

Sobrescrita de Métodos

Sobrescrita (ou override) → Conceito relacionado a herança

A sobrescrita de métodos consiste basicamente em criar um novo método na classe filha contendo a mesma assinatura e mesmo tipo de retorno do método sobrescrito. Os comportamentos desse método em geral são diferentes.

Atleta.java

```
import java.text.DecimalFormat;
import java.text.NumberFormat;

import javax.swing.JOptionPane;

public class Atleta {
    //Criação de atributos
    private int codigo;
    private String nome;
    private String esporte;
    private double valorPatrocinio;

    //Get e Set - Métodos Acessores
    public int getCodigo() {
        return this.codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNome() {
        return this.nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getEsporte() {
        return this.esporte;
    }
    public void setEsporte(String esporte) {
        this.esporte = esporte;
    }
}
```

```

    }

    public double getValorPatrocinio() {
        return this.valorPatrocinio;
    }

    public void setValorPatrocinio(double valorPatrocinio) {
        this.valorPatrocinio = valorPatrocinio;
    }

    public void inserirDados() {
        this.codigo =
Integer.parseInt(JOptionPane.showInputDialog("Digite o código: "));
        this.nome = JOptionPane.showInputDialog("Digite o nome: ");
        this.esporte = JOptionPane.showInputDialog("Digite o esporte
praticado: ");
        this.valorPatrocinio =
Double.parseDouble(JOptionPane.showInputDialog("Digite o valor do patrocínio:
"));
    }

    public double atualizarValor(double taxa) {
        this.valorPatrocinio += this.valorPatrocinio * taxa / 100;
        return this.valorPatrocinio;
    }

    public void mostrar() {
        String mensagem = "Objeto atleta criado a partir da classe
Atleta";

        mensagem += "\nCódigo: " + this.codigo;
        mensagem += "\nNome: " + this.nome;
        mensagem += "\nEsporte: " + this.esporte;

        NumberFormat formatar = DecimalFormat.getCurrencyInstance();

        mensagem += "\nValor de Patrocínio: " +
formatar.format(this.valorPatrocinio);
        JOptionPane.showMessageDialog(null, mensagem);
    }
}

```

Corredor.java

```
public class Corredor extends Atleta{

    private double velocidadeMedia;

    public double getVelocidadeMedia() {
        return this.velocidadeMedia;
    }

    public void setVelocidadeMedia(double velocidadeMedia) {
        this.velocidadeMedia = velocidadeMedia;
    }

    @Override
    public double atualizarValor(double taxa) {
        /* Veja a palavra reservada Super, ela indica a superclasse ou
        classe mãe.
        * Isso indica que, o objeto do tipo Corredor tem um método que
        atualiza seu patrocínio.
        * Ele chama o próprio método e multiplica por 2
        * */
        return super.atualizarValor(2 * taxa);
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        Atleta atleta = new Atleta();
        atleta.setValorPatrocinio(1000);
        Corredor corredor = new Corredor();
        corredor.setValorPatrocinio(1000);
        int porcentagemAumento = 15;
        atleta.atualizarValor(porcentagemAumento);
        corredor.atualizarValor(porcentagemAumento);

        System.out.println("Salário atleta: " +
        atleta.getValorPatrocinio());
        System.out.println("Salário corredor: " +
        corredor.getValorPatrocinio());
    }
}
```