Orientação a Objetos 1

Sobreposição

Prof. Dr. Vinícius Camargo Andrade

vcandrade@utfpr.edu.br

Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná

Sobreposição

Override

Sobreposição

É a redefinição na classe filha de um método definido na classe pai.

Sobreposição

Esta operação anula (sobrepõem) o método da classe pai

Sobreposição

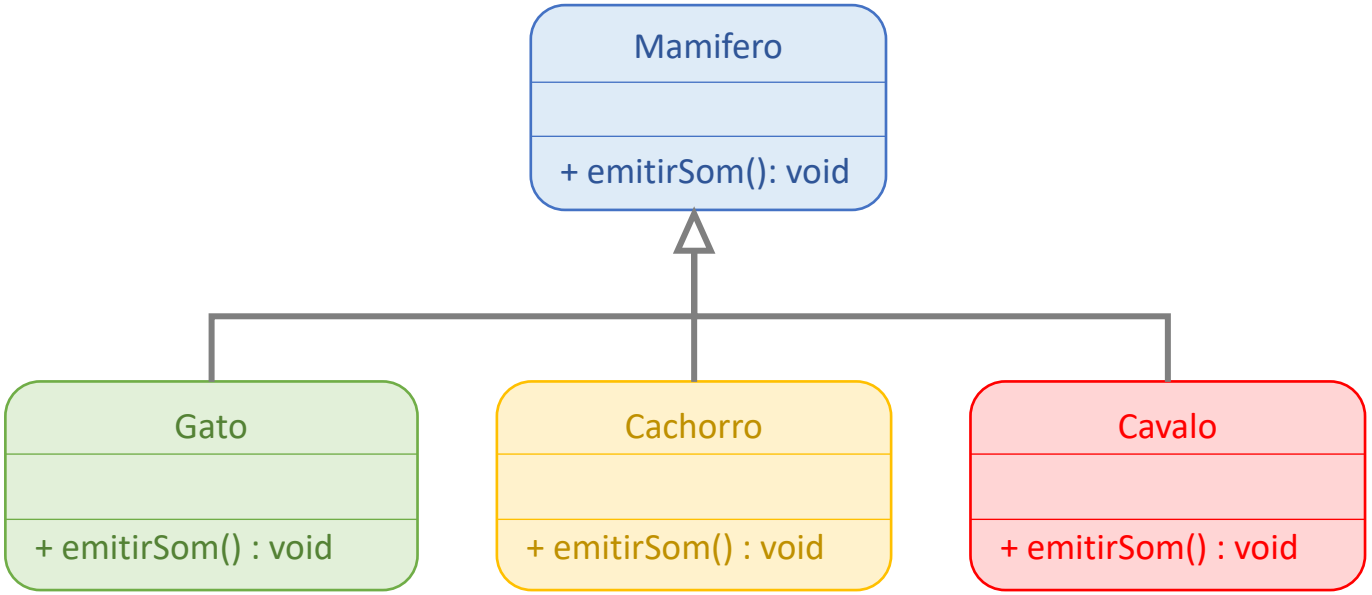
O conceito de sobreposição possibilita a uma única variável chamar métodos diferentes, dependendo do que a variável contém.

Sobreposição

Ou seja, permite escrever programas de forma genérica.

Exemplo 1

Exemplo 1


Mamifero
+ emitirSom(): void
Gato
+ emitirSom() : void
Cachorro
+ emitirSom() : void
Cavalo
+ emitirSom() : void

Exemplo 1

```java
public class Mamifero {

    public void emitirSom() {

    }
}
```

```java
public class Gato extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Miar");
    }
}
```

```java
public class Cachorro extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Latir");
    }
}
```

```java
public class Cavalo extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Relinchar");
    }
}
```

Exemplo 1

```java
public class Mamifero {

    public void emitirSom() {

    }
}
```

```java
public class Gato extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Miar");
    }
}
```

```java
public class Cachorro extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Latir");
    }
}
```

```java
public class Cavalo extends Mamifero {

    @Override
    public void emitirSom() {

        System.out.println("Relinchar");
    }
}
```

