

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

**Programação Orientada
a Objetos**
Herança – Parte 1

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Exemplo 1

Um banco possui dois tipos de contas: conta comum e conta especial. A diferença entre elas é que a conta especial permite que o cliente fique com o saldo negativo até um limite especificado pelo banco

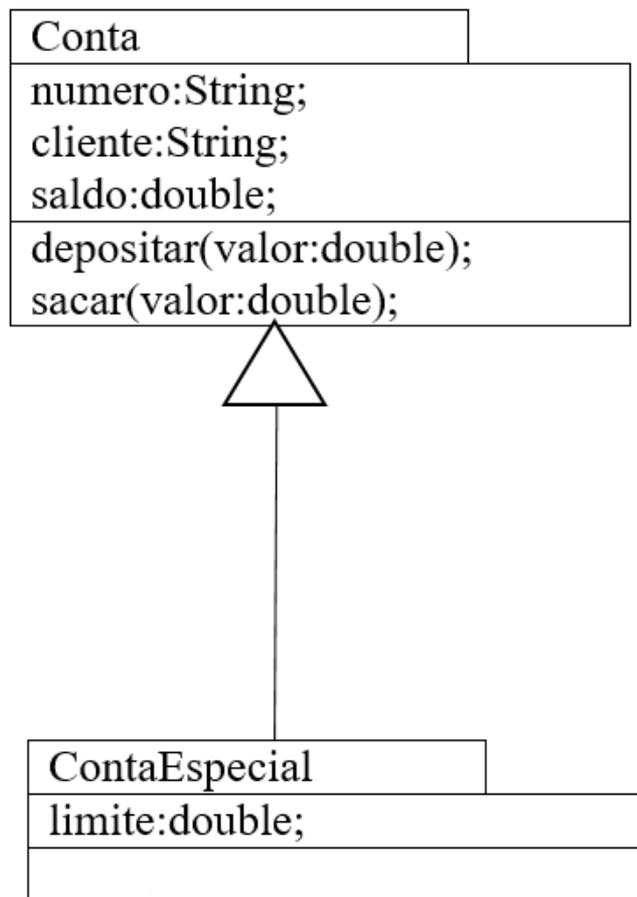
Exemplo 1

Um banco possui dois tipos de contas: conta comum e conta especial. A diferença entre elas é que a conta especial permite que o cliente fique com o saldo negativo até um limite especificado pelo banco

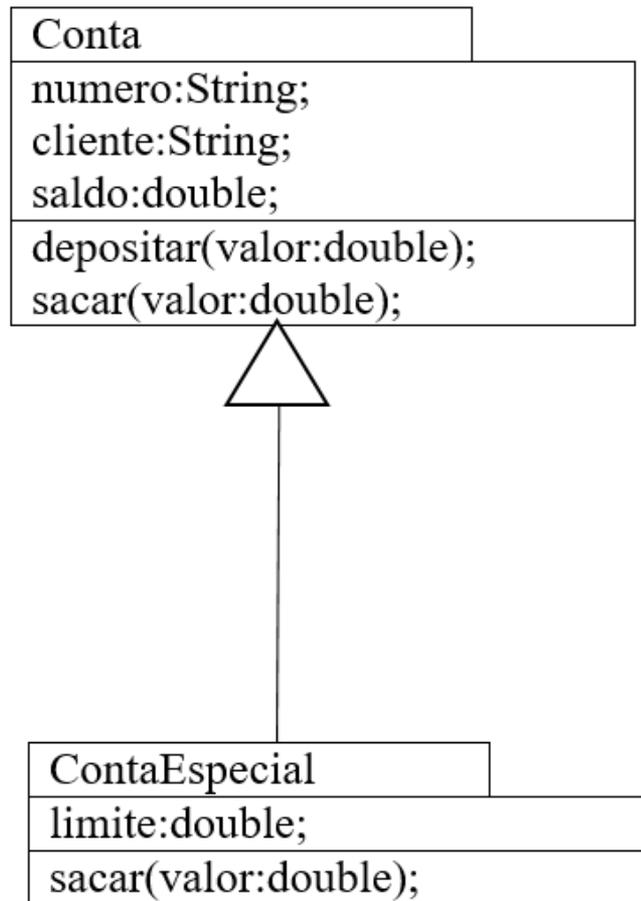
Conta
numero:String; cliente:String; saldo:double;
depositar(valor:double); sacar(valor:double);

ContaEspecial
numero:String; cliente:String; saldo:double; limite:double;
depositar(valor:double); sacar(valor:double);

