



Domain Driven Design - Java

Prof. Gilberto Alexandre das Neves
profgilberto.neves@fiap.com.br

Orientação à Objetos

Polimorfismo significa ter ***muitas formas***, que significa um único nome representando um código diferente, selecionado por algum mecanismo automático. “Um nome, vários comportamentos”.

O Polimorfismo não é um pensamento novo para nós. Ele está contido em nosso dia a dia, principalmente na linguagem. Veja os exemplos:

1. Ontem sai para **dançar** com uns amigos, mas acabamos **dançando** porque não conseguimos encontrar um lugar que nos agradasse.
2. José **cantou** a noite inteira no Karaokê e João **cantou** a noite inteira a namorada de José.



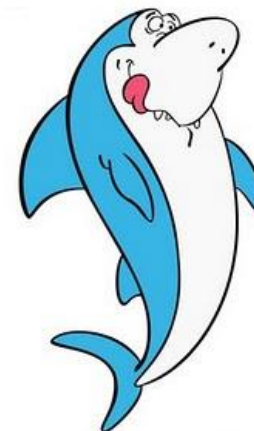
Pensando mais em objetos e funcionalidades, pense agora no termo **abrir**, por exemplo. Você pode abrir uma porta, uma caixa, uma janela e uma conta bancária.

A palavra abrir pode ser aplicada a muitos objetos do mundo real sendo que cada objeto interpreta **abrir** a sua própria maneira.

Porém, você pode simplesmente dizer **abrir**, para descrever a ação.

Os principais tipos de **polimorfismo** são:

- Sobrescrita
- Sobrecarga



Métodos com o mesmo nome, mas com funcionalidades diferentes. Exemplo:

Um objeto da classe **Carro**:

- o que ocorre quando se pressiona o pedal do acelerador enquanto se dirige cada um dos carros (básico e luxuoso)?
- o pedal do acelerador tem a capacidade de atuar de forma diferente, apesar de parecer o mesmo em todas as situações.

Cada objeto da família pode ter métodos com o mesmo nome, mas com comportamento diferente.



Ocorre quando existem dois métodos (ou mais) com mesmo nome, na mesma classe com assinaturas diferentes.

O método será escolhido de acordo com o número de parâmetros, tipo ou valor de retorno esperado.

Exemplo:

```
exibir(texto)
```

```
exibir(texto, número)
```

```
exibir("Olá Mundo!")
```

```
exibir("Astrogildo", 33)
```



Tipos de Variáveis em Java

Tipos de Variáveis

Tipos **primitivos** da linguagem, utilizados na criação de variáveis.

Tipo	Quantidade de bits	Valores
char	16	'\u0000' a '\uFFFF'
byte	8	-128 a + 127
int	32	-2.147.483.648 a +2.147.483.647
short	16	-32.768 a + 32.767
long	64	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807
float	32	-3.40292347E+38 a +3.40292347E+38
double	64	-1.79769313486231570E+308 a +1.79769313486231570E+308
boolean	8	true ou false

Tipos **não primitivos**:

Tipo	Quantidade de bits	Valores
String	??	cadeia de caracteres (usar aspas)

Tipos primitivos

Crie uma classe com o nome **Primitivo** no pacote **br.com.fiap**. Adicione o método **main** a esta classe e digite o código abaixo, teste o programa e observe seu comportamento. **Atenção** as **variáveis** seguem a mesma **convenção de nomenclatura** dos atributos e métodos.

```
1  package br.com.fiap;
2
3  ▶ public class Primitivo {
4  ▶      public static void main(String[] args) {
5          char sexo = 'f';
6          byte idade = 25;
7          short codigo = 15650;
8          float media = 8.7f;
9          int alunos = 50, classes = 11;
10         long brasileiros = 216535748;
11         double dolar = 5.65;
12         boolean alternativa = false;
13         System.out.println("Sexo: " + sexo + " Idade: " + idade + " Código: " + codigo);
14         System.out.println("Média: " + media + " Alunos: " + alunos + " Turmas: " + classes);
15         System.out.println("Habitantes: " + brasileiros + " Dólar U$ " + dolar);
16         System.out.println("Alternativa: " + alternativa);
17     }
18 }
```

Constantes

Uma constante é um tipo de “variável” que **não pode alterar seu conteúdo** depois de ter sido inicializado permanecendo com o mesmo valor durante toda a execução do programa.

