



Módulo 2

PROGRAMAÇÃO DE COMPUTADORES III

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Profa Gyslla Vasconcelos
2025-2

O que é o workspace?

É uma pasta principal que o Eclipse usa para guardar:

- seus projetos Java
- configurações do Eclipse
- histórico, atalhos, etc.

 C:\POO\WrkModulo01

Dentro dela ficarão várias pastas de projetos

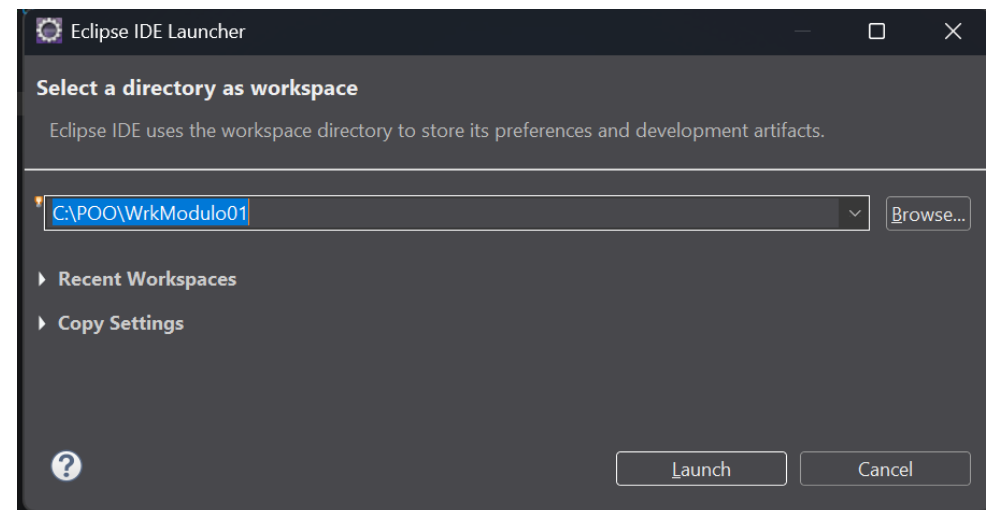
**Pense nela como sua
área de trabalho dentro
do Eclipse**

Zerando um workspace

Resetar preferências da IDE

Se quiser manter os arquivos, mas limpar a configuração da IDE:

- Com o Eclipse aberto, vá em [File] [Switch Workspace] [Other...]
- Escolha uma nova pasta vazia para ser o workspace
- Clique em Launch
- O Eclipse abrirá sem nenhum projeto, como se fosse a primeira vez
- Siga os passos feitos no Módulo 1 para configuração inicial



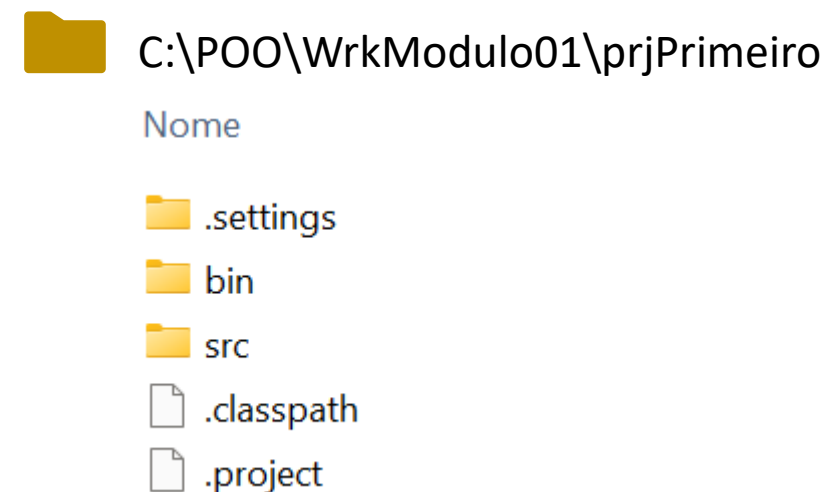
O que é o projeto?

É uma pasta dentro do workspace que representa um programa Java.

Dentro do projeto você terá:

- as classes compiladas (bin/)
- o código-fonte (src/)
- configurações específicas do projeto.

Cada projeto é independente!
Você pode ter vários projetos
diferentes no mesmo
workspace



O que é o pacote?

Um pacote (package) é um agrupamento lógico de classes. Ajuda a organizar o código, evitando conflito de nomes. Pacotes viram subpastas dentro da pasta src/.

 C:\POO\WrkModulo01\prjPrimeiro\src\controller\CtrlPrograma.java

- Nessa pasta o **controller** é o pacote
- **CtrlPrograma.java** é a classe dentro do pacote

**Por que o nome
controller?**

Por que o nome controller?

Vem de um padrão de organização chamado MVC (Model–View–Controller)

- Model – dados e regras de negócio
- View - interface com o usuário
- Controller - controle, o que conecta model e view

**Antecipação de uma
boa prática**

O que é Arquitetura MVC (Model View Controller):

<https://www.youtube.com/watch?v=iSv2QNcRPxs>

Classe?

É o código Java em si!

