



# Herança – Parte II

DC – UFRPE

Programação II

Prof. Gustavo Callou

[gustavo.callou@ufrpe.br](mailto:gustavo.callou@ufrpe.br)

# Roteiro

- ▶ Conceitos de herança em Java
- ▶ Modificador `protected`
- ▶ Classe `Object`



# Classe Object

- ▶ Em Java toda classe herda de Object
- ▶ Exemplo:

```
public class Conta {...}
```

=

```
public class Conta extends Object {...}
```

# Classe Object

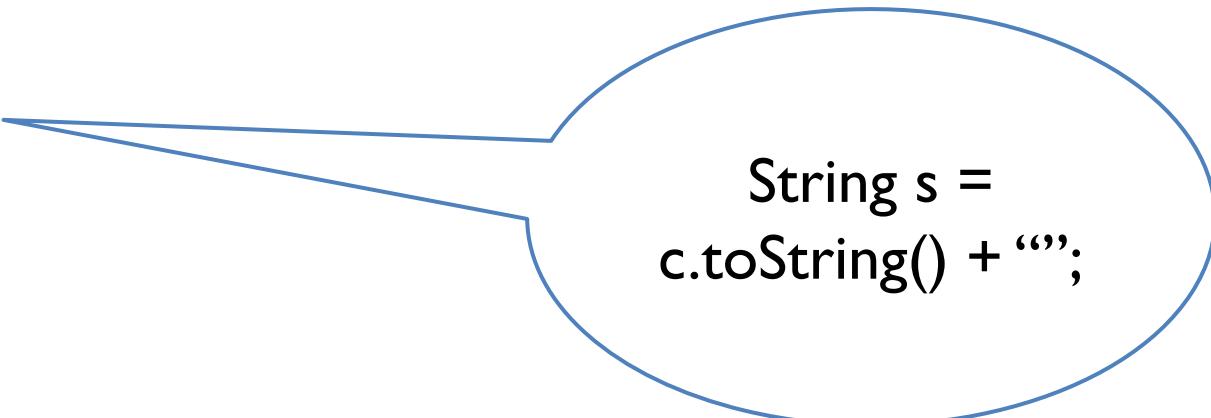
- ▶ Alguns métodos são:
  - ▶ `Class getClass()` : retorna o tipo do objeto
  - ▶ `String toString ()` : retorna a representação textual para o objeto
  - ▶ `boolean equals(Object o)` : compara se o objeto o é igual a this
  - ▶ `void finalize()` : último código executado antes do GC limpar o objeto
  - ▶ ...

# Método `toString()`

- ▶ Chamado implicitamente quando o objeto de uma classe é concatenado com uma string
  - ▶ Também pode ser chamado explicitamente
- ▶ Exemplo:

```
Conta c = new Conta("I", 0);
```

```
String s = c + "";
```



```
String s =  
c.toString() + "";
```

# Sobrescrita (ou redefinição) de método

- ▶ Quando um método da superclasse é redefinido na subclasse
  - ▶ Corpo do método em super diferente do corpo do método em this
- ▶ Assinatura do método na subclasse deve ser idêntica à assinatura da superclasse
- ▶ Override é o termo em inglês para sobrescrita

# Sobrescrevendo Object.toString()

```
public class Conta {  
    // atributos  
    // construtor  
    // outros métodos  
    @Override  
    public String toString() {  
        return this.numero + "," + this.saldo;  
    }  
}
```

Uso opcional.  
Ajuda a prevenir  
erros.

Concatenar  
uma conta com  
uma string vai  
chamar este  
método

# Modificador protected (protegido)

- ▶ Permite que subclasses (e classes do mesmo pacote) tenham acesso

# Classe EmpregadoAssalariado

```
public class EmpregadoAssalariado {  
    protected String nome;  
    protected String cpf;  
    protected double salario;  
    public EmpregadoAssalariado(String nome, String cpf, double salario) {  
        this.nome = nome;  
        this.cpf = cpf;  
        this.salario = salario;  
    }  
    //métodos pagamento, toString e métodos get/set  
}
```

# Classe EmpregadoAssalariado

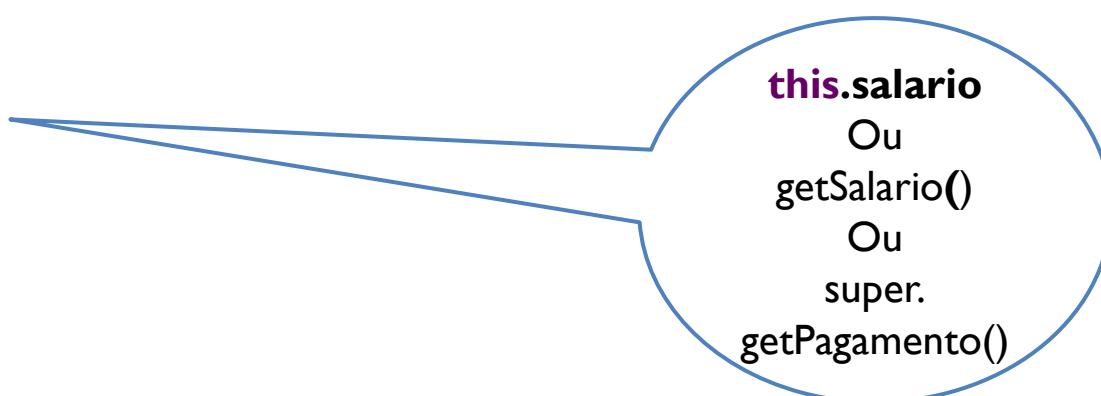
```
public class EmpregadoAssalariado {  
    //atributos e construtor  
    public double getPagamento() {  
        return salario;  
    }  
  
    public String toString() {  
        return "Nome: " + this.nome + "\nCpf: " + this.cpf +  
            "\nSalario R$ " + this.salario;  
    }  
    //métodos get/set  
}
```

# Classe EmpregadoComissionado

```
public class EmpregadoComissionado extends EmpregadoAssalariado {  
    protected double valorVendas  
    protected double comissao;  
    public EmpregadoComissionado(String nome, String cpf,  
        double salario, double valorVendas, double comissao) {  
        super(nome, cpf, salario);  
        this.valorVendas = valorVendas;  
        this.comissao = comissao;  
    }  
    // métodos get/set para valorVendas e comissao  
    // métodos getPagamento e toString  
}
```

# Classe EmpregadoComissionado

```
public class EmpregadoComissionado extends EmpregadoAssalariado {  
    // atributos, construtor, gets/sets  
    public double valorComissao() {  
        return valorVendas * comissao/100;  
    }  
    @Override  
    public double getPagamento() {  
        return super.salario + valorComissao();  
    }  
    // método toString()  
}
```



# Classe EmpregadoComissionado

```
public class EmpregadoComissionado extends EmpregadoAssalariado {  
    // atributos, construtor, gets/sets  
    // métodos valorComissao e getPagamento  
    @Override  
    public String toString() {  
        return super.toString() + "\nComissao % " +  
            comissao + "\nVendas R$ " + valorVendas;  
    }  
}
```

# Resumo dos modificadores

Modificador	Classe	Pacote	Subclasse	Outras
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
Sem modificador	✓	✓	✗	✗
private	✓	✗	✗	✗

# Exercícios

- ▶ 01 – Implementar as seguintes classes:
  - ▶ Classe Conta
  - ▶ Classe Poupanca herda de Conta
    - ▶ Atributo variação
    - ▶ Método renderJuros
  - ▶ Classe ContaEspecial herda de Conta
    - ▶ Atributo limite
    - ▶ Sobrescrever o método debitar