

13.1 - UPCasting de referência

UPCasting é um tipo de Coersão por cima

UP = Cima

Casting = Coersão

Mecanismo Para
Forçarmos ou setar,
Para moldarmos
Um objeto que
a gente instanciou
Pra ele ser enxerga-
do como uma super-
classe.

Conta

↑

Conta investimento

↑

Conta especial

Exemplo de UPCasting

Conta especial conta1 = new Conta especial()

Conta conta2 = (Conta) conta1

O Casting no objeto não tem risco de
Perder Precisão

O UPCasting precisa estar na Hierarquia de
classes para funcionar. É uma forma de
referenciar um objeto da Hierarquia das
super classes utilizando uma classe mais
acima da Hierarquia.

13.2 - O problema que o Polimorfismo resolve

Ele resolve a necessidade de lidar com diferentes tipos de objetos de forma transparente, sem a necessidade de conhecer detalhes específicos de cada tipo.

13.3 - Entendendo o Polimorfismo

Polimorfismo é a capacidade de um objeto ser enxergado de diferentes formas

Poli = muitas

Morfismos = formas

isso só é possível por causa do mecanismo de Herança

13.4 DownCasting de referência

É o ato de transformar uma superclass em uma subclasse.

#Evitar

Se for fazer DownCasting, o cast tem que ser de forma explícita. Sempre validar as instâncias.

13.5 operador instanceof

O `instanceof` serve para fazer uma comparação lógica para verificar se um objeto de referência é de um tipo específico que queremos.

implementação:

```
if (conta instanceof ContaInvestimento) {
```

↳ é uma?

Obs: Busca na hierarquia para saber se é uma

13.6 Pattern Matching Para o operador instanceof

Funcionalidade a partir do Java 16

```
if (conta instanceof ContaInvestimento contaInv) {
```

Pergunta e se faz o cast

13.7 Evitando o uso do instanceof

O `instanceof` não é uma boa prática, porque ele vai de contra ao polimorfismo

Formas alternativa a instanceof:

1) Sobrecarga \rightarrow Problema: não aproveita tanto o poder do Polimorfismo

2) Criação de método na superclasse, booleano, para determinar a ação de regra de negócio e sobrescrever na subclasse do instanceof esse mesmo método.

13.8 Criando um Projeto de Faturamento

~ Projeto ~

13.9 Classes Abstratas

Classes abstratas são classes que não são concretas, logo, elas não podem ser instanciadas. Para transformar uma classe em abstrata, usa-se o termo "abstract" depois do public.

Exemplo:

```
public abstract class NotaFiscal {  
    {
```

Classes abstratas fazem sentido na reutilização do código.

13.10 Métodos Abstratos

Métodos Abstratos são métodos sem escopo, apenas com a assinatura do método. Isso se dá quando temos um método que é obrigado a ser implementado pelas subclasses.

Métodos abstratos só podem ser criados em classes abstratas.

As subclasses que herdam a superclasse, têm a obrigatoriedade de implementar o método abstrato, ou, transformar-se em classe abstrata também.