Toda lógica do programa está dentro das classes

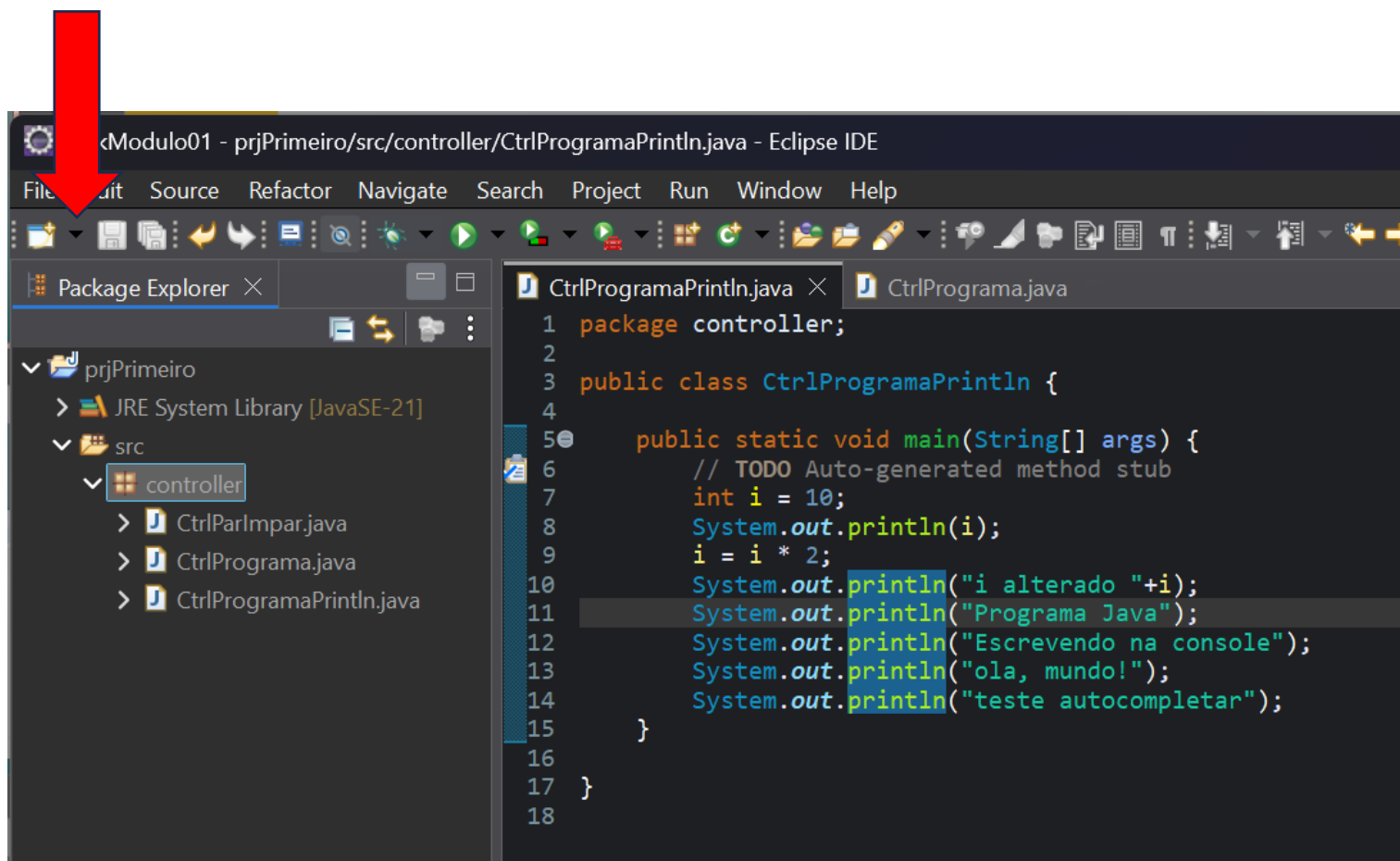
```
package controller;

public class CtrlPrograma {
    public static void main(String[] args) {
        System.out.println("Olá, mundo!");
    }
}
```

**O nome do programa deve
ser o mesmo da classe**

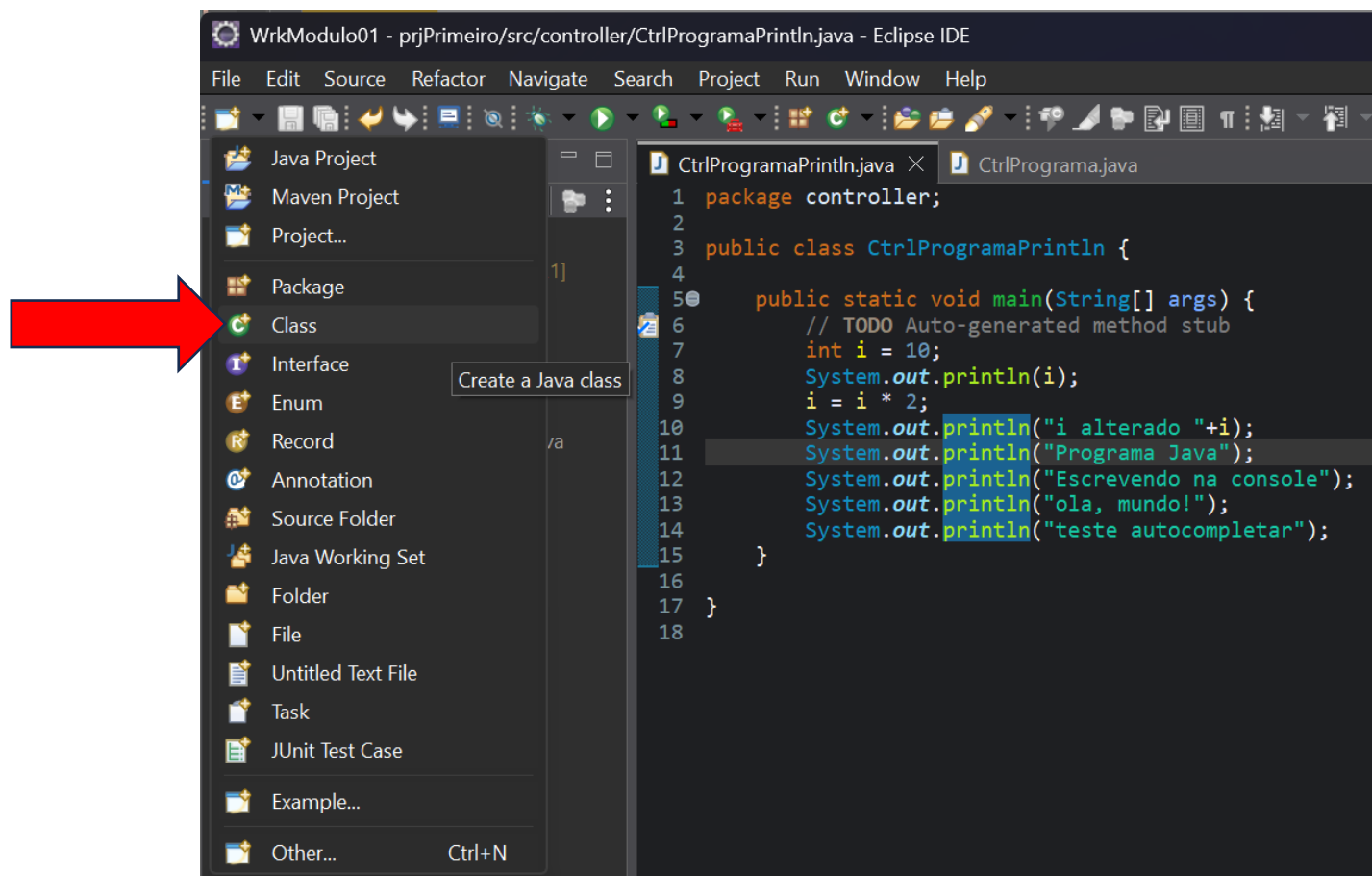
Como fazer os exercícios?

