

Orientação a Objetos 1

Upcasting e Downcasting

Prof. Dr. Vinícius Camargo Andrade

vcandrade@utfpr.edu.br

Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná

Motivação

Motivação

*Considere um sistema de gerenciamento de **imóveis** novos e usados.*

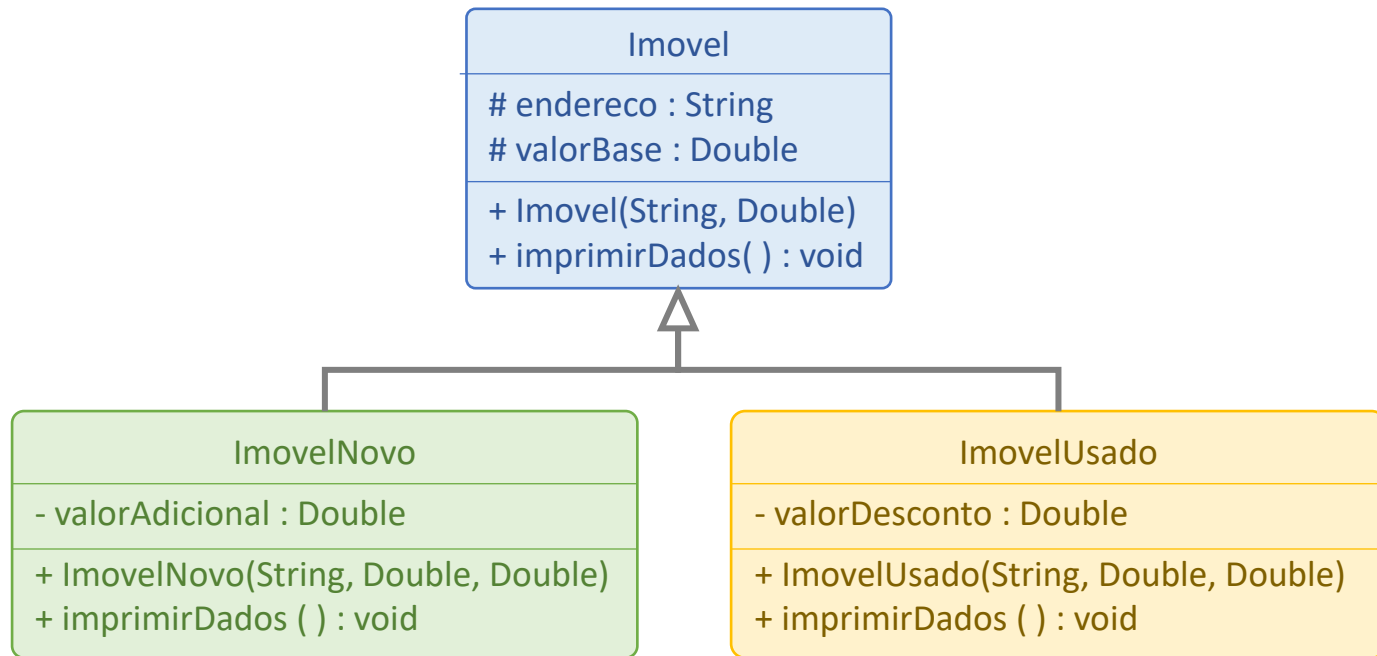


Motivação

*Ambos os imóveis possuem um **endereço** e um **valor base** e devem **imprimir seus dados**. Porém, o imóvel novo possui um **valor adicional**, enquanto o imóvel usado possui um **valor de desconto**.*



Motivação



Implementação

Implementação

```
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());  
    }  
  
    public String getEndereco() {  
        return endereco;  
    }  
}
```

Implementação

```
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));
    }

    public Double getValorAdicional() {
        return valorAdicional;
    }
}
```


Implementação

```
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));
    }

    public Double getValorDesconto() {
        return valorDesconto;
    }
}
```

Upcasting

Upcasting

Casting da subclasse para superclasse.

