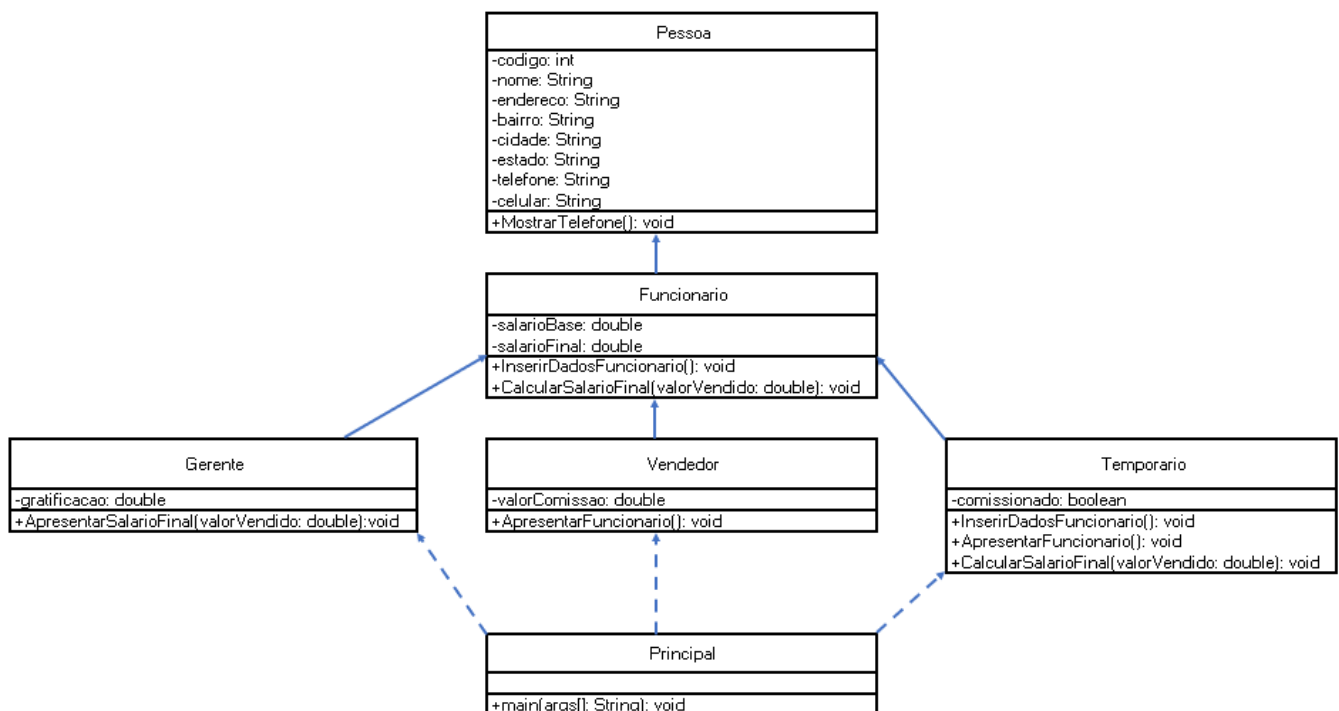


Orientação a objetos - Sobrescrita

A sobre escrita de métodos está diretamente relacionada com a herança e consiste em reescrever um método herdado, mudando seu comportamento, mas mantendo exatamente a assinatura. Para exemplificar, utilizaremos o cálculo do salário final dos funcionários conforme diagrama abaixo:



Na superclasse Funcionário, foi implementado o método **CalcularSalarioFinal** considerando uma regra geral de cálculo (somar 10% do valor vendido ao salário base). Consequentemente, esse método foi herdado por suas subclasses **Gerente**, **Vendedor** e **Temporário**. O problema é que, de acordo com as normas da empresa, o cálculo do salário dos gerentes e dos temporários é diferente. Para os vendedores, o método **CalcularSalarioFinal** herdado está correto, porém, para **Gerente** e **Temporário**, não. O problema pode ser solucionado com a **sobrescrita** dos métodos **CalcularSalarioFinal** nas classes **Gerente** e **Temporário**. Na classe **Funcionário**, o método foi codificado da seguinte forma:

```

public void CalcularSalarioFinal(double valorVendido) {
    this.setSalarioFinal( this.getSalarioBase() + (valorVendido * 0.1) );
}
  