Modo 1



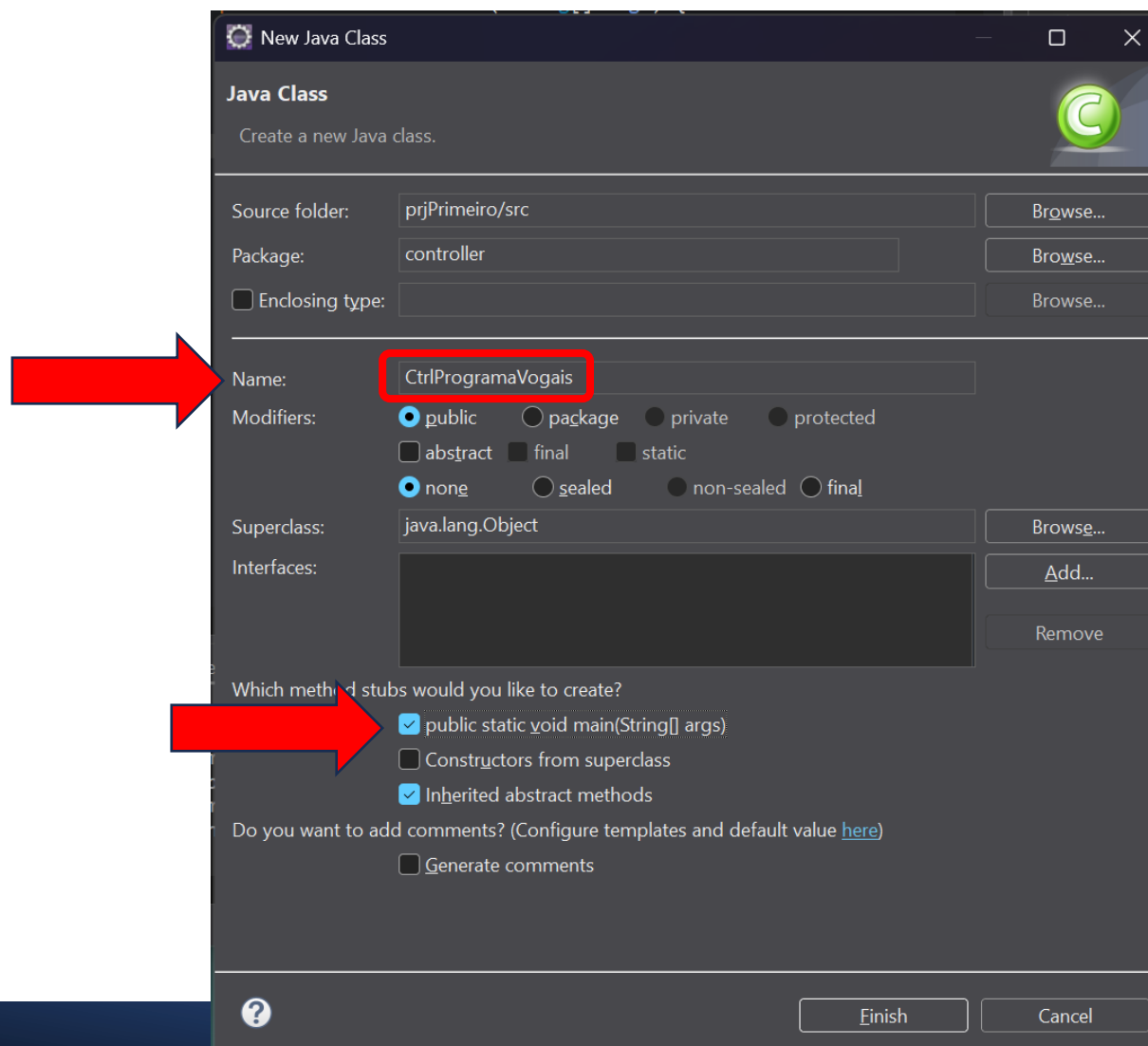
Como fazer os exercícios?

Modo 1



Como fazer os exercícios?

Modo 1



Como fazer os exercícios?