Upcasting

```
public class ImovelTeste {  
    public static void main(String[] args) {  
        // UPCASTING  
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);  
        imovel1.imprimirDados();  
  
        Imovel imovel2 = new ImovelUsado("Balduino Taques, 3021", 500000.00, 20000.00);  
        imovel2.imprimirDados();  
    }  
}
```

Upcasting

```
public class ImovelTeste {  
  
    public static void main(String[] args) {  
  
        // UPCASTING  
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);  
        imovel1.imprimirDados();  
  
        Imovel imovel2 = new ImovelUsado("Balduino Taques, 3021", 500000.00, 20000.00);  
        imovel2.imprimirDados();  
    }  
}
```

Downcasting

Downcasting

Casting da superclasse para subclasse;

Downcasting

*Palavra-chave **instanceof***

Downcasting

```
public class ImovelTeste {  
  
    public static void main(String[] args) {  
  
        // UPCASTING  
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);  
        Imovel imovel2 = new ImovelUsado("Balduino Taques, 3021", 500000.00, 20000.00);  
  
        // DOWNCASTING  
        ImovelNovo imovelNovo1 = (ImovelNovo) imovel1;  
        imovelNovo1.imprimirDados();  
  
        ImovelUsado imovelUsado1 = (ImovelUsado) imovel2;  
        imovelUsado1.imprimirDados();  
    }  
}
```

Downcasting

```
public class ImovelTeste {  
  
    public static void main(String[] args) {  
  
        // UPCASTING  
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);  
        Imovel imovel2 = new ImovelUsado("Balduino Taques, 3021", 500000.00, 20000.00);  
  
        // DOWNCASTING  
        ImovelNovo imovelNovo1 = (ImovelNovo) imovel1;  
        imovelNovo1.imprimirDados();  
  
        ImovelUsado imovelUsado1 = (ImovelUsado) imovel2;  
        imovelUsado1.imprimirDados();  
    }  
}
```

Problema

Downcasting

Problema

Ao fazer um *downcasting*, pode ocorrer do objeto instanciado *não coincidir* com o mesmo tipo de objeto que o receberá.



Problema

```
public class ImovelTeste {  
  
    public static void main(String[] args) {  
  
        // UPCASTING  
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);  
  
        // DOWNCASTING  
        ImovelNovo imovelNovo1 = (ImovelNovo) imovel1;  
        imovelNovo1.imprimirDados();  
  
        ImovelUsado imovelUsado1 = (ImovelUsado) imovel1;  
        imovelUsado1.imprimirDados();  
    }  
}
```

Input - FundamentosOO (run) x

run:
Endereço : Vicente Machado, 1010
Valor Base: 500000.0
Valor Adicional: 30000.0
Valor Final: 530000.0
Exception in thread "main" java.lang.ClassCastException: class orientacaoobjeto.aula10.ImovelNovo
 at orientacaoobjeto.aula10.ImovelTeste.main(ImovelTeste.java:19)
C:\Users\vinic\AppData\Local\NetBeans\Cache\11.0\executor-snippets\run.xml:111: The follow
C:\Users\vinic\AppData\Local\NetBeans\Cache\11.0\executor-snippets\run.xml:68: Java return
BUILD FAILED (total time: 0 seconds)

Solução

Downcasting

Solução

*Antes de fazer **downcasting**, deve-se realizar uma verificação se o objeto é instância da mesma classe que irá recebê-lo.*



Solução

*Para isso, utiliza-se a palavra reservada **instanceof**.*


```

public class ImovelTeste {

    public static void main(String[] args) {

        // UPCASTING
        Imovel imovel1 = new ImovelNovo("Vicente Machado, 1010", 500000.00, 30000.00);

        // DOWNCASTING
        if (imovel1 instanceof ImovelNovo) {

            System.out.println("imovel1 é instância de ImovelNovo");
            ImovelNovo imovelNovo1 = (ImovelNovo) imovel1;
            imovelNovo1.imprimirDados();

        }
        if (imovel1 instanceof ImovelUsado) {

            System.out.println("imovel1 é instância de ImovelUsado");
            ImovelUsado imovelUsado1 = (ImovelUsado) imovel1;
            imovelUsado1.imprimirDados();

        }

    }

}

```

```

run:
imovel1 é instância de ImovelNovo
Endereço : Vicente Machado, 1010
Valor Base: 500000.0
Valor Adicional: 30000.0
BUILD SUCCESSFUL (total time: 0 seconds)

```