

Programação Orientada a Objetos

Prof. Walter Gima



Herança

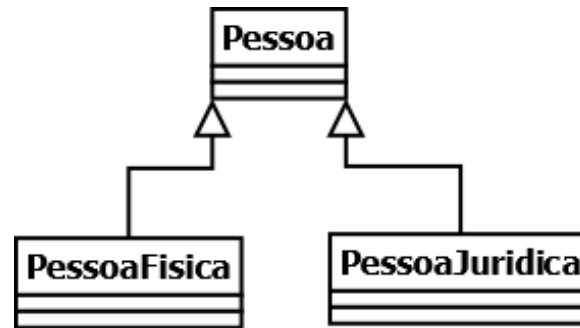
- É um mecanismo de reutilização de código, a partir do qual uma nova classe é criada, absorvendo membros de uma classe existente e aprimorada com novas potencialidades.
- Por meio de herança, os desenvolvedores economizam tempo durante o desenvolvimento de sistemas.

leia mais em: Cap. 8 Reutilização de classes do livro SANTOS, Rafael.

Introdução à Programação Orientada a Objetos Usando Java. 2ª ed. Rio de Janeiro: Campus - Elsevier, 2013. (PLT 78)

Cap.9 Herança do livro Serson, Roberto Rubinstein. Programação Orientada a Objetos com Java 6. Brasport, 2007.

Herança



- A classe pessoa é chamada de superclasse ou classe pai.
- As classes PessoaFisica e PessoaJuridica são chamadas de subclasses.
- As subclasses podem se tornar superclasses para futuras subclasses.
- A palavra reservada **extends** permite utilizar herança entre as classes.

Herança

```
public class ContaPoupanca extends Conta{  
  
    private String aniversario;  
  
    public ContaPoupanca() {  
    }  
  
    public ContaPoupanca(int agencia, int numeroConta, int codigoCliente, String aniversario) {  
        super(agencia, numeroConta, codigoCliente);  
        this.aniversario = aniversario;  
    }  
  
    public String getAniversario() {  
        return aniversario;  
    }  
  
    public void setAniversario(String aniversario) {  
        this.aniversario = aniversario;  
    }  
}
```

leia mais em: Cap. 8 Reutilização de classes do livro SANTOS, Rafael. Introdução à Programação Orientada a Objetos Usando Java. 2ª ed. Rio de Janeiro: Campus - Elsevier, 2013. (PLT 78)
Cap.9 Herança do livro Serson, Roberto Rubinstein. Programação Orientada a Objetos com Java 6. Brasport, 2007.

Herança

- Sobreposição de métodos: (Override)
 - Se uma classe pai possui um método toString() e na classe filha for criado outro método toString(), esse método irá sobrepor a método da classe pai, ou seja, quando a classe filha ser instanciada e ao ser chamado o método toString(), vai ser executado o método da classe filha.
- Em resumo, Os métodos criados na classe filha com o mesmo nome tomam o lugar (sobrepoem) dos métodos da classe pai.

Herança

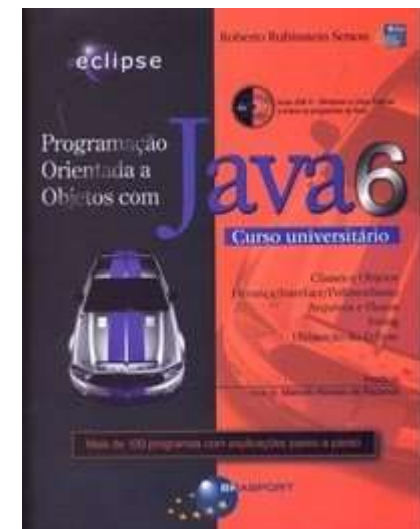
- Visibilidade:
 - Métodos e atributos declarados na classe pai como `private` não podem ser acessados na classe filha.
 - Métodos e atributos declarados na classe pai como `protected` são visíveis para as classes filhas.

Herança

- Palavra reservada **super**:
 - Para uma classe filha acessar o construtor da classe pai deve utilizar a palavra reservada `super`:
 - Ex: `super(nome);`
 - Quando necessário acessar um método da superclasse também se utiliza a palavra `super` + o nome do método.
 - Ex: `super.toString();`

Livro Texto:

- » 1) SANTOS, Rafael. Introdução à Programação Orientada a Objetos Usando Java. 2ª ed. Rio de Janeiro: Campus - Elsevier, 2013.
- » 2) DEITEL, Harvery M.. Java: Como Programar. 6ª ed. São Paulo: Pearson - Prentice Hall, 2007.
- » 3) SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java. 1ª ed. Rio de Janeiro: Alta Books, 2005.
- » 4) Serson, Roberto Rubinstein. Programação Orientada a Objetos com Java 6. Brasport, 2007.





Anhanguera

Dúvidas ?

walter.gima@anhanguera.com