Modo 1

WrkModulo01 - prjPrimeiro/src/controller, CtrlProgramaVogais.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

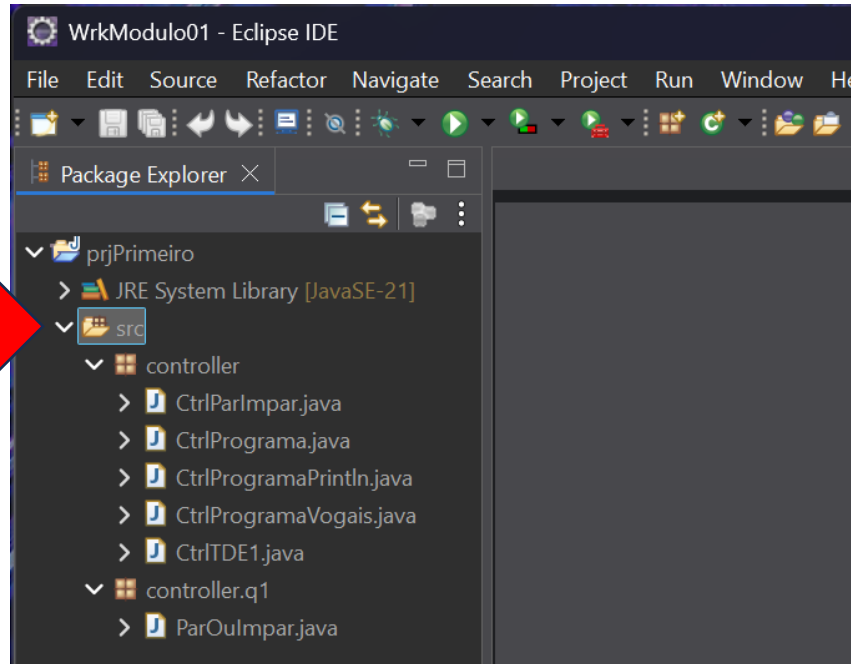
- prjPrimeiro
 - JRE System Library [JavaSE-21]
 - src
 - controller
 - CtrlParImpar.java
 - CtrlPrograma.java
 - CtrlProgramaPrintln.java
 - CtrlProgramaVogais.java

```

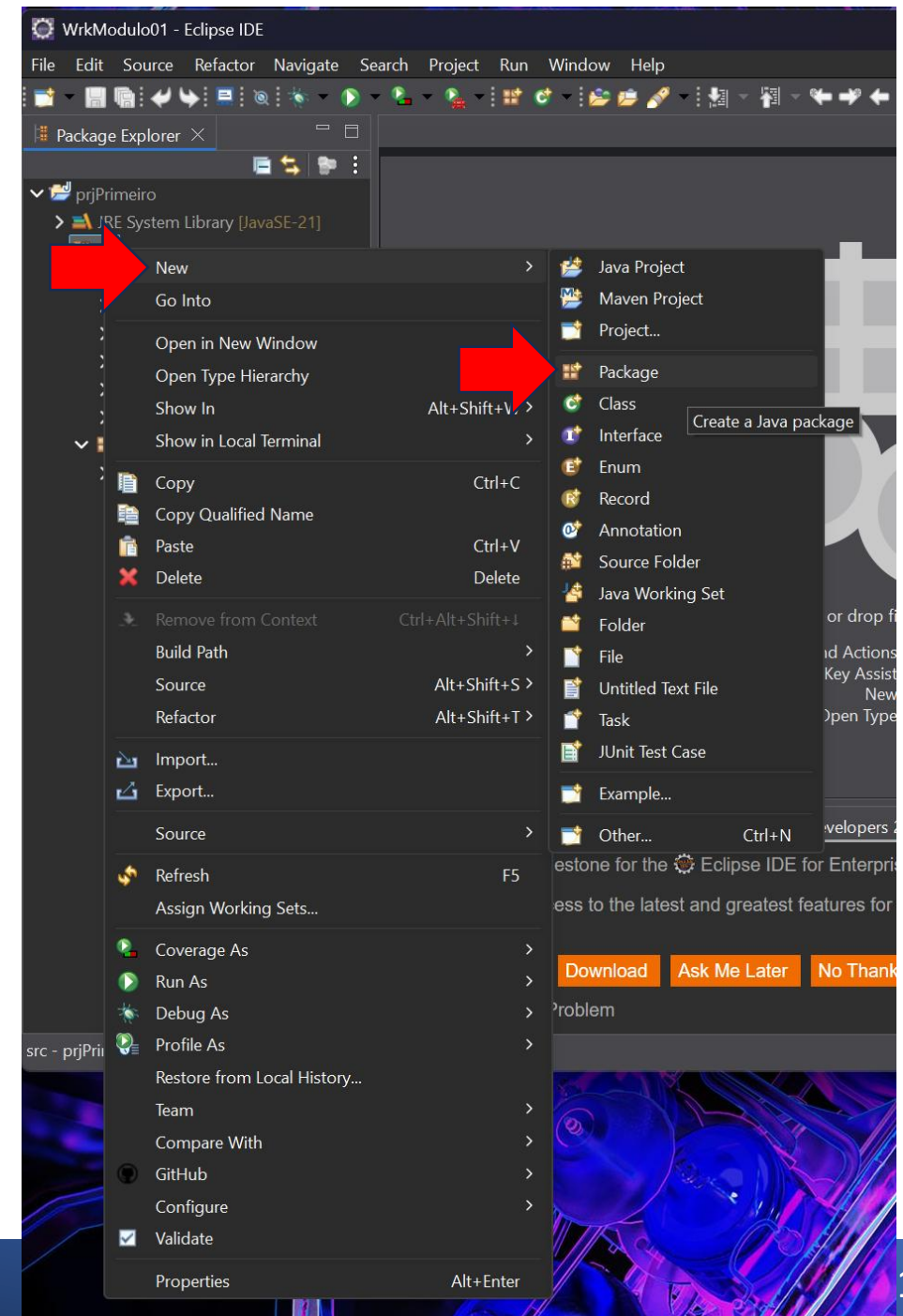
1 package controller;
2
3 public class CtrlProgramaVogais {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
  
```

Como fazer os exercícios?