@Override

A notação @Override deixa explícito ao interpretador que o método em questão substituirá o método declarado na superclasse.

```java
public class MamiferoTeste {

    public static void main(String[] args) {

        String opcao = "Cachorro";

        Mamifero mamifero = null;

        switch (opcao) {
            case "Gato":
                mamifero = new Gato();
                break;

            case "Cachorro":
                mamifero = new Cachorro();
                break;

            case "Cavalo":
                mamifero = new Cavalo();
                break;

            default:
                System.out.println("Opção inválida.");
        }
        mamifero.emitirSom();
    }
}
```

```
run:
Latir
BUILD SUCCESSFUL (total time: 0 seconds)
```

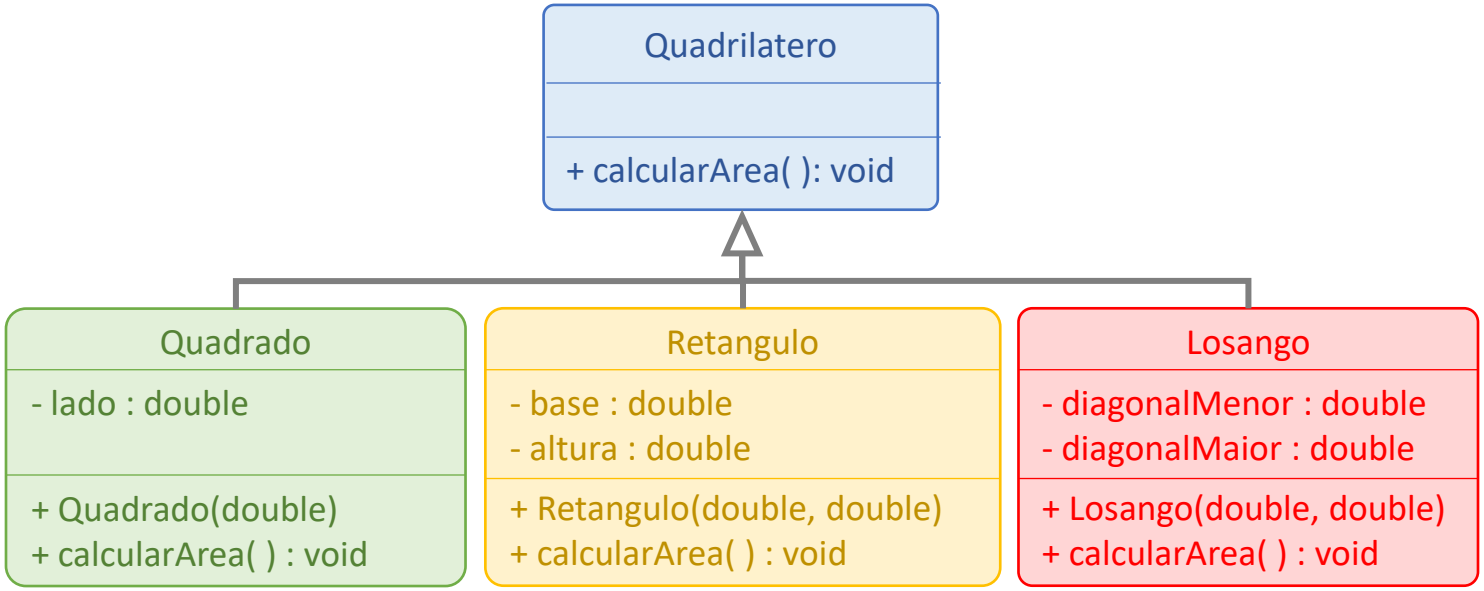
Exercício

Exercício

O Quadrado, Retângulo e Losango são definidos como Quadriláteros, porém cada um possui uma fórmula específica para calcular sua respectiva área.

