para fazer comentários em Java, usa-se duas barras // .

Agora que temos a nossa classe criada, precisamos criar os objetos. Por agora vamos criar apenas um método.

```
public class Main{  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

Agora, dentro da nossa classe foi criado o método main. Isso também segue um padrão imposto pela linguagem. Há, novamente, alguns pontos que merecem destaque.

- A função “System.out.println()” é utilizada para escrever algum texto no console;
- Todas as linhas de código devem ser terminadas com um ponto e vírgula.

Colocando este código para correr, deverá aparecer no console a mensagem “*Hey, Hello World!*”. Pronto! Você acaba de criar seu primeiro código!

2.4.1 Criando Variáveis

Working on it! :)

Chapter 3

Operadores

Um artifício essencial para quem programa é a utilização de operadores. Sejam eles matemáticos, lógicos ou relacionais, é praticamente impossível desenvolver um código sem utilizar um operador. Nessa seção serão apresentados e explicados os operadores em linguagem Java.

3.1 Operações Matemáticas

É muito comum utilizarmos as operações matemáticas em programação. Frequentemente estaremos somando ou subtraindo duas variáveis. Com isso, torna-se necessário conhecer os operadores matemáticos em java. Os básicos estão listados na tabela abaixo.

Existem também os operadores incrementais. Estes operadores são utilizados quando queremos acrescentar (ou diminuir) uma unidade no valor de uma variável. Eles estão listados na tabela abaixo.

Table 3.1: Operações Matemáticas

Operador	Uso	Descrição
+	$x + y$	Soma x e y
-	$x - y$	Subtrai y de x
*	$x * y$	Multiplica x por y
/	x / y	Divide x por y
%	$x \% y$	Calcula o resto da divisão de x por y

Table 3.2: Operações Matemáticas Úteis

Operador	Uso	Descrição
++	b++	Incrementa 1 em b, avaliando b antes de incrementar
++	b++	Incrementa 1 em b, avaliando b depois de incrementar
-	b--	Decrementa 1 em b, avaliando b antes de incrementar
-	b--	Decrementa 1 em b, avaliando b depois de incrementar

Table 3.3: Operações Relacionais

Operador	Uso	Descrição
>	x > y	x é maior que y
>=	x >= y	x é maior ou igual a y
<	x < y	x é menor que y
<=	x <= y	x é menor ou igual a y
==	x == y	x é igual a y
!=	x != y	x é diferente de y

3.2 Operadores Relacionais

Os operadores relacionais são utilizados quando queremos estabelecer relações entre as variáveis, ou seja, quando queremos comparar uma variável com a outra. Eles estão apresentados na tabela abaixo.

Para utilizar outras ferramentas matemáticas, é necessário invocar a classe matemática do Java. Para isso, faz-se `double valor = Math.` e seleciona-se a função desejada (irá aparecer automaticamente uma lista com todas as opções possíveis. Por exemplo, é possível calcular a raiz quadrada, exponencial e cosseno conforme apresentado a seguir.

```
double raiz = Math.sqrt(9); // raiz quadrada
double exponencial = Math.exp(2); // e elevado ao quadrado
double cosseno = Math.cos(Math.PI); //cosseno de pi em radianos
```

3.3 Lógica Booleana e Operadores Lógicos

A lógica booleana foi desenvolvida no século 19 pelo matemático George Boole como um esquema para usar métodos algébricos na formalização da lógica e raciocínio. A álgebra booleana trabalha com dois números, 1 e 0. Para a programação, trataremos 1 como sendo “Verdadeiro” e 0 como sendo “falso”.

As funções relacionais sempre gerarão resultados booleanos. Dessa forma, é possível utilizar operadores lógicos para trabalhar com as informações e gerar novos

Table 3.4: Operações Matemáticas Úteis

Operador	Descrição
&	AND lógico
ou	OR lógico
^	XOR lógico
!	NOT lógico

Table 3.5: Operações com Operador AND

Valor 1	Operador	Valor 2	Resultado
true	&	true	true
true	&	false	false
false	&	false	false

resultados. Os operadores lógicos são amplamente utilizados quando queremos criar expressões lógicas. Os operadores lógicos estão apresentados na tabela abaixo.

3.3.1 Operador AND

O operador AND, ou “e”, retorna “verdadeiro” só e somente só quando está comparando dois booleanos com valor “verdadeiro”.

3.3.2 Operador OR

O operador OR, ou “ou”, retorna “verdadeiro” quando está comparando dois booleanos em que ao menos um tenha valor “verdadeiro”.

3.3.3 Operador XOR

O operador XOR, ou “ou exclusivo”, retorna “verdadeiro” se só e somente só estiver comparando dois booleanos em que somente um dos valores seja “ver-

Table 3.6: Operações com Operador OR

Valor 1	Operador	Valor 2	Resultado
true		true	true
true		false	true
false		false	false

Table 3.7: Operações com Operador XOR

Valor 1	Operador	Valor 2	Resultado
true	\wedge	true	false
true	\wedge	false	true
false	\wedge	false	false

Table 3.8: Operações com Operador XOR

Operador	Valor	Resultado
!	true	false
!	false	true
!!	true	true

dadeiro.

3.3.4 Operador NOT

O operador NOT, ou “não”, retorna o valor contrário ao valor escolhido. Note que este operador compara dois valores, apenas muda de verdadeiro para falso e vice-versa.

Exercícios {-}

Chapter 4

Extruturas de Decisão ou Seleção

Working on it! :)

4.1 If-Else

Working on it! :)

4.2 Operador Ternário

Working on it! :)

4.3 Switch

Working on it! :)

Exercícios

Working on it! :)

Chapter 5

Estruturas de Repetição

Working on it! :)

5.1 For

Working on it! :)

5.2 While

Working on it! :)

5.3 Do-While

Working on it! :)

Exercicios