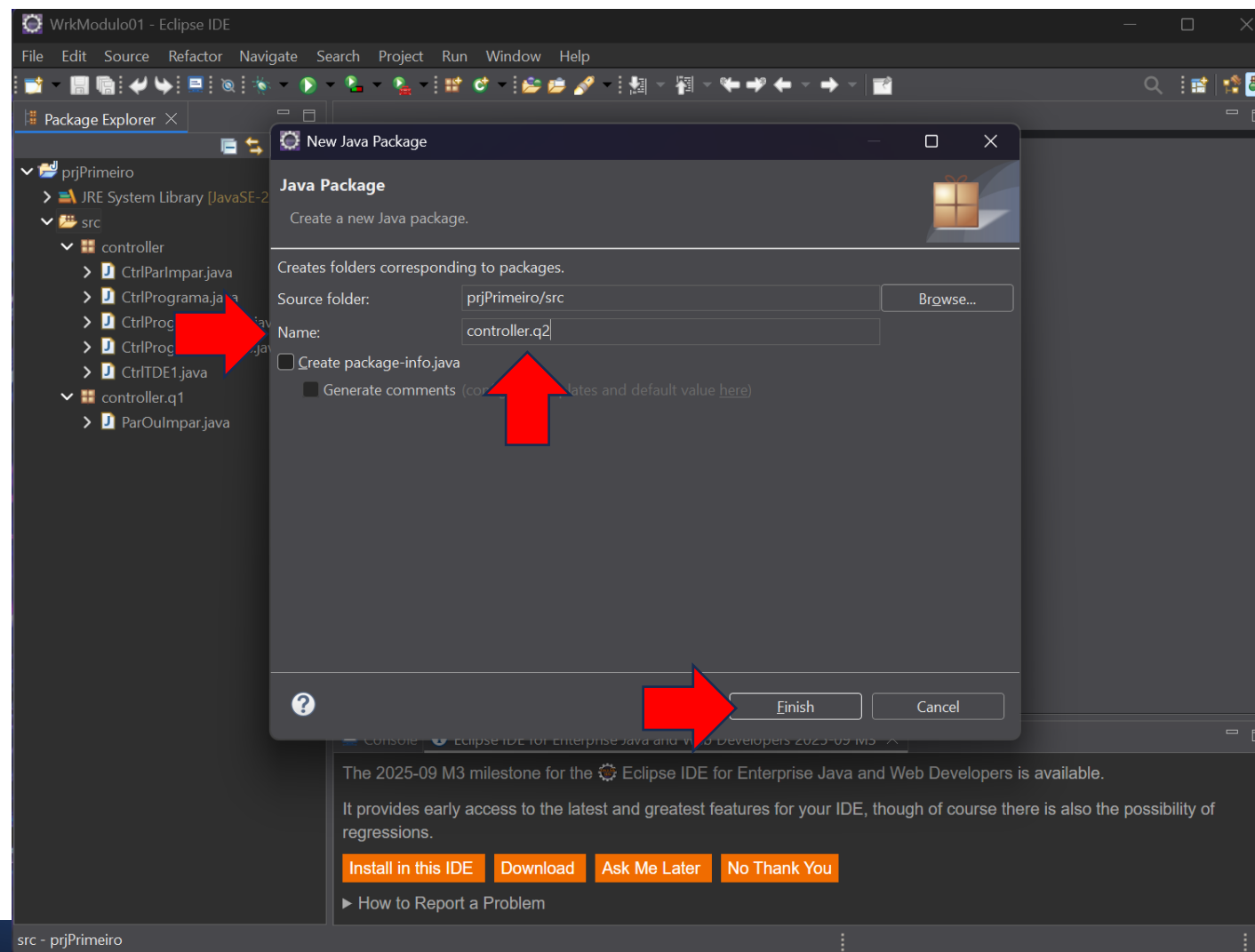
Botão direito



Modo 2
Criando sub pacotes

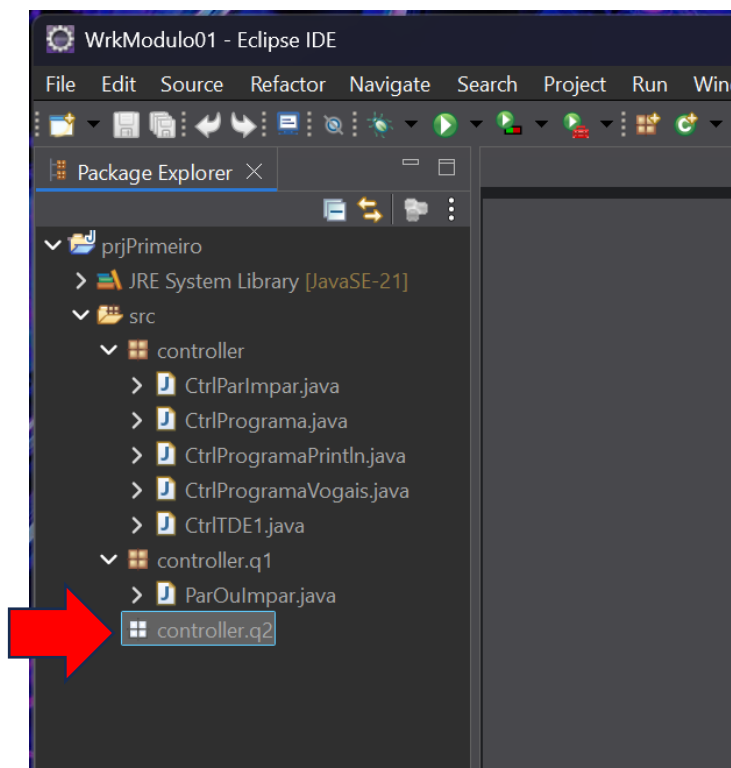


Como fazer os exercícios?