Exemplo 1 – Herança



Exemplo 1 – Herança



Exemplo 2

Uma universidade possui dois tipos de funcionários: Técnico e Docente. Esses funcionários se diferenciam em alguns dados que precisam ser armazenados e também no cálculo do salário

- Técnico: salário + hora extra
- Docente: salário + adicional por nível

Exemplo 2

Uma universidade possui dois tipos de funcionários: Técnico e Docente. Esses funcionários se diferenciam em alguns dados que precisam ser armazenados e também no cálculo do salário

- Técnico: salário + hora extra
- Docente: salário + adicional por nível

Tecnico
nome:String; funcao:String; salario:double; horaextra:double;
calcularSalario():double;

Docente
nome:String; titulacao:String; salario:double; nivel:integer;
calcularSalario():double;

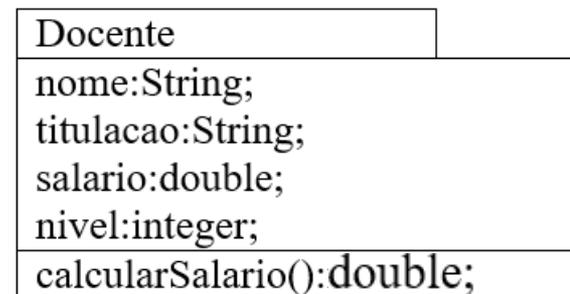
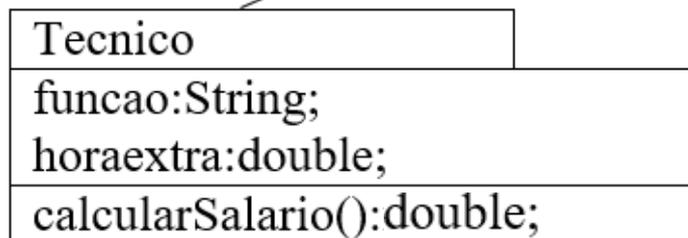
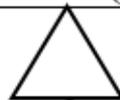
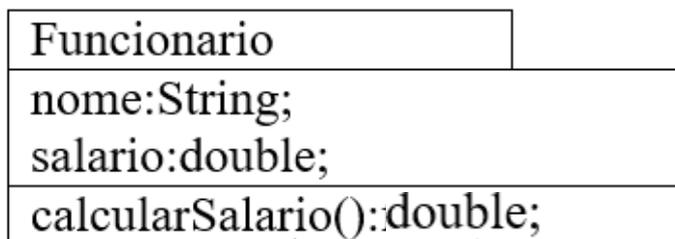
Exemplo 2 - Herança

Funcionario
nome:String; salario:double;
calcularSalario():double;