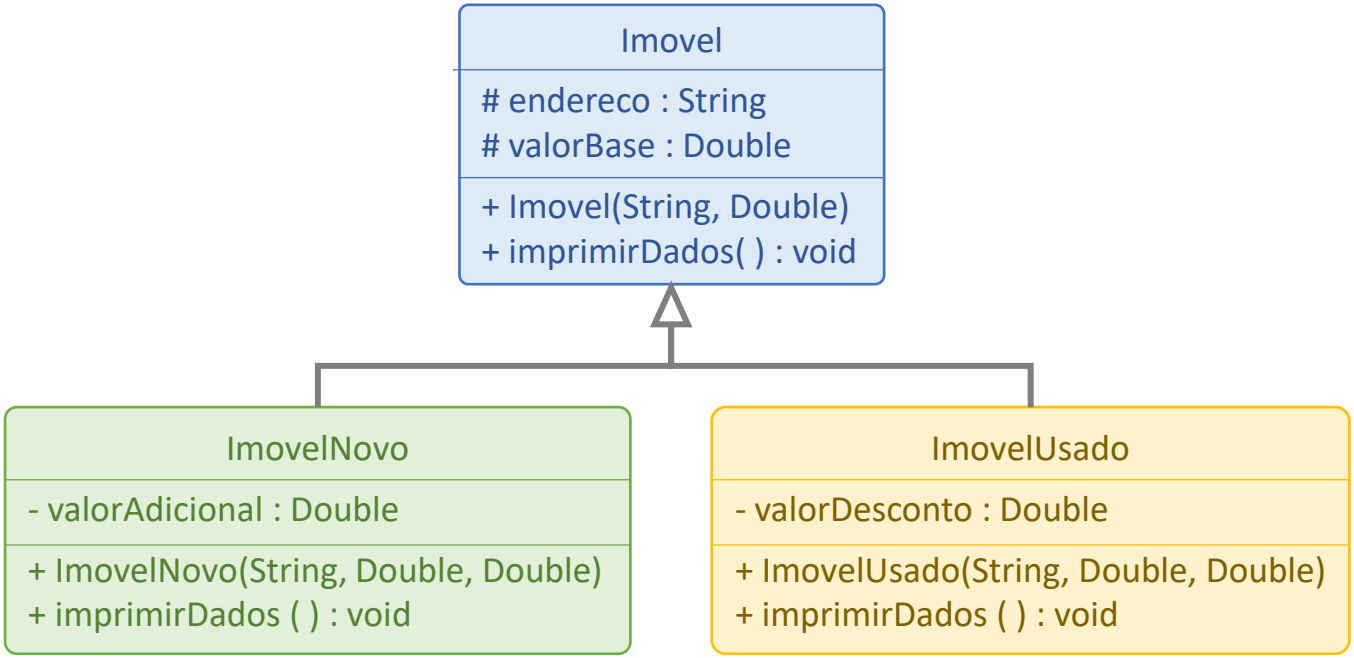
Codifique o cálculo da área de cada quadrilátero de acordo com o modelo do próximo slide.