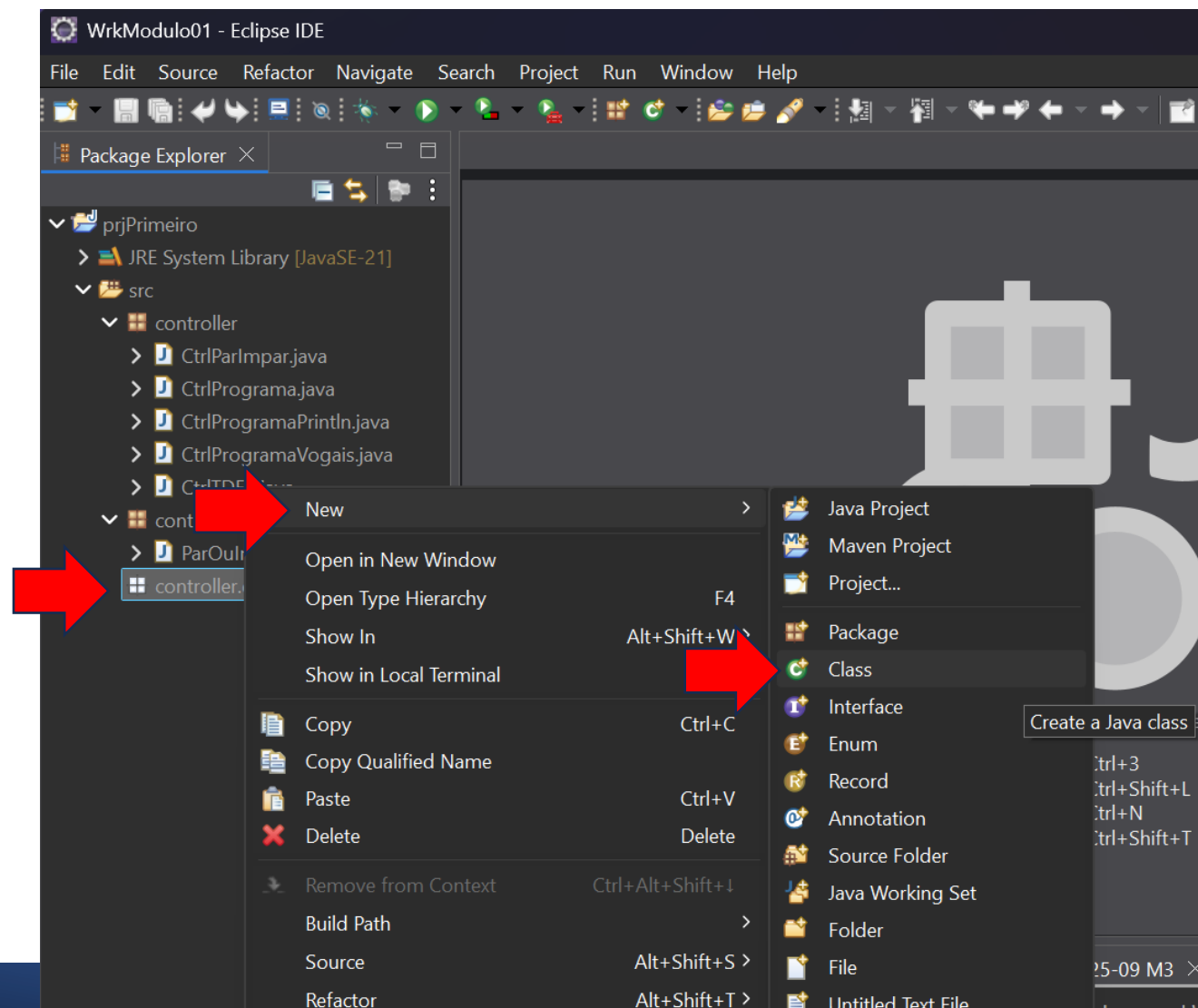


Em nome escreva o nome do pacote, seguido de “.” e o nome da subpasta. Pr exemplo: controller.q2

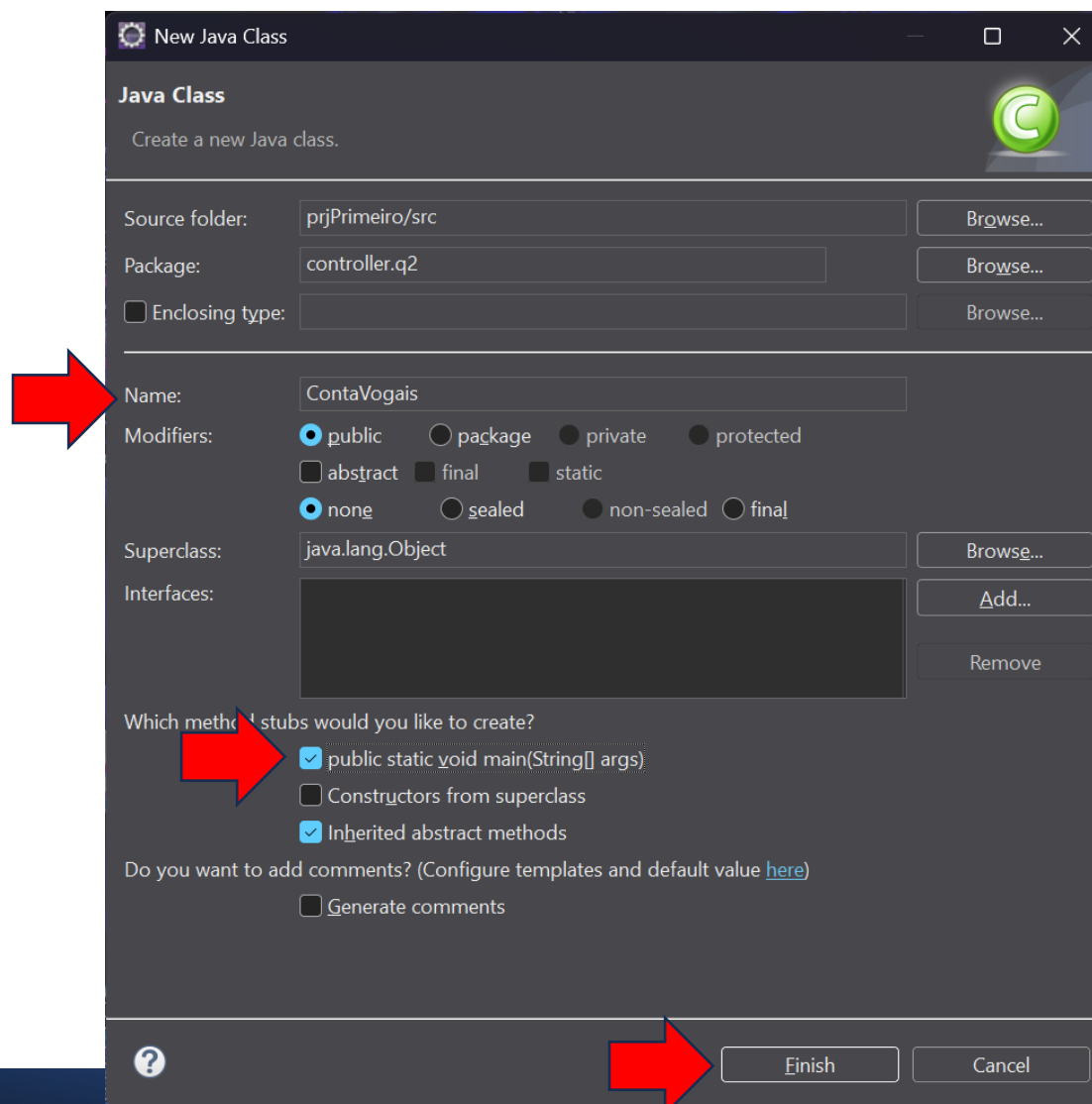
Como fazer os exercícios?