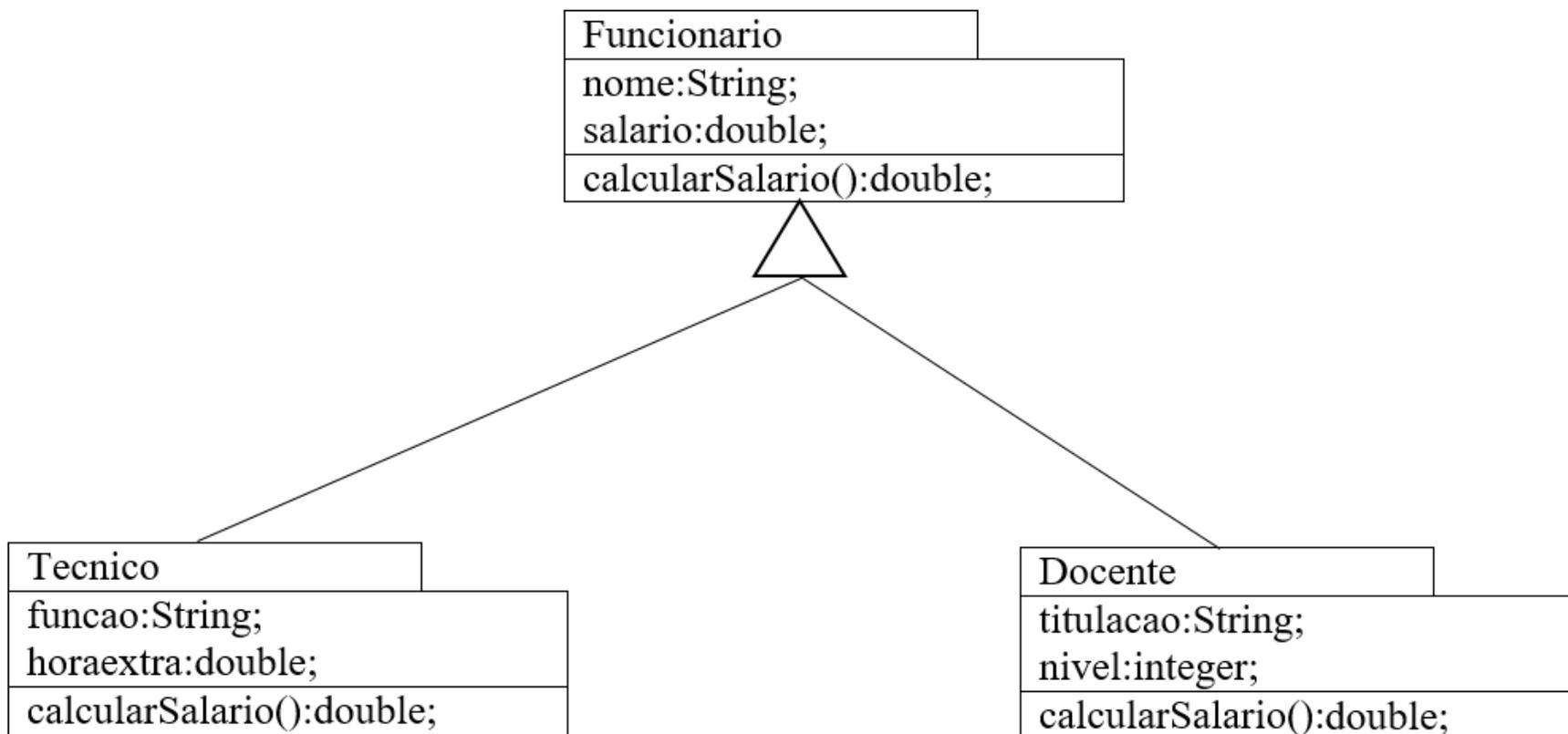
Tecnico
nome:String; funcao:String; salario:double; horaextra:double;
calcularSalario():double;

Docente
nome:String; titulacao:String; salario:double; nivel:integer;
calcularSalario():double;

Exemplo 2 - Herança



Exemplo 2 - Herança



Herança

- Nos exemplos, podemos constatar que alguns atributos e métodos podem ser comuns às diferentes classes
 - Dados pessoais
 - Calculo de salário
- Com o que vimos até agora, esses atributos e métodos estariam duplicados nas diferentes classes
 - A correção de um erro ou uma modificação de código levaria o programador a modificar os métodos de todas as classes

Herança

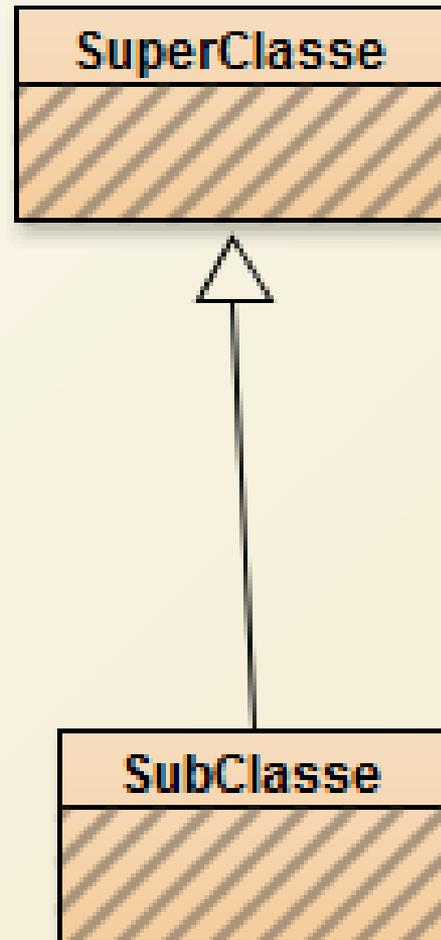
- Em POO há um outro tipo de relacionamento entre objetos que é conhecido como **Herança**
- Ela é uma forma de **reutilização de software** na qual uma nova classe é criada
- Ela permite que membros (atributos e métodos) de uma classe existente sejam aprimorados com capacidades novas ou modificadas

