

Introdução a Orientação a Objetos

Conceitos e definições relacionados a
orientação a objetos, classes, objetos,
atributos e métodos

Dados Internacionais de Catalogação na Publicação (CIP)

S729i

Souza, Wilton Marinho Carneiro.

Introdução a orientação a objetos. Gama, DF: UNICEPLAC, 2022.

28 p.

1. Orientação a objetos. 2. Programação. 3. software e sistemas. I. Título.

CDU: 004.41

Introdução a Orientação a Objetos

- **Orientação a objetos:**

- É um paradigma aplicado na programação que consiste na interação entre diversas unidades chamadas de objetos;
- Uma maneira resolver problemas de software utilizando conceitos do Mundo Real. O componente fundamental é o objeto que combina estrutura e comportamento em uma única entidade.

Introdução a Orientação a Objetos

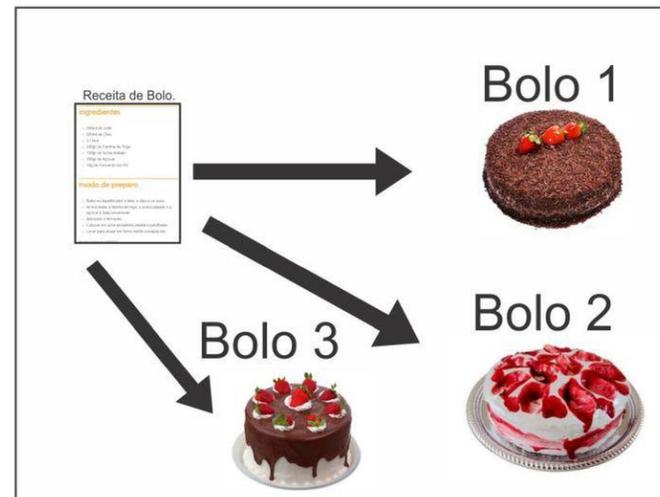


Introdução a Orientação a Objetos



Introdução a Orientação a Objetos

Classes e Objetos



Origem

- Nos anos 70 surge Smalltalk, a primeira linguagem totalmente em Orientação a Objeto (O.O).
- C++, evolução de C, já possuía conceitos O.O.
- Na década de 80 várias linguagens já usavam conceitos O.O.
 - Delphi;
 - Java.

OO x Estruturada

Programação estruturada

- **Base**
 - **Sequência:** Uma tarefa é executada após a outra, linearmente.
 - **Decisão:** A partir de um teste lógico, determinado trecho de código é executado, ou não.
 - **Iteração:** A partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes.
- **Vantagens**
 - É fácil de entender. Ainda muito usada em cursos introdutórios de programação.
 - Execução mais rápida.
- **Desvantagens**
 - Baixa reutilização de código.
 - Códigos confusos: Dados misturados com comportamento.

OO x Estruturada

Programação OO

- **Base**
 - Classes e Objetos.
 - Métodos e Atributos.
- **Vantagens**
 - Melhor organização do código.
 - Bom reaproveitamento de código.
- **Desvantagens**
 - Desempenho mais baixo que o paradigma Estruturado.
 - Mais difícil compreensão.

Classes e Objetos

- A estrutura de um objeto são:
- **Atributos**
 - Representa as características do objeto (estado);
- **Métodos**
 - Representa os comportamentos do objeto (operações);
 - Operações podem mudar os valores dos atributos assim mudando o estado de um objeto.

Classes e Objetos

- **Atributos**

- Janela;
- Porta;
- Chaminé;
- Garagem;
- Quarto;
- Etc;



= Objeto

- **Métodos**

- Abrir porta;
- Fechar janela;
- Lavar garagem;
- Acender chaminé;
- Etc;



= Classe

Classes e Objetos

- A orientação a objetos é um paradigma de análise, projeto e programação de sistema de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.
- Orientação a objetos é uma maneira de programar que auxilia na organização do código e resolve muitos problemas na manutenção de código da programação estruturada.

Métodos

```
class Conta {  
    double saldo;  
  
    boolean sacar(double valor) {  
        if (saldo < valor) {  
            return false;  
        } else {  
            saldo = saldo - valor;  
            return true;  
        }  
    }  
}
```

-

Métodos

Tipo de
retorno

```
class Conta {  
    double saldo;  
  
    boolean sacar(double valor) {  
        if (saldo < valor) {  
            return false;  
        } else {  
            saldo = saldo - valor;  
            return true;  
        }  
    }  
}
```

-

Métodos

```
class Conta {  
    double saldo;  
    boolean sacar(double valor) {  
        if (saldo < valor) {  
            return false;  
        } else {  
            saldo = saldo - valor;  
            return true;  
        }  
    }  
}
```

Nome do método

Tipo de retorno

Métodos

Tipo de retorno

```
class Conta {  
    double saldo;  
    boolean sacar(double valor) {  
        if (saldo < valor) {  
            return false;  
        } else {  
            saldo = saldo - valor;  
            return true;  
        }  
    }  
}
```

Nome do método

Parâmetros do método

•

Métodos

```
class Conta {  
    double saldo;  
    boolean sacar(double valor) {  
        if (saldo < valor) {  
            return false;  
        } else {  
            saldo = saldo - valor;  
            return true;  
        }  
    }  
}
```

Tipo de retorno

Nome do método

Parâmetros do método

Lista de instruções do método

Exemplo de retorno do método

```
Conta minhaConta = new Conta();  
minhaConta.saldo = 1000;  
boolean resposta = minhaConta.sacar(2000);  
if (resposta ){  
    System.out.println("Operação efetuada com sucesso");  
} else {  
    System.out.println("Sem saldo");  
}
```

Objetos são acessados por referências

- Quando declaramos uma variável para associar a um objeto, na verdade, essa variável não guarda o objeto, e sim referência ao objeto;

Objetos são acessados por referências

```
public static void main(String args[]) {  
    Conta c1 = new Conta();  
    Conta c2 = new Conta();  
}
```

- Dizemos que c1 se refere a um objeto. Não é correto dizer que c1 é um objeto, pois c1 é uma variável referência.