Modo 2
Criando sub pacotes



Como fazer os exercícios?



New Java Class

Create a new Java class.

Source folder: prjPrimeiro/src Browse...

Package: controller.q2 Browse...

☐ Enclosing type: Browse...

Name: ContaVogais

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

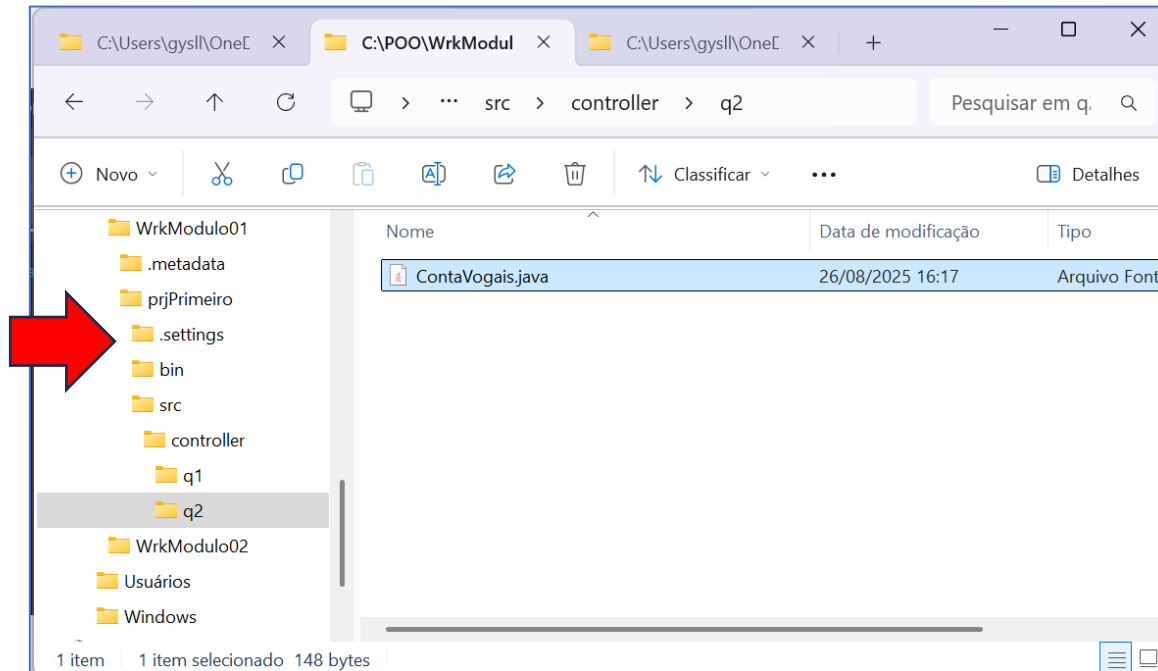
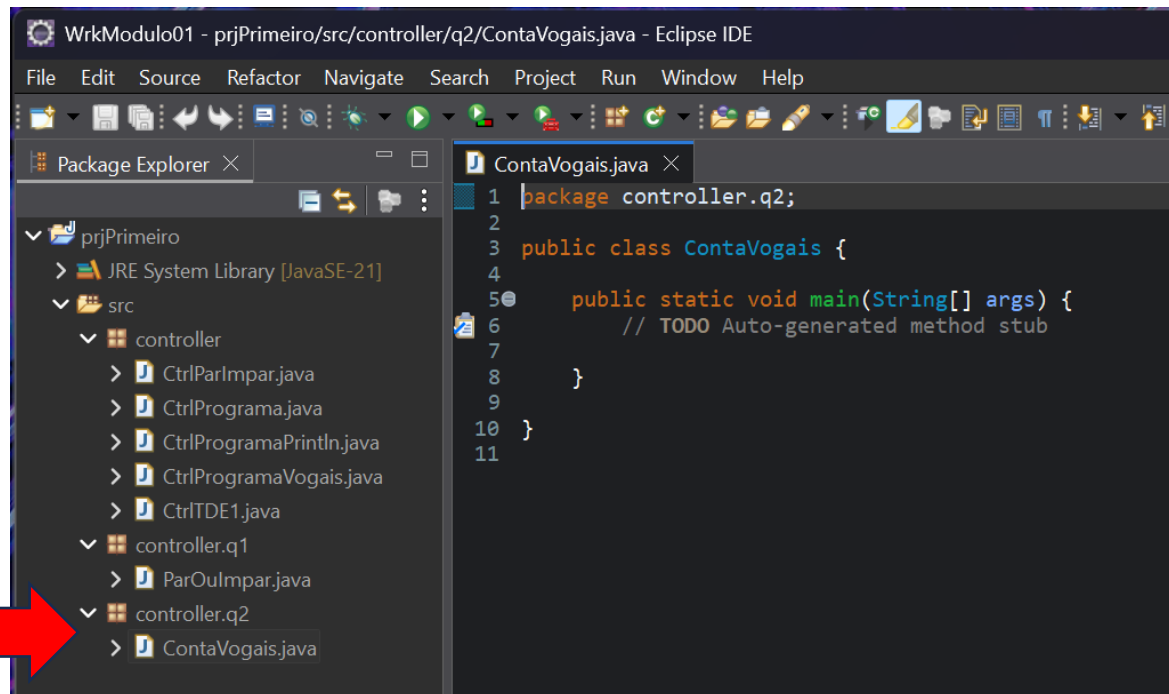
Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments


