

UNIP
UNIVERSIDADE PAULISTA

Linguagem de Programação Orientada a Objeto

LPOO 07- Classe Abstrata


Prof. Msc Wanderley Gonçalves Freitas

Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 1 de 24

UNIP
UNIVERSIDADE PAULISTA

Agenda

- Conceito de classe Abstrata
- Métodos abstratos
- Extensão com abstrata em java
- Diferenciar entre Classe Abstrata x Classe Concreta
- Exercícios de fixação
 - Classe abstrata



Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 2 de 24

UNIP
UNIVERSIDADE PAULISTA

Objetivo

- Capacitar o aluno a compreender os conceitos de orientação a objetos e sua utilização no desenvolvimento de software de qualidade numa linguagem de programação moderna (JAVA)
- Distinguir entre classes concretas, classe abstratas e interface.
- Diferenciar métodos abstratos e métodos concretos
- Como declarar métodos abstratos para criar classes abstratas.
- Como declarar e implementar interfaces

Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 3 de 24

UNIP
UNIVERSIDADE PAULISTA

Classe Abstrata

- Classe abstrata é uma classe que define o comportamento e atributos para uma **subclasses concretas**
- Uma **classe abstrata** não pode ser instanciada
- Uma classe abstrata pode possuir **métodos concretos e métodos abstratos**.
- A implementação de um método abstrato é feita na **subclasse concreta por intermédio de sobrescrever o método (overriding)**
- São definidas através da palavra-chave **abstract**:

Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 4 de 24

UNIP
UNIVERSIDADE PAULISTA

Métodos abstratos

- Um método abstrato é o método contido em uma **classe abstrata** que **não possui implementação**
- Os métodos abstratos devem ser implementados pelas subclasses ou classe também serão **classes abstratas**
- Se uma classe possuir pelo **menos um método abstrato**, ela deve ser **classe abstrata**

```

3 public abstract class Funcionario {
4
5     protected double salario;
6
7     public abstract double calcularBonificacao();
8     public abstract void imprimir();
9
10    public double calcularSalario(){
11        double vCalculo = (salario * 0.
12        return vCalculo;
13    }
14
  
```

Metodos Abstratos não possui implementação

Metodos Concretos possui implementação

Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 5 de 24

UNIP
UNIVERSIDADE PAULISTA

Diferenciar : Métodos concretos e Métodos Abstratos

Métodos abstratos X Métodos concretos

- **Métodos abstratos** são métodos sem corpo(sem implementação), nesse caso vc tem apenas a assinatura do método

```

public abstract double calcularBonificacao();
public abstract void imprimir();
  
```

- **Métodos Concretos** são métodos com corpo (implementação) Existe algum código ou não entre parêntese de abertura{ e parêntese fechante}, também chamado de bloco.

```

public double calcularSalario(){
    double vCalculo = (salario * 0.20 ) + salario;
    return vCalculo;
}
public void buscarNome(String pNome){
  
```

Prof. Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 6 de 24

Diferenciar : classe abstrata , classe concreta

Classes Abstratas X Classes Concretas

- Uma *classe abstrata* é uma classe que **não pode ser instanciada**, mas cujas classes descendentes (subclasses concretas) podem ter instâncias diretas. **A classe abstrata pode possuir métodos concretos e métodos abstratos**, basta ter apenas um método abstrato para ser considerada como classe abstrata

```

3 public abstract class Funcionario {
4
5     protected double salario;
6
7     public abstract double calcularBonificacao();
8     public abstract void imprimir();
    
```

- Uma *classe concreta* é uma **classe que pode ser instanciada**.

```

3 public class Professor extends Funcionario {
4
5     public double getPrimeiraParcela() {
6         return (salario * 0.40);
7     }
    
```

Prof: Msc Wanderley Gonçalves Freitas 7 de 24

Classe Abstrata - Java

Sintaxe:

```

abstract class NomeT
// corpo da class
...
}
    
```

```

3 public abstract class Funcionario {
4
5     protected double salario;
6
7     public abstract double calcularBonificacao();
8     public abstract void imprimir();
9
10    public double calcularSalario() {
11        double vCalculo = (salario * 0.20 ) + salario;
12        return vCalculo;
13    }
14
    
```

Metodos Abstratos não possui implementação

Metodos Concretos possui implementação

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 8 de 24

Implementação dos Metodos Abstrato

```

3 public class Professor extends Funcionario {
4
5
6     public double calcularBonificacao() {
7         double vCalculo = (salario * 0.70);
8         return vCalculo;
9     }
10
11    public void imprimir() {
12        System.out.println("salario: " + salario);
13    }
14 }
    
```

Metodos sobrescritos da superclasse

Metodos Concretos possui implementação

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 9 de 24

Classe Abstrata - Exercício

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 10 de 24

Exercicio (1).....corrigir

- Na UML, uma classe abstrata e método abstrato são representada com o seu **nome em itálico**.

Classe Abstrata

Método Abstrato

Método Concreto

Classe concreta

powered by Astah

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 11 de 24

Exercícios(1)

- 1 – Crie a classe abstrata Pessoa.
 - Pacote : br.unip.lpoo.np2.si2p30.
 - Atributos (protected) : nome(String), rendimento(double), imposto(double).
 - Cria os seguintes métodos concretos conforme a descrição abaixo?

Tipo Retorno	Nome método	Parâmetro	Objetivo do método - corpo
double.	gerarPagamento	ausente	Dever retorna o resultado da formula: Formula : vCalculo= rendimento - imposto;

- Cria o método abstrato, conforme a descrição abaixo:

Tipo Retorno	Nome método	Parâmetro	Objetivo do método - corpo
void	calcularImposto	ausente	

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 12 de 24

Exercícios(1)

2 - Crie classe PessoaFisica subclasse da classe Pessoa:

- Atributo(protected) : cpf(String).