Objetos são acessados por referências

- O que será escrito no console quando executamos o código abaixo?

```
class TestaReferencias {  
    public static void main(String args[]) {  
        Conta c1 = new Conta();  
        c1.deposita(100);  
        Conta c2 = c1;  
        c2.deposita(200);  
  
        System.out.println(c1.saldo);  
        System.out.println(c2.saldo);  
    }  
}
```

Objetos são acessados por referências

- O que acontece aqui? O operador = copia o valor de uma variável. Mas qual é o valor da variável c1? É o objeto? Não. Na verdade, o valor guardado é a referência (endereço) de onde o objeto se encontra na memória principal.
- Na memória, o que acontece nesse caso:
 - Conta c1 = new Conta();
 - Conta c2 = c1;

Exemplo

- Exemplo projeto em Java
- 1º Passo – Criar um projeto java no Eclipse;

Exemplo

- Exemplo projeto em Java
- 2º Passo – Criar a classe “Casa”;

```
public class Casa {  
  
    String janela;  
    String porta;  
    String tipoChamine;  
    String garagem;  
    String quarto;  
  
    public void abrirPorta() {  
        System.out.println("Porta aberta");  
    }  
  
    public void fecharPorta() {  
        System.out.println("Porta fechada");  
    }  
  
    public void abrirJanela() {  
        System.out.println("Janela aberta");  
    }  
  
    public void fecharJanela() {  
        System.out.println("Janela fechada");  
    }  
  
}
```

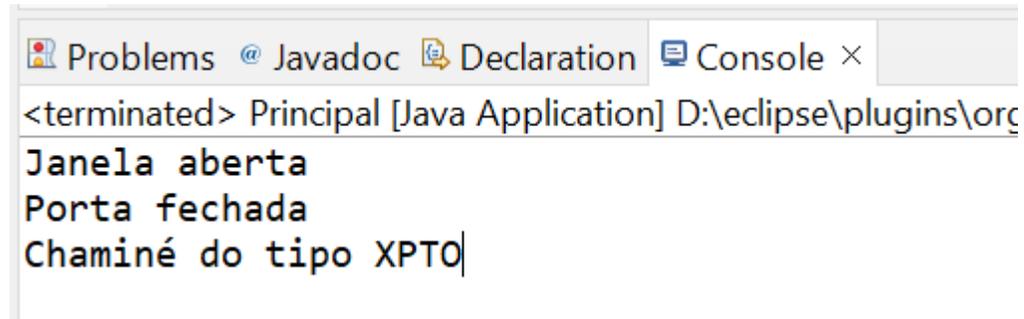
Exemplo

- Exemplo projeto em Java
- 3º Passo – Criar a classe principal que irá criar (instanciar) objetos do tipo “Casa”;

```
public class Principal {  
  
    public static void main(String[] args) {  
        Casa minhaCasa = new Casa();  
  
        minhaCasa.tipoChamine = "Chaminé do tipo XPTO";  
        minhaCasa.janela = "Quadrada";  
  
        minhaCasa.abrirJanela();  
        minhaCasa.fecharPorta();  
  
        System.out.println(minhaCasa.tipoChamine);  
  
    }  
}
```

Exemplo

- Exemplo projeto em Java
- 4º Passo – Executar a classe principal e verificar o resultado do Console;



```
Problems @ Javadoc Declaration Console ×  
<terminated> Principal [Java Application] D:\eclipse\plugins\org  
Janela aberta  
Porta fechada  
Chaminé do tipo XPTO|
```

Função Scanner

- Semelhante ao “Scanf()” na linguagem C;
- É utilizada para ler valor no console;
- É necessário importar a classe Scanner();
 - **import java.util.Scanner;**
- Exemplo

```
Scanner input = new Scanner(System.in);
int number1; // primeiro número a adicionar
int number2; // segundo número a adicionar
int sum;      // soma de number1 e number2

System.out.print( " Digite o primeiro número: "); // prompt
number1 = input.nextInt(); // lê o primeiro nº fornecido pelo usuário

System.out.print( " Digite o segundo número: "); // prompt
number2 = input.nextInt(); // lê o segundo nº fornecido pelo usuário
```

Referências bibliográficas

Edward, F. OCA Java SE 8: Guia de Estudos para o Exame 1Z0-808. Bookman; Grupo A, 2018. 9788582604779. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582604779/>. Acesso em: 07 Jul 2022.

Sérgio, F. Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações. Editora Saraiva, 06/2015. 9788536519340. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519340/>. Acesso em: 07 Jul 2022.

Evolução que
TRANSFORMA

Dúvidas?

wilton.souza@uniceplac.edu.br



uniceplac.edu.br



[/uniceplac](https://www.facebook.com/uniceplac)