Herança

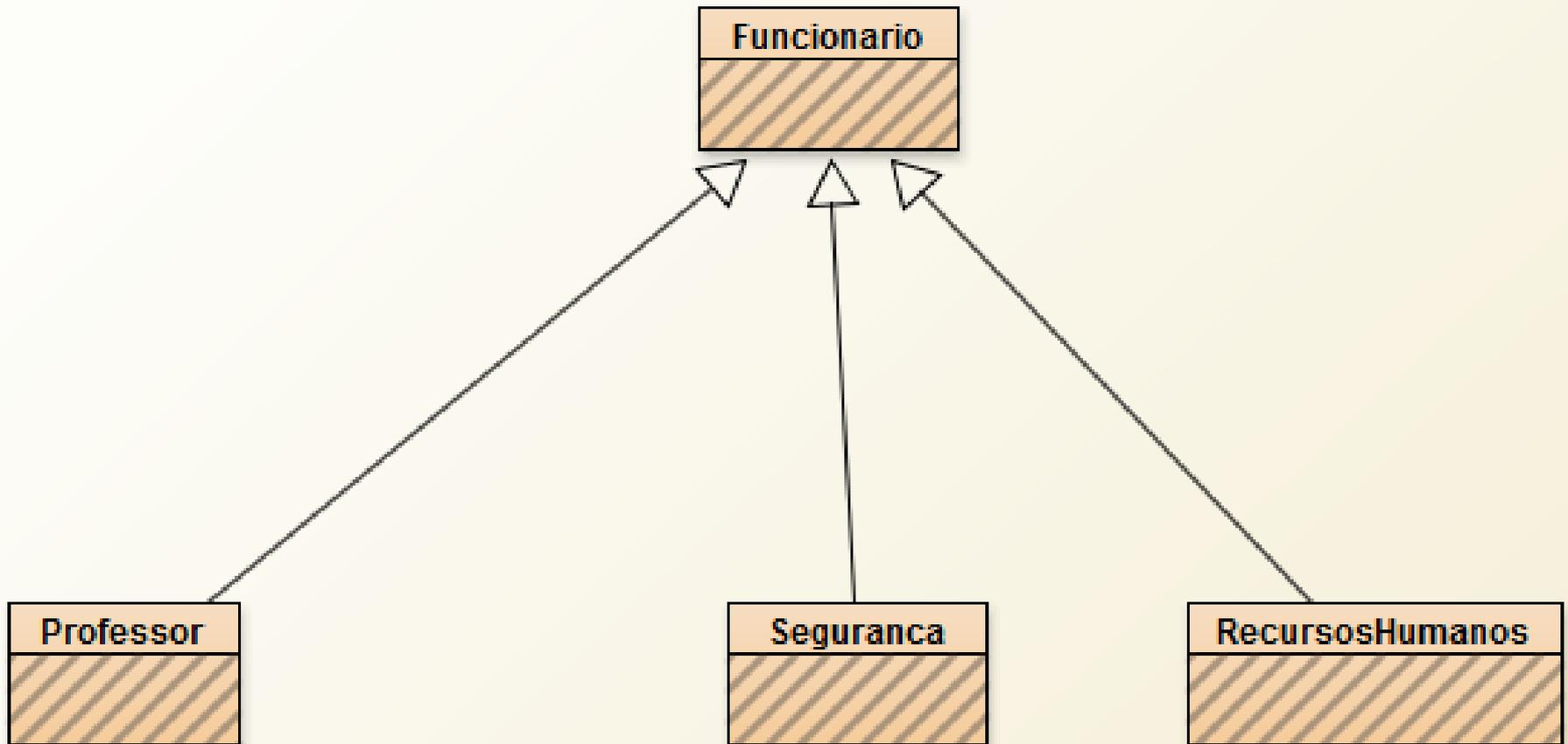
- Ao criar uma classe, em vez de declarar membros completamente novos, o programador pode designar que a nova classe deverá **herdar membros de uma classe existente**
 - Em outras palavras, ela **estende** uma classe existente
- A classe mãe é chamada de **superclasse** e a classe filha de **subclasse**
- Também podemos chamá-las de **classe base** e **classe derivada**
- Cada subclasse pode tornar-se a superclasse para futuras subclasses