```

Já na subclasse Gerente fica da seguinte maneira:

```
public void CalcularSalarioFinal(double valorVendido) {
    double meta = Double.parseDouble( JOptionPane.showInputDialog(
        "Digite a meta de vendas do mês: "
    ));

    if (valorVendido > meta) {
        this.setGratificacao(valorVendido * 0.15);
    } else {
        this.setGratificacao(0);
    }
    this.setSalarioFinal( this.getSalarioBase() + this.getGratificacao() );
}
```

E na subclasse Temporário ficou da seguinte forma:

```
public void CalcularSalarioFinal(double valorVendido) {
    if (this.comissionado) {
        this.setSalarioFinal( this.getSalarioBase() + (valorVendido * 0.1) );
    } else {
        this.setSalarioFinal( this.getSalarioBase() );
    }
}
```

Os Objetos que representarão um gerente, um vendedor e um temporário serão instanciados a partir das classes Gerente, Vendedor e Temporário. Nesse momento (da criação dos objetos), o Java considera a estrutura de cada subclasse, a qual dá origem ao objeto. Então, um objeto do tipo Gerente considera a codificação do método CalcularSalarioFinal da classe Gerente; um do tipo Temporário leva em conta a codificação da classe Temporário, e um do tipo Vendedor focaliza o método herdado, mas todos são invocados exatamente da mesma forma, conforme mostra abaixo:

```
Gerente gerente = new Gerente();
Vendedor vendedor = new Vendedor();
Temporario temporario = new Temporario();

gerente.CalcularSalarioFinal(5000);
vendedor.CalcularSalarioFinal(5000);
temporario.CalcularSalarioFinal(5000);
```

A sobrescrita de métodos também proporciona vantagens relacionadas ao gerenciamento polimórfico de objetos.

Exemplo-1: Todo professor de mestrado deve ser um professor de graduação. Os professores de graduação possuem os seguintes atributos: **matrícula, nome, salário bruto e quantidade de disciplinas**.

Além dos atributos de um professor de graduação, os professores de mestrado também possuem: **ano do término do doutorado, quantidade de artigos científicos escritos**.

O salário de um professor de graduação é calculado da seguinte forma:

$$= \text{salário bruto} - (\text{salário bruto} * 0.2) + \text{quantidade de disciplinas} * 50$$

O salário de um professor de mestrado é calculado da seguinte forma:

$$= \text{salário do professor de graduação} + \text{quantidade de artigos científicos} * 150$$

Na classe do método main, instancie um objeto do tipo professor de graduação, e outro do tipo professor de mestrado, e exiba o salário líquido de cada um.

```
package exemplosobrescrita1;

public class Graduacao {
    // declarando os atributos da classe
    public double salProfessor;
    public int matricula;
    public String nome;
    public double salBruto;

    // metodo da classe
    public void CalculaSalario() {
        this.salProfessor = this.salBruto - (this.salBruto * 0.2);
    }
}
```



```
package exemplosobrescrita1;

public class ProfessorGraduacao extends Graduacao {
    public double qtdeDisc;

    // metodo da classe sendo sobrescrita
    // quer dizer, manter a assinatura do metodo (nome e parametro)
    // reescrever o conteudo;
    @Override
    public void CalculaSalario() {

        super.salProfessor += this.qtdeDisc * 50;

    }
}
```

```
package exemplosobrescrita1;

public class ProfessorMestrado extends Graduacao {
    public String anoTermDoc;
    public double qtdeArtigos;

    // metodo da classe sendo sobrescrita
    // quer dizer, manter a assinatura do metodo (nome e parametro)
    // reescrever o conteudo;
    @Override
    public void CalculaSalario() {

        this.salProfessor += this.qtdeArtigos * 150;

    }
}
```

```
package exemplosobrescrita1;

import javax.swing.JOptionPane;

public class Principal {

    public static void main(String[] args) {
        ProfessorGraduacao profGrad = new ProfessorGraduacao();
        ProfessorMestrado profMes = new ProfessorMestrado();

        String aux;
        profGrad.matricula = 1257835;
        profGrad.nome = "Professor de graduação";
        profGrad.qtdeDisc = 20;
        profGrad.salBruto = 3200;
        profGrad.CalculaSalario();
        // preparando a impressao para o prof. de graduacao
        aux = "\nDados Professor graduação\n";
        aux += "Matricula: " + profGrad.matricula + "\n";
        aux += "Nome prof: " + profGrad.nome + "\n";
        aux += "Qtd Disc.: " + profGrad.qtdeDisc + "\n";
        aux += "Sal. Prof: R$ " + profGrad.salProfessor + "\n";

        profMes.matricula = 98765;
        profMes.nome = "Professor de Mestrado";
        profMes.anoTermDoc = "2 anos";
        profMes.qtdeArtigos = 15;
        profMes.salBruto = 4200;
        profMes.CalculaSalario();
        // preparando a impressao para o prof. de graduacao
        aux += "\n\nDados Professor mestrado\n";
        aux += "Matricula: " + profMes.matricula + "\n";
        aux += "Nome prof: " + profMes.nome + "\n";
        aux += "Qtd Artigos: " + profMes.qtdeArtigos + "\n";
        aux += "Term. Docto: " + profMes.anoTermDoc + "\n";
        aux += "Sal. Prof: R$ " + profMes.salProfessor + "\n";

        // impressao
        JOptionPane.showMessageDialog(null, aux);
    }
}
```

