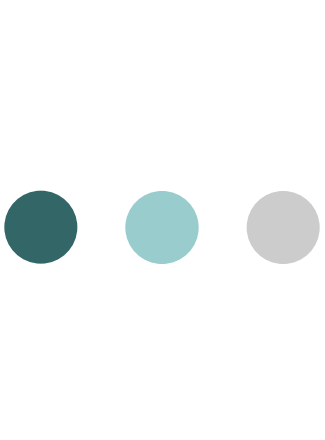


Instance Of /Casting

MATA 55

Rita Suzana

“instanceof”

- Em algumas ocasiões
  - Não temos acesso a hierarquia de classes
  - Funcionalidades são especificas da aplicação e não da hierarquia de classes

- “variável referência” instanceof “nome da classe”
  - Retorna true or false

Polimorfismo

- Uso de instanceof
  - Exemplo
    - Para a hierarquia Pessoa, Funcionário e Chefe de Departamento
    - Classe Empréstimo Bancário
      - A classe EmprestimoBancario calcula o valor de emprétimos que podem ser dados a pessoas (instâncias das classes Pessoa, Funcionario e ChefeDeDepartamento) por um banco popular. Empréstimos a pessoas que não são funcionárias são de valor fixo, enquanto empréstimos dados a funcionários e chefes de departamento são baseados nos salários que estes recebem.

      *,e sobrecarga, implementando o mesmo método mais especializado para instâncias de classes herdeiras da classe Funcionario.

```
class EmprestimoBancario
 {
 public static void main(String[] argumentos)
  {
  // Criamos uma instÃ¢ncia da classe Pessoa
  Pessoa p1 = new Pessoa("Kurt GÃ¶del",10973213,
                    new Data((byte)23,(byte)12,(short)1904));
  // Criamos trÃªs instÃ¢ncias da classe Funcionario
  Funcionario f1 = new Funcionario("Henri PoincarÃ©",19283712,
                           new Data((byte)12,(byte)7,(short)1897),
                           new Data((byte)28,(byte)1,(short)1918),
                           2500.0f);
  Funcionario f2 = new Funcionario("Paul Dirac",98736812,
                           new Data((byte)20,(byte)1,(short)1885),
                           new Data((byte)31,(byte)3,(short)1909),
                           3200.0f);
  Funcionario f3 = new Funcionario("Wolfgang Pauli",33886620,
                           new Data((byte)14,(byte)9,(short)1902),
                           new Data((byte)16,(byte)11,(short)1930),
                           3600.0f);
  // Criamos uma instancia da classe ChefeDeDepartamento
  ChefeDeDepartamento c1 = new ChefeDeDepartamento("Edwin Hubble",4259782,
                                 new Data((byte)20,(byte)1,(short)1875),
                                 new Data((byte)20,(byte)7,(short)1899),
                                 4100.0f,
                                 "Laboratorio de Astrofisica",
                                 new Data((byte)20,(byte)7,(short)1899));
  // Calculamos os emprÃ©stimos que podem ser feitos a cada pessoa
  System.out.println(calculaEmprestimo(p1));
  System.out.println(calculaEmprestimo(f1));
  System.out.println(calculaEmprestimo(f2));
```

```
 public static float calculaEmprestimo(Pessoa p)
    {
    return 1000.0f; // qualquer pessoa pode ter um emprestimo de 1000 reais.
    } // fim do metodo calculaEmprestimo

public static float calculaEmprestimo(Funcionario f)
    {
    float emprestimo = 0f; // inicialmente consideramos o emprestimo zero
    // Primeiro verificamos se a instancia  é uma instancia da classe
    // ChefeDeDepartamento. Se for, calculamos o emprestimo como sendo quatro
   //vezes o  salario recebido.

    if (f instanceof ChefeDeDepartamento)
      {
      emprestimo = 4.0f*f.qualSalario();
      }
    // Se a instancia nao for da classe ChefeDeDepartamento, verificaremos se ela
// é instancia da classe Funcionario e, se for, calculamos o emprestimo como //sendo duas
vezes o salario recebido.
    else if (f instanceof Funcionario)
      {
      emprestimo = 2.0f*f.qualSalario();
      }
    return emprestimo;
    } // fim do metodo calculaEmprestimo

 } // fim da classe EmprestimoBancario
```

Conversão de Tipos entre Classes

- Cast
- Assim como tipos primitivos
  - Instâncias de Classes podem ser convertidas para outras
    - Implícita
    - Explícita

Conversão Implícita

- Quando fazemos um "casting" de dados primitivos da direita para esquerda, eles ocorrem automaticamente:

  byte -> short -> int -> long -> float -> double

- Isso ocorre, pq quem está na direita é "mais genérico" do que quem está na esquerda.
  - Ou seja, vc faz um "upcasting". O contrário seria um *downcasting*.