Exercício

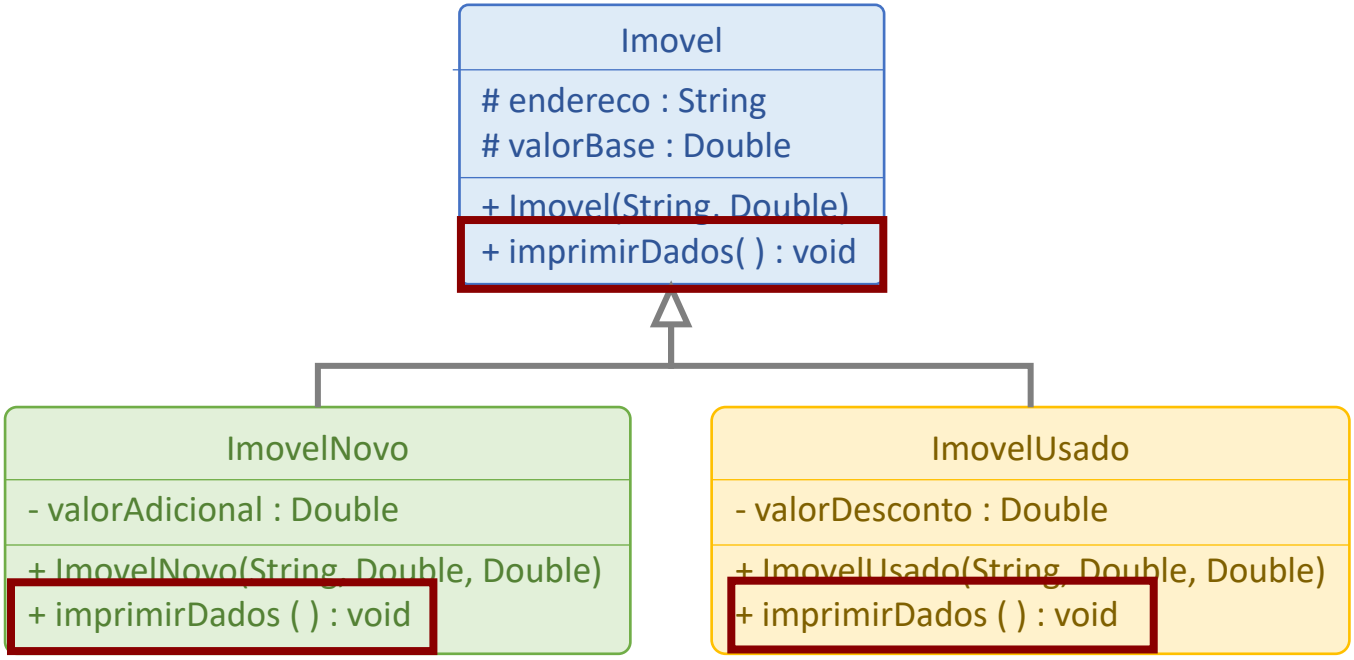

Quadrilatero
+ calcularArea( ): void
Quadrado
- lado : double
+ Quadrado(double)
+ calcularArea( ) : void
Retangulo
- base : double
- altura : double
+ Retangulo(double, double)
+ calcularArea( ) : void
Losango
- diagonalMenor : double
- diagonalMaior : double
+ Losango(double, double)
+ calcularArea( ) : void

Exemplo 2

Exemplo 2


Imovel
endereco : String
valorBase : Double
+ Imovel(String, Double)
+ imprimirDados( ) : void
ImovelNovo
- valorAdicional : Double
+ ImovelNovo(String, Double, Double)
+ imprimirDados ( ) : void
ImovelUsado
- valorDesconto : Double
+ ImovelUsado(String, Double, Double)
+ imprimirDados ( ) : void

Exemplo 2


Imovel
endereco : String
valorBase : Double
+ Imovel(String, Double)
+ imprimirDados( ) : void
ImovelNovo
- valorAdicional : Double
+ ImovelNovo(String, Double, Double)
+ imprimirDados ( ) : void
ImovelUsado
- valorDesconto : Double
+ ImovelUsado(String, Double, Double)
+ imprimirDados ( ) : void

Implementação

Im

```java
public class ImovelNovo extends Imovel {

    private Double valorAdicional;

    public ImovelNovo(String endereco, Double valorBase, Double valorAdicional) {

        super(endereco, valorBase);
        this.valorAdicional = valorAdicional;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
        System.out.println("Valor Adicional: " + this.getValorAdicional());
        System.out.println("Valor Final: " + (super.valorBase + this.valorAdicional));
    }
```

```java
public class ImovelUsado extends Imovel {

    private Double valorDesconto;

    public ImovelUsado(String endereco, Double valorBase, Double valorDesconto) {

        super(endereco, valorBase);
        this.valorDesconto = valorDesconto;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
        System.out.println("Valor Desconto: " + this.getValorDesconto());
        System.out.println("Valor Final: " + (super.valorBase - this.valorDesconto));
    }
```

Im

```java
public class ImovelNovo extends Imovel {

    private Double valorAdicional;

    public ImovelNovo(String endereco, Double valorBase, Double valorAdicional) {

        super(endereco, valorBase);
        this.valorAdicional = valorAdicional;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
        System.out.println("Valor Adicional: " + this.getValorAdicional());
        System.out.println("Valor Final: " + (super.valorBase + this.valorAdicional));
    }
```

```java
public class ImovelUsado extends Imovel {

    private Double valorDesconto;

    public ImovelUsado(String endereco, Double valorBase, Double valorDesconto) {

        super(endereco, valorBase);
        this.valorDesconto = valorDesconto;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
        System.out.println("Valor Desconto: " + this.getValorDesconto());
        System.out.println("Valor Final: " + (super.valorBase - this.valorDesconto));
    }
```

