

Programação Orientada a Objeto

14. Herança e Ligação Dinâmica (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

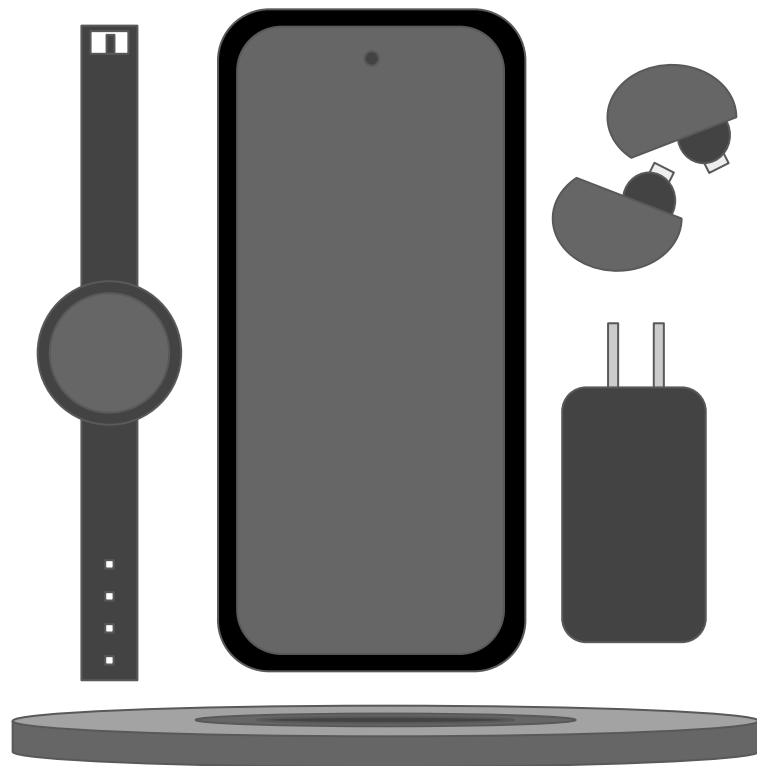
2º Semestre de 2020

Objetivos

- Exercitar os conceitos de POO vistos na aula anterior!
 - Escrevendo, compilando e interpretando (executando) aplicações em Java;
 - Modelando (e/ou interpretando) sistemas baseados em OO;
 - Relações entre classes;

Exercício #1 Parte #0


- Vamos **re**modelar parte do sistema de vendas *online* da AP04, que registra estoque de produtos; vendas (carrinho com produtos com venda concluída); e pagamentos (com parcelamento).
- Em especial, vamos reutilizar nossa implementação anterior* e incluir uma entidade que modela um cliente [...]



Exercício #1 Parte #1

- Refinar o diagrama de classes (na perspectiva conceitual), que contém as classes **Produto**, **Pagamento** e **Venda** e seus relacionamentos, e incluir as classes que modelam cliente (**Cliente**, **ClientePFisica** e **ClientePJuridica**);
 - **Associação**: objetos podem usar outros objetos, principalmente como atributos
 - **Agregação**: objetos podem conter outros objetos como um elemento “parte”. Os elementos “parte” podem existir mesmo sem a existência do elemento “todo”.
 - **Composição**: um objeto “todo” podem ser formados por outros objetos “partes”. Os elementos “parte” só existem enquanto o elemento “todo” existir.
 - **Especialização/Generalização**: relacionamento onde, de um lado da leitura, podemos dizer que um objeto da classe A “é um” objeto da classe B também

Exercício #1 Parte #2

- Um **Cliente** tem um **identificador** (valor inteiro, sequencial, começando em 1, e atribuído automaticamente) e uma **dataCadastro** (**String**), ambos fixos e **obrigatórios**;
 - O construtor da classe **Cliente** recebe e atribui um valor para **dataCadastro**;
 - Um **Cliente** tem um método **imprimeDados** que imprime seus dados. Se **identificador** = 11 e **dataCadastro** = "01/02/2003" então a impressão é "Cliente: #11\nData de cadastro: 01/02/2003";
 - **Cliente** não tem outros métodos (exceto os *getters* e/ou *setters* **necessários**) [...]
- 