    Exemplo:

    ```
    int i = 5;
    long j = i;     //Temos um upcasting ou conversao implícita.
    byte b1 = i; //Errado. “Type Mismatch”.
    byte b2 = (byte) i ; //Correto. Downcasting ou conversão
                                                explicito é necessário.
    ```

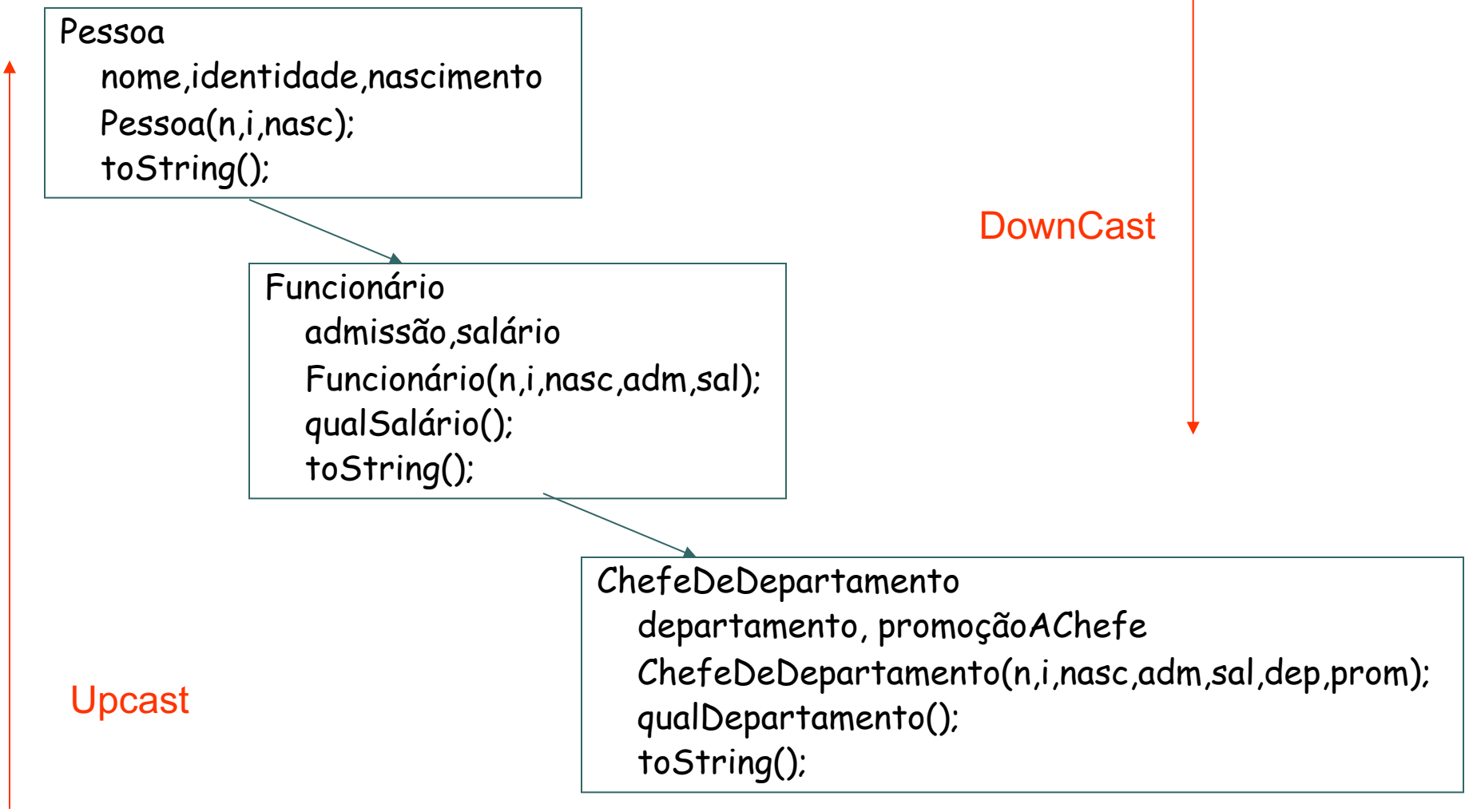
Conversão Implícita

- Upcasting
- A conversão deve ser da classe mais específica para a mais genérica.
- As classes devem estar em uma hierarquia
  - Descendente -> Ancestral
  - Subclasse -> SuperClasse
    - Parent p = new Child();
    - p=c
      - P referencia da classe pai, c referencia da classe filha
  - Já vimos isto no slide 41
  - Blue J, exercício Empresa, Funcionário, Gerente.

Conversão de Tipos entre Classes

- Conversão explicita das instâncias de classes
  - DownCasting
    - (nome da classe filha) variável referência da classe pai
      - Exemplo
        - ChefeDepartamento aux = (ChefeDepartamento)p
        - P é um variável para instâncias da Classe Pessoa.

Upcast X DownCast



Exemplo

- Se na Classe EmprestimoBancario (slide 51) quiséssemos fazer um ‘so método para calcular o valor do empréstimo para Pessoa, Funcionário e Chefe de Departamento?

```
public static float calculaEmprestimo(Pessoa p)
   {
   float emprestimo = 1000.f;
   if (p instanceof ChefeDeDepartamento)
     {

     ChefeDeDepartamento tempora¡rio = (ChefeDeDepartamento)p;
     emprestimo = 4.0f*temporario.qualSalario();
     }
  else if (p instanceof Funcionario)
     {

     Funcionario temporario = (Funcionario)p;
     emprestimo = 2.0f*temporario.qualSalario();
     }
   return emprestimo;
   } // fim do metodo calculaEmprestimo

 } // fim da classe EmprestimoBancarioComCast
```

Não podemos acessar o método qualSalário da instância p pois esta é uma instância da classe Pessoa. Devemos criar uma instância temporária da classe ChefeDeDepartamento a partir de p e usa-la para chamar o método qualSalário. Notem que não é necessário inicializar a instância com a palavra-chave new, ela será somente outra referência a p.