Implementação

```java
public class Imovel {

    protected String endereco;
    protected Double valorBase;

    public Imovel(String endereco, Double valorBase) {

        this.endereco = endereco;
        this.valorBase = valorBase;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
    }
```

Implementação

```java
public class Imovel {

    protected String endereco;
    protected Double valorBase;

    public Imovel(String endereco, Double valorBase) {

        this.endereco = endereco;
        this.valorBase = valorBase;
    }

    public void imprimirDados() {

        System.out.println("Endereço : " + this.getEndereco());
        System.out.println("Valor Base: " + this.getValorBase());
    }
```

Im

```java
public class ImovelNovo extends Imovel {

    private Double valorAdicional;

    public ImovelNovo(String endereco, Double valorBase, Double valorAdicional) {

        super(endereco, valorBase);
        this.valorAdicional = valorAdicional;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Adicional: " + this.getValorAdicional());
        System.out.println("Valor Final: " + (super.valorBase + this.valorAdicional));
    }
```

```java
public class ImovelUsado extends Imovel {

    private Double valorDesconto;

    public ImovelUsado(String endereco, Double valorBase, Double valorDesconto) {

        super(endereco, valorBase);
        this.valorDesconto = valorDesconto;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Desconto: " + this.getValorDesconto());
        System.out.println("Valor Final: " + (super.valorBase - this.valorDesconto));
    }
```

```java
public class ImovelNovo extends Imovel {

    private Double valorAdicional;

    public ImovelNovo(String endereco, Double valorBase, Double valorAdicional) {

        super(endereco, valorBase);
        this.valorAdicional = valorAdicional;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Adicional: " + this.getValorAdicional());
        System.out.println("Valor Final: " + (super.valorBase + this.valorAdicional));
    }
```

```java
public class ImovelUsado extends Imovel {

    private Double valorDesconto;

    public ImovelUsado(String endereco, Double valorBase, Double valorDesconto) {

        super(endereco, valorBase);
        this.valorDesconto = valorDesconto;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Desconto: " + this.getValorDesconto());
        System.out.println("Valor Final: " + (super.valorBase - this.valorDesconto));
    }
```

```java
public class ImovelNovo extends Imovel {

    private Double valorAdicional;

    public ImovelNovo(String endereco, Double valorBase, Double valorAdicional) {

        super(endereco, valorBase);
        this.valorAdicional = valorAdicional;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Adicional: " + this.getValorAdicional());
        System.out.println("Valor Final: " + (super.valorBase + this.valorAdicional));
    }
```

```java
public class ImovelUsado extends Imovel {

    private Double valorDesconto;

    public ImovelUsado(String endereco, Double valorBase, Double valorDesconto) {

        super(endereco, valorBase);
        this.valorDesconto = valorDesconto;
    }

    @Override
    public void imprimirDados() {

        super.imprimirDados();
        System.out.println("Valor Desconto: " + this.getValorDesconto());
        System.out.println("Valor Final: " + (super.valorBase - this.valorDesconto));
    }
```

Exercício 2

Exercício 2

Desenvolva um sistema para calcular o valor total de imposto pago por pessoas físicas e jurídicas.
Sabe-se que uma pessoa física é identificada pelo seu nome, renda anual e gasto com saúde. A pessoa jurídica é identificada pelo seu nome, renda anual e número de funcionários.

Exercício 2

As regras para o cálculos são:

- *Pessoa Física:*
  - *Renda anual abaixo de R$ 20.000,00, pagam 15% de imposto. R$ 20.000,00 ou mais, 25%.*
  - *Se a pessoa teve gasto com saúde, 50% destes gastos devem ser abatidos no imposto.*
- *Pessoa Jurídica:*
  - *Pagam 18% de imposto, porém se a empresa possuir mais de 15 funcionários, a empresa paga 13% de imposto.*