Exercício #1 Parte #2

- **Clientes** podem ser pessoas físicas (**ClientePFisica**) ou pessoas jurídicas (**ClientePJuridica**);
- Uma **ClientePFisica** tem, adicionalmente, um **nome** e um **cpf** (ambos da classe **String**, fixos e são **obrigatórios**);
- Uma **ClientePJuridica** tem, adicionalmente, uma **razaoSocial** e um **cpnj** (ambos da classe **String**, fixos e são **obrigatórios**);
- **ClientePFisicas** e **ClientePJuridicas** também tem um método **imprimeDados** que mostram **todos** seus dados (gerais e específicos) - seguindo modelo de impressão anterior.

Exercício #1 Parte #2

- Clientes podem ser pessoas físicas (ClientePFisica) ou pessoas jurídicas (ClientePJuridica);
- Uma ClientePFisica tem, adicionalmente, um nome e um cpf (ambos da classe String e não nulos);
- *Os atributos de Cliente, ClientePFisica e ClientePJuridica devem ser **privados**!*
- ClientePFisicas e ClientePJuridicas também tem um método `imprimeDados` que mostram **todos** seus dados (gerais e específicos) - seguindo modelo de impressão anterior.

Exercício #1 Parte #2

- Altere **Venda** para, ao invés de manter **cpf**, manter e gerenciar uma referência **cliente** para **Cliente**;
- O construtor de **Venda** deve ser atualizado de acordo;
- O método **imprimeRecibo** de **Venda** deve, ao invés de exibir o valor do antigo atributo **cpf**, imprimir os dados do **cliente** da **Venda** (que são privados);

```
" Dados da Venda:  
  <dados-do-cliente>  
  Data: <data-da-venda>  
  ...  
"
```


Exercício #1 Parte #3

- Altere o método *main* de **Aplicacao** da AP04;
- Crie dois clientes (**Cliente**) e os instancie com como pessoas física e jurídica, um de cada;
- Com os **Produtos** já criados, crie duas **Vendas**;
- Registre **produto1** e **produto5**, 1 unidade cada, na venda da pessoa física;
- Registre **produto1** e **produto3**, 200 unidades cada, na venda da pessoa jurídica;
- Faça o pagamento em duas e uma parcelas nas **Vendas** de pessoa física e jurídica, respectivamente.
- Imprima os recibos das duas **Vendas**;

Exercício #1 - Exemplo de Execução

Dados da Venda:

Cliente: #1

Data de cadastro: 04/03/2021

Nome: Fulano de Tal

CPF: 123.456.789-10

Data: 09/03/2021

Total de produtos na compra: 2

1. iPhone 1 R\$ 6999.0

2. Carregador com fio 1 R\$ 69.9

Valor total: R\$ 7068.9

Parcelas:

1. 09/03/2021 R\$ 3534.45

2. 09/04/2021 R\$ 3534.45

Dados da Venda:

Cliente: #2

Data de cadastro: 09/12/2019

Razão Social: Beltrana & Beltrano comércio de eletrônicos

CNPJ: 12.345.678/0009-10

Data: 09/03/2021

Total de produtos na compra: 2

1. iPhone 200 R\$ 1399800.0

2. Pear watch 200 R\$ 599800.0

Valor total: R\$ 1999600.0

Parcelas:

1. 09/03/2021 R\$ 1999600.0

Atividades

- Entrega de um arquivo **.zip** (contendo a estrutura de diretórios da aplicação e os sete arquivos .java e uma imagem/PDF)
 - Entregue o diagrama de classe do Exercício #1 Parte #1
 - Entregue a implementação do Exercício #1 Partes #2 a #3
 - Entrega até às 23:55h de 18/03/2021



Programação Orientada a Objeto

14. Herança e Ligação Dinâmica (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020