Herança

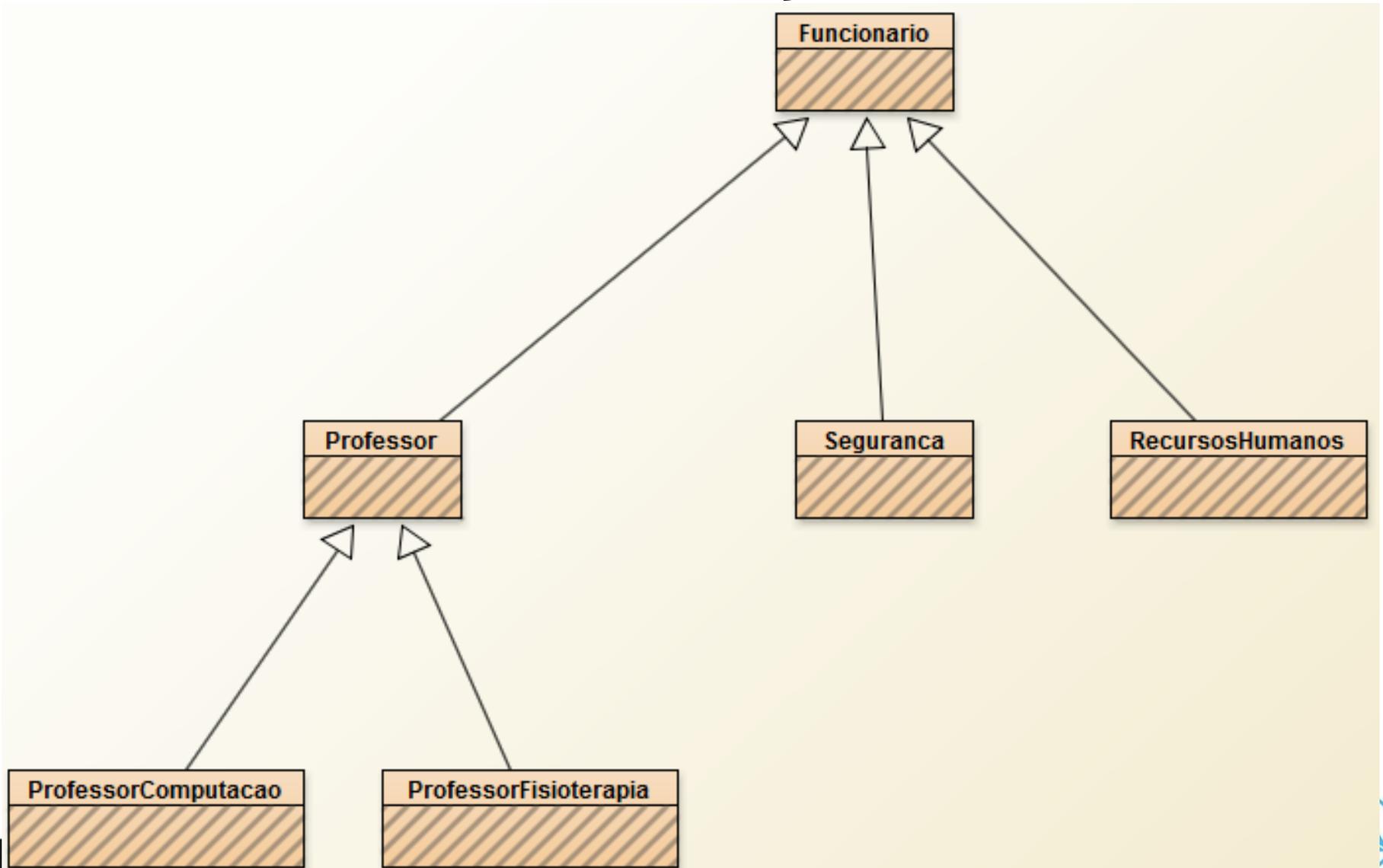
- Representação Gráfica



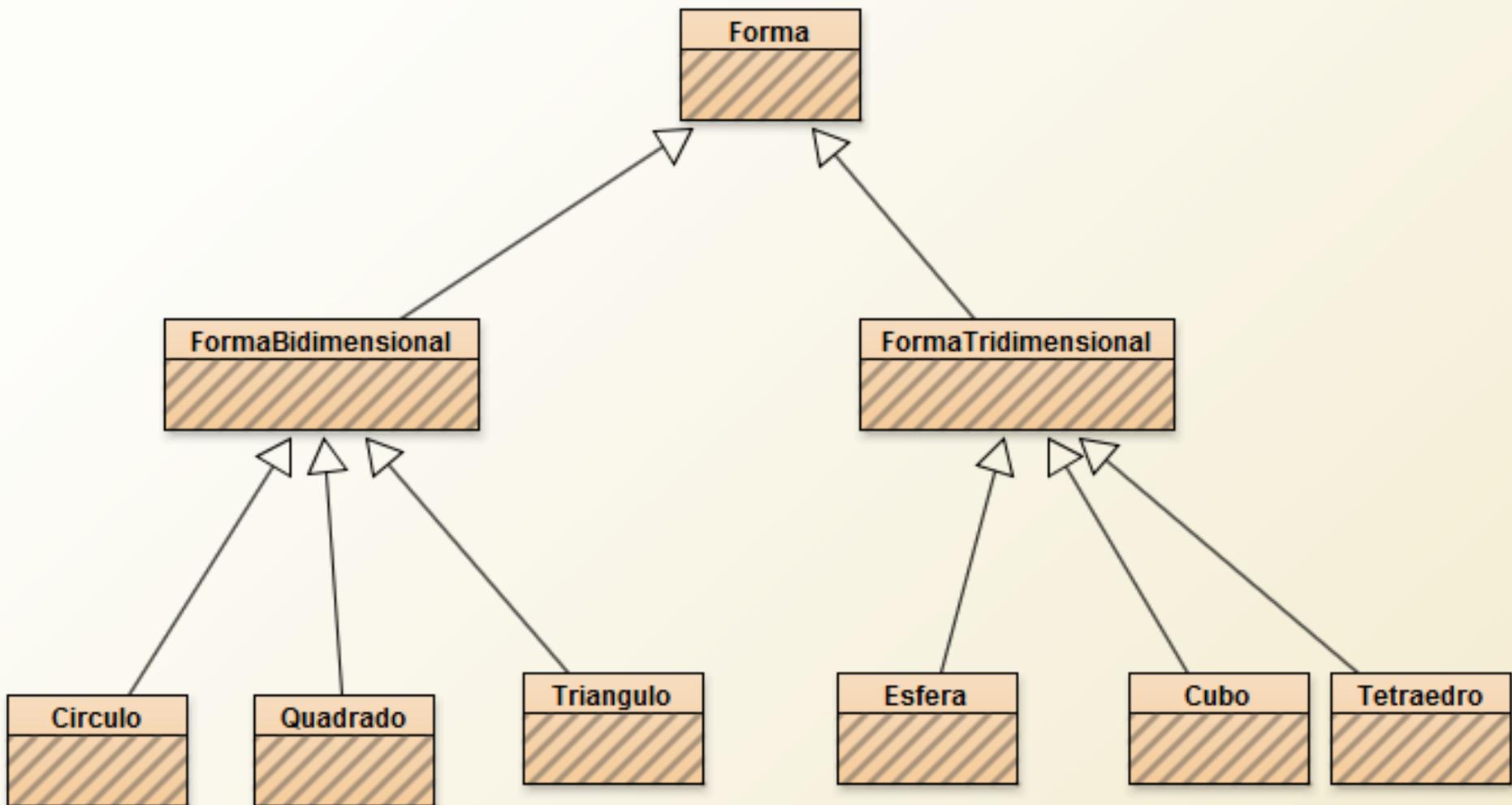
Herança



Herança



Herança: outro exemplo



Herança: sintaxe java

- Em java a palavra **extends** é utilizada para indicar que uma classe irá estender outra

```
class SuperClasse{
```

```
.....
```

```
}
```

```
class SubClasse extends SuperClasse{