- O método será implementado, conforme tabela

Nome método	Objetivo do método - corpo
calcularImposto	<ul style="list-style-type: none"> ❖ Regra de Negócio – Para qualquer Pessoa física, paga imposto 20% sobre o valor do rendimento adicionado um valor de R\$ 100,00 <p>Formula : $\text{imposto} = (\text{rendimento} * 0.20) + 100$</p>

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 13 de 24

Exercícios(1)

3 - Crie classe concreta PessoaJuridica subclasse da classe Pessoa:

- Atributo(protected) : cnpj(String).

- O método será implementado conforme tabela

Nome método	Objetivo do método - corpo
calcularImposto	<ul style="list-style-type: none"> ❖ Regra de Negócio – Para qualquer Pessoa jurídica, paga de imposto 40% sobre o valor do rendimento adicionado um valor de R\$ 400 <p>Formula : $\text{imposto} = (\text{rendimento} * 0.40) + 400$</p>

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 14 de 24

Exercícios(2)

1 - Crie outra classe "Teste" com o método main com as seguintes atividades:

- 1) Atividade
 - Instancie um **objeto** da classe Pessoa. Analise com que aconteceu ????
- 2) Atividade
 - Instancie um **objeto** da classe Pessoa Fisica
 - Atribua valores para **atributos** nome = "maria" e rendimento = 10000
 - Invoque o metodo calcularImposto
 - Exibe no console "imposto" + invoque o metodo getImposto
 - Exibe no console "Pagamento" + Invoque o metodo GerarPagamento
- 3) Atividade
 - Instancie um **objeto** da classe Pessoa juridica
 - Atribua valores para **atributos** nome = "padaria" e rendimento = 10000
 - Invoque o metodo calcularImposto
 - Exibe no console "imposto" + invoque o metodo getImposto
 - Exibe no console "Pagamento" + Invoque o metodo GerarPagamento

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 15 de 24

Resposta(1)

1- Crie a classe abstrata Pessoa.

```

3 public abstract class Pessoa {
4     protected String nome;
5     protected double rendimento;
6     protected double imposto;
7
8     public abstract void calcularImposto();
9
10    public double GerarPagamento() {
11        double vCalculo= rendimento - imposto;
12        return vCalculo;
13    }
14

```

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 16 de 24

Resposta(1)

2 - Crie classe PessoaFisica subclasse da classe Pessoa:

```

3 public class PessoaFisica extends Pessoa{
4
5     private String cpf;
6
7     public void calcularImposto() {
8         imposto = (rendimento * 0.20) + 100;
9     }

```

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 17 de 24

Resposta(1)

3 - Crie classe concreta PessoaJuridica subclasse da classe Pessoa:

```

3 public class PessoaJuridica extends Pessoa{
4
5     private String cnpj;
6
7     public void calcularImposto() {
8         imposto=(rendimento * 0.40) + 400 ;
9     }

```

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 18 de 24

Resposta(2)

1 - Atividade

```
public static void main(String[] args) {
    Pessoa pessoa = new Pessoa();
}
```

- Faz comentário da linha para tirar o erro.

Cannot instantiate the type Pessoa

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 19 de 24

Resposta(2)

2 - Atividade

```
public static void main(String[] args) {
    //atividade 2
    PessoaFisica marcos = new PessoaFisica();
    System.out.println("objeto - marcos");
    marcos.setCpf("222.234.422-22");
    marcos.setNome("marcos");
    marcos.setRendimento(1000);
    marcos.calcularImposto();
    System.out.println("Imposto " + marcos.getImposto());
    System.out.println("Pagamento " + marcos.GerarPagamento());
}
```

```
<terminated> TestePessoa [Java Application] C:\Pro
objeto - marcos
Imposto 300.0
Pagamento 700.0
```

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe 20 de 24

Resposta(2)

3 - Atividade

```
public static void main(String[] args) {
    //atividade 3
    System.out.println("");
    PessoaJuridica padaria = new PessoaJuridica();
    System.out.println("objeto - padaria");
    padaria.setCnpj("2312.321312.321312-77");
    padaria.setNome("Bom Pão");
    padaria.setRendimento(1000);
    padaria.calcularImposto();
    System.out.println("Imposto " + padaria.getImposto());
    System.out.println("Pagamento " + padaria.GerarPagamento());
}
```

```
objeto - padaria
Imposto 800.0
Pagamento 200.0
```

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – C 21 de 24

Referências Bibliográficas

BRUCE, Eckel. **Thinking in Java**. 4 ed. Massachusetts: Editora Prentice hall: 2006.

FOWLER, Martin. **UML essencial : um breve guia para a linguagem** – padrão de modelagem de objetos. Tradução Vera Pezerico e Christian Thomas Prices. 3. ed. Porto Alegre : Bookman, 2005.

FURLAN, Jose Davi **Modelagem de Objetos Através da UML** São Paulo: Editora Makron Books, 2000, ISBN 8534609241,

GRADY, Booch; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário** . tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus. 2000.

SIERRA, kathy; BATES, Bert. **Certificação Sun para programadores e desenvolvedor Java 2**. Rio de Janeiro: Editora Altas Books. 2003.

SIERRA, kathy; BATES, Bert. **Head first java 5. 2 ed**. Sebastopol: Editora O' Reilly Media. Books. 2005.

TOM, Pender. **UML, a bíblia**. Rio de Janeiro: Elsevier, 2004.

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 22 de 24

Perguntas

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 23 de 24

Obrigado

Wanderley

Wanderley.unip@gmail.com

www.wg.pro.br

Prof: Msc Wanderley Gonçalves Freitas LPOO 07 – Classe Abstrata 24 de 24