Lista 3

Wesley Sant'Anna

8 de setembro de 2025

1. Encapsulando Dados em uma Classe

   Livro.java

```java

public class Livro {
    private String titulo;
    private String autor;
    private float preco;

    // Construtor
    public Livro(String titulo, String autor, float preco) {
        this.titulo = titulo;
        this.autor = autor;
        this.preco = preco;
    }

    // Getters
    public String getTitulo() {
        return titulo;
    }

    public String getAutor() {
        return autor;
    }

    public float getPreco() {
        return preco;
    }

    // Setters
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public void setAutor(String autor) {
        this.autor = autor;
    }

    public void setPreco(float preco) {
        this.preco = preco;
    }

```

```
    // Metodo para exibir informacoes
    @Override
    public String toString() {
        return "Livro{" +
                "titulo='" + titulo + '\'' +
                ", autor='" + autor + '\'' +
                ", preco=" + preco +
                '}';
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        Livro l1 = new Livro("Senhor dos Aneis", "J.R.R Tolkien",
            59.99f);
            System.err.println(l1);
    }
}
```

2. Validaçao de Dados Usando Setters

Mundado setter preço em livro.java

```
public void setPreco(float preco) {
    if (preco > 0.0f) {
        this.preco = preco;
    } else {
        throw new IllegalArgumentException("O preco nao pode ser
            negativo ou zero!");
    }
}
```

Modificaçao na Main.java

```

public class Main {
public static void main(String[] args) {
    try {
        Livro l = new Livro("Dom Casmurro", "Machado de Assis",
            25.0f);
        l.setPreco(-10.0f);
    } catch (IllegalArgumentException e) {
        System.out.println("Erro: " + e.getMessage());
    }
}

}
```

3. Implementaçao de Propriedade Somente para Leitura

```java
public class Pessoas {
    private String nome;
    private int idade;

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


    public String getNome(){
        return nome;
    }

    public int getIdade(){
        return idade;
    }

    @Override
    public String toString(){
        return "Nome: " + nome + "\nidade: " + idade;
    }
}
```

Main.java

```java
public class Main {
    public static void main(String[] args) {

        Pessoas p1 = new Pessoas("Wesley", 22);
        System.err.println(p1);
    }
}
```

4. Encapsulamento e Ocultaçao de Informaçao

```java
public class ContaBancaria {
    private int numeroConta;
    private float saldo;
    private int senha;

    // Construtor
    public ContaBancaria(int numeroConta, int senha) {
        this.numeroConta = numeroConta;
        this.senha = senha;
        this.saldo = 0.0f; // saldo inicial
    }

    public void depositar(float valor, int senha) {
        if (senha == this.senha) {
            if (valor > 0) {
                this.saldo += valor;
                System.out.println("Dep sito realizado com
                    sucesso!");
            } else {
                System.out.println("Valor invalido para dep sito
                    .");
            }
        } else {
            System.out.println("Senha incorreta. Dep sito nao
                realizado.");
        }
    }

    public boolean sacar(float valor, int senha) {
        if (senha == this.senha) {
            if (valor > 0 && valor <= saldo) {
                saldo -= valor;
                System.out.println("Saque realizado com sucesso!"
                    );
                return true;
            } else {
                System.out.println("Saldo insuficiente ou valor
                    invalido.");
                return false;
            }
        } else {
            System.out.println("Senha incorreta. Saque nao
                realizado.");
            return false;
        }
    }

    public float getSaldo(int senha) {
        if (senha == this.senha) {
            return saldo;
        } else {
            System.out.println("Senha incorreta. Nao foi possivel
```

```
                consultar o saldo.");
47              return -1; // indica falha
48          }
49      }
50
51      public int getNumeroConta() {
52          return numeroConta;
53      }
54  }
```

Main.java

```
1   public class TestaConta {
2       public static void main(String[] args) {
3           ContaBancaria conta = new ContaBancaria(12345, 4321);
4
5           conta.depositar(500, 4321); // senha correta
6           conta.depositar(200, 1111); // senha errada
7
8           conta.sacar(100, 4321); // saque valido
9           conta.sacar(1000, 4321); // saque maior que o saldo
10          conta.sacar(50, 1111); // senha incorreta
11
12          System.out.println("Saldo final: " + conta.getSaldo(4321)
                );
13          conta.getSaldo(9999); // senha errada
14      }
15  }
```

5. Declaraçao Basica de Classe e Instanciaçao Carro.java

```java
public class Carro {
    public String marca;
    public String modelo;
    public int ano;

    public Carro(String marca, String modelo, int ano){
        this.marca = marca;
        this.modelo = modelo;
        this.ano = ano;
    }


    public void setMarca (String marca) {
        this.marca = marca;
    }

    public void setModelo (String modelo) {
        this.modelo = modelo;
    }

    public void setAno (int ano) {
        this.ano = ano;
    }

    @Override
    public String toString() {
        return "Marca: " + marca + "\nModelo: " + modelo +  "\
           nAno: " + ano;
    }

}
```

TestarCarro.java

```java
public class TestarCarro {
    public static void main(String[] args) {
        Carro c1 = new Carro("Ferrari", "Laferrari", 2023);

        System.out.println(c1);
    }
}
```

6. Adicionando Comportamento a uma Classe

Adicionando metodos referente ao atributo LigarMotor()

```java
public boolean LigarMotor() {
    this.motorLigado = true;
    return motorLigado;
}

public boolean DesligarMotor() {
    this.motorLigado = false;
    return motorLigado;
}
```

