POO usando Java: Classes

Programação Orientada a Objetos

Profs. Aleffer Rocha
aleffer.rocha@ifpr.edu.br
Prof. William Deus
william.deus@ifpr.edu.br

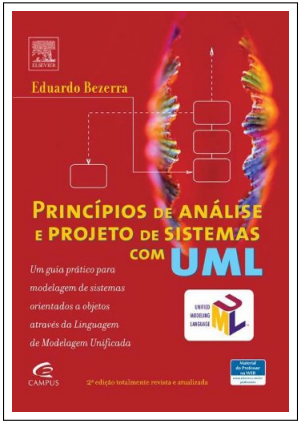
INSTITUTO FEDERAL
Paraná
Campus
Pinhais

Material atualizado em 6 de agosto de 2024

Materiais utilizados

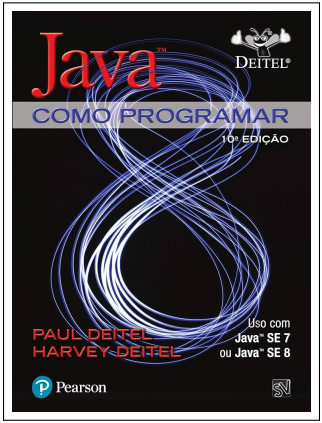
ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores:** algoritmos , Pascal, C/C++ (padrão ANSI) e Java. 3. ed. São Paulo: Pearson Education do Brasil, 2012.

Materiais utilizados



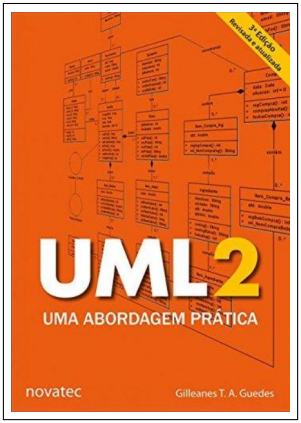
BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML.** 2. ed. rev. atual. Rio de Janeiro, RJ: Elsevier, 2007.

Materiais utilizados



DEITEL, Paul J.; DEITEL, Harvey M. **Java:** como programar. 10. ed. São Paulo: Pearson Education do Brasil, 2017.

Materiais utilizados

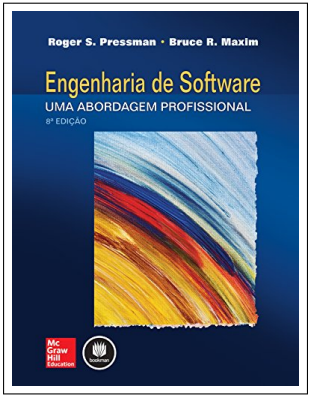


GUEDES, Gilleanes T. A. **UML 2:** uma abordagem prática. 3. ed. rev. atual./ Reimpr. São Paulo: Novatec, 2018

Materiais utilizados

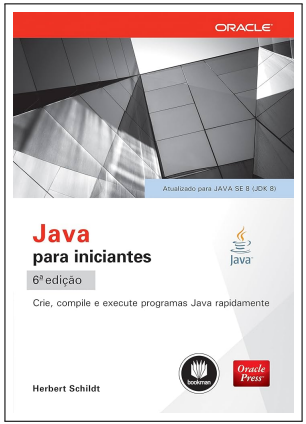
MEDEIROS, Ernani. **Desenvolvendo software com UML 2.0:** definitivo. São Paulo, SP: Makron Books, 2004.

Materiais utilizados



PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software:** uma abordagem profissional. 8. ed. Porto Alegre: MacGraw Hill, 2016.

Materiais utilizados



SCHILDT, Hebert. **Java para iniciantes:** crie, compile e execute programas java rapidamente. 6. ed. Porto Alegre: Bookman, 2015.

Classes

INSTITUTO FEDERAL
Paraná
Campus Pinhais

```
1 <especificador_de_acesso> class <nome_classe> {
2     <especificador_de_acesso> <declaracao_dos_atributos>;
3     <especificador_de_acesso> <declaracao_dos_metodos>;
4 }
```

- **<especificador_de_acesso>:** define a visibilidade dos atributos e métodos e os possíveis identificadores.
- **<nome_classe>:** é o identificador da classe.
- **<declaracao_dos_atributos>:** onde os atributos são definidos com especificador de acesso, tipo e nome.
- **<declaracao_dos_metodos>:** onde os métodos são definidos com especificador de acesso, tipo do retorno, nome e lista de parâmetros.