Exemplo-2: Desenvolva um projeto em Java que leia um número inteiro e imprima se o nro escolhido aleatoriamente pela classe Random é um nro primo.

Desenvolva um método em uma subclasse praticando a sobrescrita:

- Antecessor primo e seu sucessor também um nro primo.

```
package exemplosobrescrita2;

public class NroPrimos {

    // metodo da classe
    public String CalculaSePrimo(int nro) {
        int i, cont=0;

        for (i = 1; i <= nro; i++) {
            if (nro % i == 0) {
                cont++;
            }
        }
        if (cont <= 2) {
            return "O nro: " + nro + " é primo!";
        } else {
            return "O nro: " + nro + " não é primo!";
        }
    }
}
```

```
package exemplosobrescrita2;

public class CalculaPrimoAntecessor extends NroPrimos {

    // praticando a sobrescrita
    @Override
    public String CalculaSePrimo(int nro) {
        int i, cont;

        while (true) {
            cont = 0;
            for (i = 1; i <= nro; i++) {
                if (nro % i == 0) {
                    cont++;
                }
            }
            if (cont <= 2) {
                return "O nro: " + nro + " antecessor é primo!";
            } else {
                nro -= 1;
            }
        }
    }
}
```

```
package exemplosobrescrita2;

public class CalculaPrimoSucessor extends NroPrimos {

    // praticando a sobrescrita
    @Override
    public String CalculaSePrimo(int nro) {
        int i, cont;

        while (true) {
            cont = 0;
            for (i = 1; i <= nro; i++) {
                if (nro % i == 0) {
                    cont++;
                }
            }
            if (cont <= 2) {
                return "O nro: " + nro + " sucessor é primo!";
            } else {
                nro += 1;
            }
        }
    }
}
```



```
/*
    Desenvolva um projeto em Java que leia um número inteiro e imprima se
    o nro digitado pelo usuario é um nro primo.

    Desenvolva um metodo em uma subclasse praticando a sobrescrita:
    - antecessor primo e seu sucessor também um nro primo.
*/
package exemplosobrescrita2;

import java.util.Random;
import javax.swing.JOptionPane;

public class Principal {

    public static void main(String[] args) {
        CalculaPrimoAntecessor calcA = new CalculaPrimoAntecessor();
        CalculaPrimoSucessor calcS = new CalculaPrimoSucessor();
        Random rand = new Random();

        int nro = rand.nextInt(1000);
        String aux = "";

        aux = "Nro sorteado: " + nro + "\n";
        aux += calcA.CalculaSePrimo(nro) + "\n";
        aux += calcS.CalculaSePrimo(nro) + "\n";
        JOptionPane.showMessageDialog(null, aux);

    }

}
```