TestarCarro.java

```java
import java.util.Random;

public class TestaCarro {
    public static void main(String[] args) {
        Random rand = new Random();

        // Instanciando varios carros
        Carro c1 = new Carro("Fiat", "Uno", 2010, false);
        Carro c2 = new Carro("Volkswagen", "Gol", 2015, false);
        Carro c3 = new Carro("Toyota", "Corolla", 2020, false);

        Carro[] carros = {c1, c2, c3};

        // Ligando ou desligando motores aleatoriamente
        for (Carro carro : carros) {
            if (rand.nextBoolean()) {
                carro.LigarMotor();
                System.out.println(carro.marca + " " + carro.
                    modelo + " -> Motor ligado");
            } else {
                carro.DesligarMotor();
                System.out.println(carro.marca + " " + carro.
                    modelo + " -> Motor desligado");
            }
        }

        System.out.println("\n=== Estado final dos carros ===");
        for (Carro carro : carros) {
            System.out.println(carro);
            System.out.println("--------------------");
        }
    }
}
```

7. Sobrecarga de Construtores

Adicionando sobregarga no construtor Carro.java

```
1
2       public Carro() {
3           this.marca = "Desconhecida";
4           this.modelo =  "Desconhecida";
5           this.ano = 0;
6           this.motorLigado = false;
7       }
```

TestarCarro.java

```
1 Carro c3 = new Carro("Toyota", "Corolla", 2020, false);
2 Carro c4 = new Carro();
```

8. Sobrecarga de Metodos

```java
public class Carro {
    public String marca;
    public String modelo;
    public int ano;
    public boolean motorLigado;
    public double velocidade;

    // Construtor padrao
    public Carro() {
        this.marca = "Desconhecida";
        this.modelo = "Desconhecida";
        this.ano = 0;
        this.motorLigado = false;
        this.velocidade = 0;
    }

    // Construtor completo
    public Carro(String marca, String modelo, int ano, boolean
       motorLigado, double velocidade) {
        this.marca = marca;
        this.modelo = modelo;
        this.ano = ano;
        this.motorLigado = motorLigado;
        this.velocidade = velocidade;
    }

    // Metodos ligar/desligar motor
    public boolean ligarMotor() {
        this.motorLigado = true;
        return motorLigado;
    }

    public boolean desligarMotor() {
        this.motorLigado = false;
        this.velocidade = 0;
        return motorLigado;
    }

    //  Sobrecarga de metodos acelerar
    public double acelerar() {
        return acelerar(10);
    }

    public double acelerar(double valor) {
        if (motorLigado) {
            this.velocidade += valor;
        } else {
            System.out.println("Nao e possivel acelerar: motor
               desligado.");
        }
        return velocidade;
    }
```

```
51
52      //  Sobrecarga de metodos frear
53      public double frear() {
54          return frear(10);
55      }
56
57      public double frear(double valor) {
58          this.velocidade -= valor;
59          if (velocidade < 0) velocidade = 0;
60          return velocidade;
61      }
62
63      @Override
64      public String toString() {
65          return "Marca: " + marca +
66                 "\nModelo: " + modelo +
67                 "\nAno: " + ano +
68                 "\nMotor: " + (motorLigado ? "Ligado" : "Desligado
                       ") +
69                 "\nVelocidade: " + velocidade + " km/h";
70      }
71  }
```

TestarCarro.java

```
1   import java.util.ArrayList;
2
3   public class TestaCarro {
4       public static void main(String[] args) {
5           ArrayList<Carro> carros = new ArrayList<>();
6
7           // Criando alguns carros
8           carros.add(new Carro("Fiat", "Uno", 2010, false, 0));
9           carros.add(new Carro("Volkswagen", "Gol", 2015, false, 0)
                );
10          carros.add(new Carro("Toyota", "Corolla", 2020, false, 0)
                );
11
12          // Exercitando metodos
13          for (Carro carro : carros) {
14              System.out.println("\n=== Novo Carro ===");
15              System.out.println(carro);
16
17              // Liga motor
18              carro.ligarMotor();
19
20              // Acelera com valor padrao
21              carro.acelerar();
22              System.out.println("Ap s acelerar padrao: " + carro.
                    velocidade + " km/h");
23
24              // Acelera com valor especificado
25              carro.acelerar(30);
26              System.out.println("Ap s acelerar +30 km/h: " +
```

```
                carro.velocidade + " km/h");
27
28              // Freia com valor padrao
29              carro.frear();
30              System.out.println("Ap s frear padrao: " + carro.
                    velocidade + " km/h");
31
32              // Freia com valor especificado
33              carro.frear(15);
34              System.out.println("Ap s frear 15 km/h: " + carro.
                    velocidade + " km/h");
35
36              // Desliga motor
37              carro.desligarMotor();
38              System.out.println("Ap s desligar motor:");
39              System.out.println(carro);
40          }
41      }
42  }
```

Referências

[1] H. M. Deitel and P. J. Deitel. *Java How To Program: Early Objects*. Prentice Hall, Boston, 11 edition, 2020.

[2] H. M. Deitel and P. J. Deitel. *Java How To Program: Late Objects*. Prentice Hall, Boston, 11 edition, 2020.

[3] Cay Horstmann and Gary Cornell. *Core Java, Volume I: Fundamentals*. Pearson, New York, 11 edition, 2018.

[4] Cay Horstmann and Gary Cornell. *Core Java, Volume II: Advanced Features*. Pearson, New York, 11 edition, 2018.

[5] Ricardo Santos. *Introdução à Programação Orientada a Objetos usando Java*. Campus/Elsevier, Rio de Janeiro, 2 edition, 2013.

[6] Herbert Schildt. *Java: The Complete Reference*. Oracle Press, New York, 9 edition, 2014.