```

```
....
```

```
}
```

Herança: sintaxe java

```
class Funcionario{
```

```
.....
```

```
}
```

```
class Professor extends Funcionario{
```

```
.....
```

```
}
```

```
class Seguranca extends Funcionario{
```

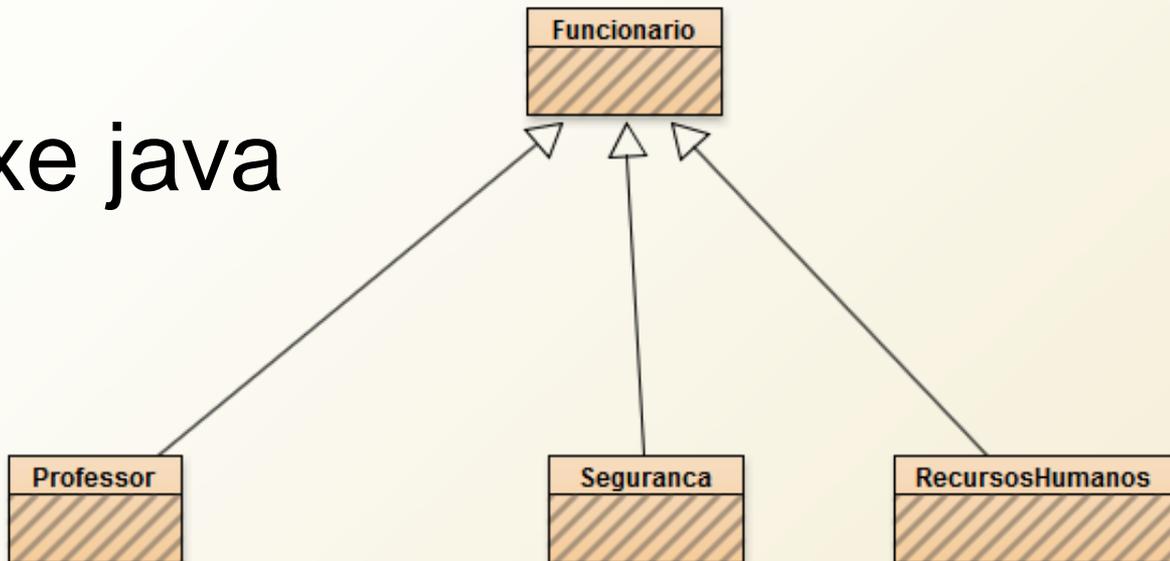
```
.....
```

```
}
```

```
class RecursosHumanos extends Funcionario{
```