Na realidade não existem constantes em Java, o que existe é um tipo de variável com comportamento semelhante a uma constante de outras linguagens.

Em Java, essa variável é definida como **final**. A **convenção de nomenclatura** é identificar as variáveis do tipo **final** com **todas as letras maiúsculas**, e quando existe mais de uma palavra elas são separadas pelo caractere de **underscore** (_).

Constantes

Crie uma nova classe com o nome **Constante** neste projeto (dentro do mesmo pacote) também com o método **main** (somente por **questões didáticas**, normalmente em um projeto somente uma classe irá possuir o método **main**). Digite o código como indicado e execute o programa.

```
1 package br.com.fiap;
2
3 public class Constante {
4     public static void main(String[] args) {
5         final double PI = 3.1416;
6         final float GRAVIDADE_TERRESTRE = 9.8f;
7
8         System.out.println("O valor de PI é: " + PI);
9         System.out.println("O valor da gravidade do planeta Terra é de: " + GRAVIDADE_TERRESTRE);
10
11     }
12 }
13
```

Operadores Aritméticos

Veja a seguir os operadores aritméticos usados na linguagem Java.

Função	Sinal	Exemplo
Adição	+	$x + y$
Subtração	-	$x - y$
Multiplicação	*	$x * y$
Divisão	/	x / y
Resto da divisão inteira	%	$x \% y$
Sinal negativo	-	$-x$
Sinal positivo	+	$+x$
Incremento unitário	++	$x++$
Decremento unitário	--	$x--$

Sequência de escape

Sequência de escape

A linguagem de programação Java também possui algumas sequências de escape, que são atalhos utilizados para representar um caractere especial como, por exemplo, uma quebra de linha (`\n`).

São utilizados **dentro de aspas** em uma sequência de caracteres.

Sequência	Caractere especial
<code>\b</code>	Backspace
<code>\f</code>	Form feed
<code>\n</code>	Nova linha
<code>\r</code>	Retorno
<code>\t</code>	Tabulação
<code>\"</code>	Aspas
<code>\'</code>	Apóstrofo
<code>\\</code>	Barra invertida

Praticando...

Teste seus conhecimentos

Utilizando suas próprias palavras, responda:

1. Explique Orientação à Objetos.
2. Explique o que é objeto.
3. Explique o que é classe.
4. Explique o que é abstração.
5. Explique o que é herança.
6. Explique o que é encapsulamento.
7. Explique o que é polimorfismo.
8. Quais os principais tipos de polimorfismo? Explique cada um deles.



- Inicie um novo projeto **Java**, crie os pacotes **br.com.fiap** e dentro deste pacote crie uma nova classe chamado **Questionario**.
- Dentro desta classe crie o método **main**.
- Dentro do método **main**, exiba no console com o método **println()** cada uma das questões e sua respectiva resposta.

Dica: Se você utilizar o método **println** sem colocar nenhuma mensagem, ele apenas vai pular de linha no console.

- No mesmo projeto do praticando anterior crie uma nova classe chamada **Operadores**.
- Dentro desta classe crie o método **main**.
- Dentro do método **main**, crie variáveis dos mais diversos tipos numéricos, realize operações aritméticas com essas variáveis e armazene os resultados e finalmente exiba no console com o método **println()** cada valor das variáveis, a operação utilizada e seus resultados (utilize as **sequências de escape** para pular de linha e organizar a mensagem exibida).



Java como programar. Paul Deitel e Harvey Deitel. Pearson, 2011.

Java 8 – Ensino Didático : Desenvolvimento e Implementação de Aplicações. Sérgio Furgeri. Editora Érica, 2015.

Até breve!