Atributos

INSTITUTO FEDERAL
Paraná
Campus Pinhais

Visibilidade: nível de encapsulamento do atributo.

- `(+) public:` permite que os atributos e métodos sejam acessados por qualquer classe.
- `(-) private:` permite que atributos e métodos sejam acessados (visíveis) somente dentro da classe onde foram declarados.
- `(#) protected:` o atributo é acessado somente por classes que estejam no mesmo pacote.

Atributos

INSTITUTO FEDERAL
Paraná
Campus Pinhais

Tipo do dado: refere-se ao tipo daquele atributo. Ele pode ser primitivo ou de referência. Existem 8 tipos primitivos em Java.

TIPO	DESCRIÇÃO
int	Número inteiro
double	Número de ponto flueuante de dupla precisão
float	Número de ponto flutuante de precisão simples
long	Número inteiro longo
short	Número inteiro curto
byte	Número inteiro de 8 bits
boolean	Valor lógico verdadeiro ou falso
char	Caractere Unicode de 16 bits

Métodos

INSTITUTO FEDERAL
Paraná
Campus Pinhais

Métodos possuem cinco elementos básicos:

1. Visibilidade (`public`, `private` ou `protected`);
2. Tipo de retorno (primitivo ou de referência);
3. Nome (composto por um verbo);
4. Parâmetros de acordo com a necessidade;
5. Conjunto de instruções que serão executadas pelo método (conhecido como corpo).

Métodos

INSTITUTO FEDERAL
Paraná
Campus Pinhais

```
1 <especificador_de_acesso> <tipo_de_retorno> <nome_metodo>(<
      lista_parametros>) {
2     // corpo
3 }
```

- **<especificador_de_acesso>:** define a visibilidade do método.
- **<tipo_de_retorno>:** define o tipo de retorno.
- **<nome_metodo>:** é o nome do método.
- **<lista_parametros>:** parâmetros necessários para a execução do método.
- **// corpo:** é o conjunto de instruções executados pelo métdo.

Como os objetos são criados

O operador new aloca memória em tempo de execução para um objeto e retorna uma referência a ele.

```
1 Veiculo fusca; //declara uma referencia ao objeto
2 fusca = new Veiculo(); // aloca um objeto Veiculo
```

ou

```
1 Veiculo fusca = new Veiculo();
```

Exemplo

```java
public class Pessoa {
    public String nome;
    public int idade;

    public void exibirInformacoes(){
        System.out.println("Nome: " + nome);
        System.out.println("Idade: " + idade);
    }
}
```

```java
public class Principal {
    public static void main(String[] args) {
        Pessoa pessoa = new Pessoa();

        pessoa.nome = "Aleffer";
        pessoa.idade = 22;

        pessoa.exibirInformacoes();
    }

}
```

A classe Pessoa possui dois atributos:

- `nome`
- `idade`

Um método:

- `exibirInformacoes()`

Método construtor

Trata-se de um método da classe que possui o mesmo nome da classe.

```java
public class MinhaClasse {
    // Atributos da classe
    private int meuAtributo;

    // Metodo construtor
    public MinhaClasse(int valorInicial) {
        this.meuAtributo = valorInicial;
    }

    // Demais metodos e funcionalidades da classe...
}
```

Este método é automaticamente chamado quando uma nova instância da classe é criada, inicializando os atributos do objeto.

Exemplo com o método construtor

INSTITUTO FEDERAL
Paraná
Campus Pinhais

As informações dos atributos são passados como parâmetro de entrada no momento da instanciação do objeto.

```java
public class Pessoa {
    String nome;
    int idade;

    public Pessoa(String nome, int idade){
        this.nome = nome;
        this.idade = idade;
    }

    public void exibirInformacoes(){
        System.out.println("Nome: " + nome);
        System.out.println("Idade: " + idade);
    }
}
```

```java
public class Principal {
    public static void main(String[] args) {
        Pessoa pessoa = new Pessoa("Aleffer", 22);

        pessoa.exibirInformacoes();
    }

}
```