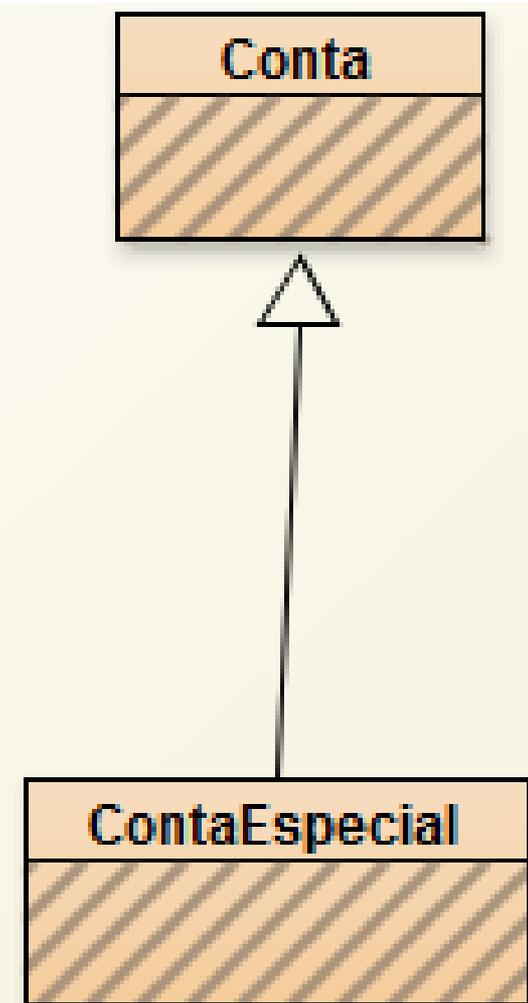
```
.....
```

```
}
```



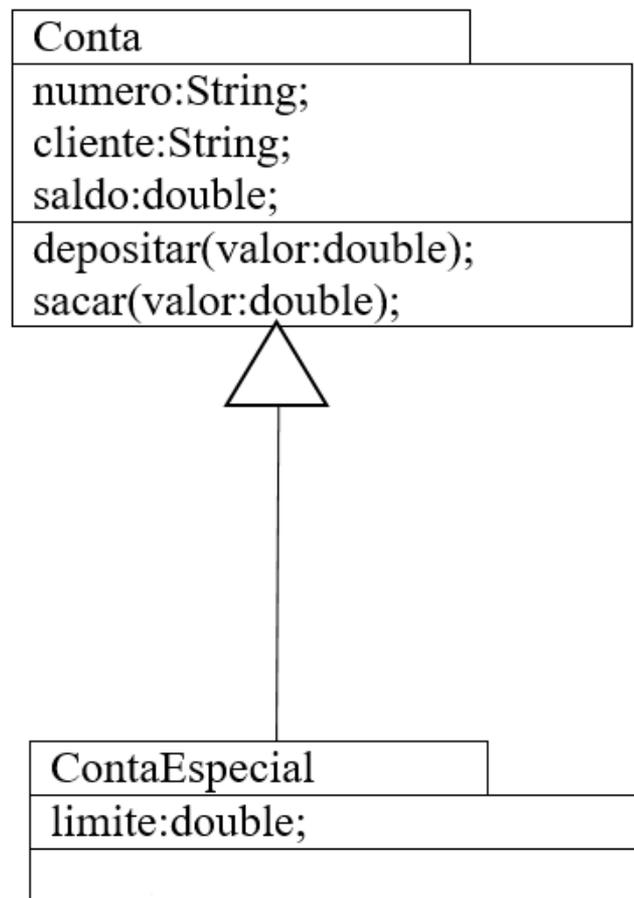
Exemplo

- Conta corrente normal
 - Permite o saque se o saldo não ficar negativo
- Conta corrente especial
 - Possui limite negativo para saque



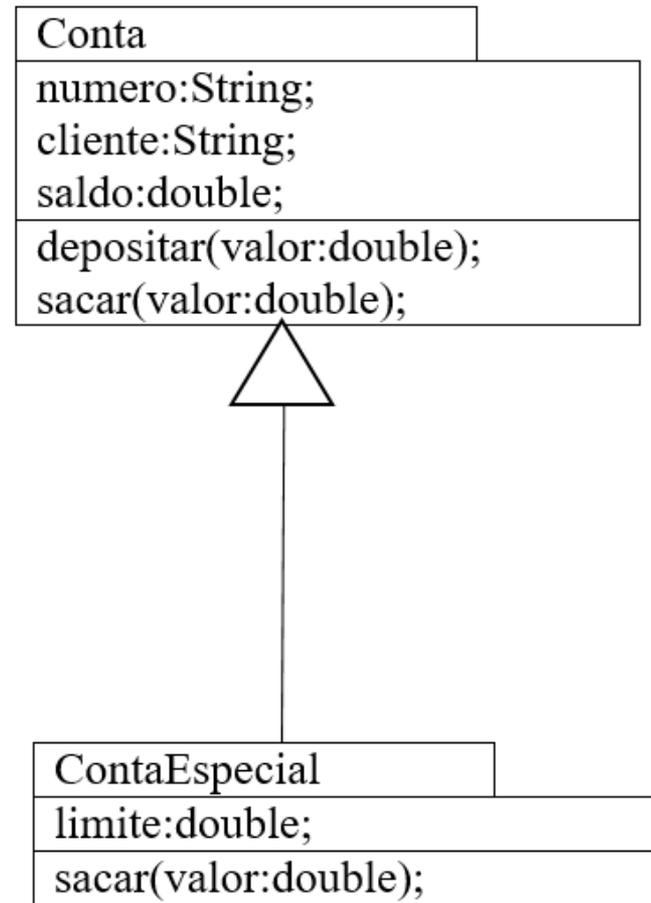
Exemplo

- Conta corrente normal
 - Permite o saque se o saldo não ficar negativo
- Conta corrente especial
 - Possui limite negativo para saque



Exemplo

- Conta corrente normal
 - Permite o saque se o saldo não ficar negativo
- Conta corrente especial
 - Possui limite negativo para saque



Exemplo Prático

Referências

BIBLIOGRAFIA BÁSICA

1. SINTES, A., Aprenda programação orientada a objetos em 21 dias, Pearson Education do Brasil, 2002.
2. VAREJÃO, F., Linguagens de programação : Java, C e C++ e outras : conceitos e técnicas, Campus, 2004.
3. DEITEL, H. M., DEITEL, P. J., **Java**: como programar, São Paulo: Pearson Education do Brasil, 2010. 1144p.
4. DEITEL, H. M., DEITEL, P. J., **Java**: como programar, Porto Alegre: Bookman, 2003. 1386p.
5. SAVITCH, W. J., C++ absoluto, Pearson Education : Addison Wesley, 2004.

BIBLIOGRAFIA COMPLEMENTAR

1. BERMAN, A. M. *Data Structures via C++: Objects by Evolution*, Oxford University Press Inc., 1997.
2. BARNES, D.J. & KÖLLING, M., Programação orientada a objetos com Java, Pearson Education : Prentice Hall, 2004.
3. DEITEL, H. M. e DEITEL, P. J. *C++: Como Programar*, Bookman, 2001.
4. GILBERT, R. F. e FOROUZAN, B. A. *Data Structures: A Pseudo Approach with C++*, Brooks/Cole Thomson Learning, 2001.
5. MUSSER, D. R. e SAINI, A. *STL Tutorial and Reference Guide: Programming with the Standard Template Library*, Addison-Wesley, 1996.
6. SEBESTA, R. W. *Conceitos de Linguagem de Programação*, 4ª Ed., Bookman, 2003.
7. SEDGEWICK, R. *Algorithms in C++*, Addison-Wesley, 2002.
8. STROUSTRUP, B. *A Linguagem de Programação C++*, 3ª Ed., Bookman, 2000.

Links

- <http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>
- http://www.di.ubi.pt/~pprata/poo_10_11.htm
- <https://sites.google.com/site/fpaulovich/Home/Teaching/scc-604-programacao-orientada-a-objetos>

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

**Programação Orientada
a Objetos**
Herança – Parte 1

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br