? Finish Cancel

Modo 2
Criando sub pacotes

Como fazer os exercícios?

Modo 2 Criando sub pacotes

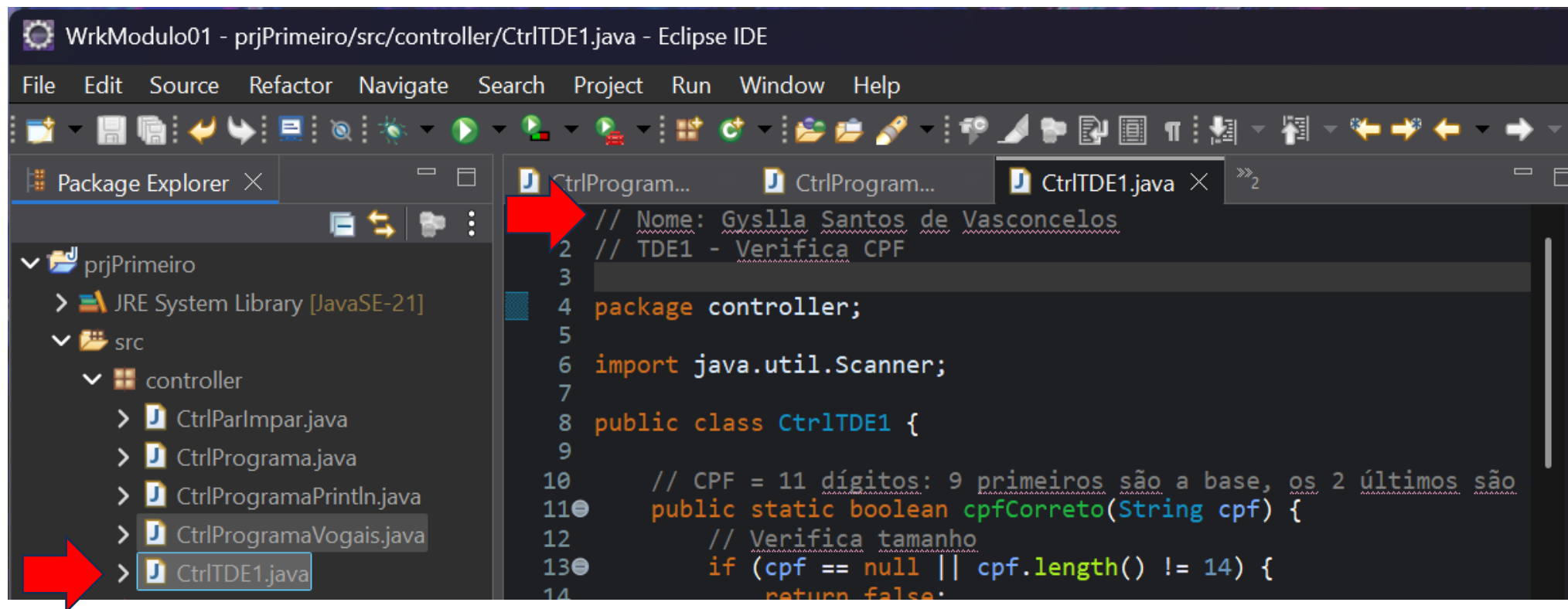


 C:\POO\WrkModulo01\prjPrimeiro\src\controller\

Nome

- q1
- q2
- CtrlParImpar.java
- CtrlPrograma.java
- CtrlProgramaPrintln.java
- CtrlProgramaVogais.java
- CtrlTDE1.java

Como fazer os TDEs?



2

Módulo 2: Primeiros Conceitos do Modelo Orientado a Objetos

Módulo 2

O que veremos!

1. Introdução
2. Paradigma Orientado a Objetos (OO)
3. Objetos e Suas Características
4. Classes vs. Objetos
5. Exemplos práticos de implementação em Java

Objetivos

Nesta aula, exploraremos o conceito de Programação Orientada a Objetos (POO), entendendo como classes são utilizadas para organizar e estruturar programas de forma modular e reutilizável.

O que vamos aprender?

- Paradigma Orientado a Objetos (OO)
- Objetos e Suas Características
- Classes vs. Objetos
- Exemplos práticos de implementação em **Java**

Paradigma OO

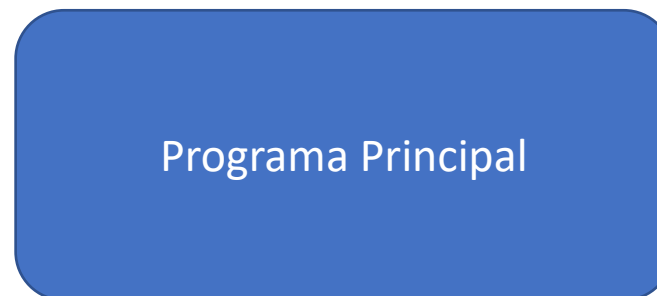
Organização do programa

Paradigma procedimental versus OO

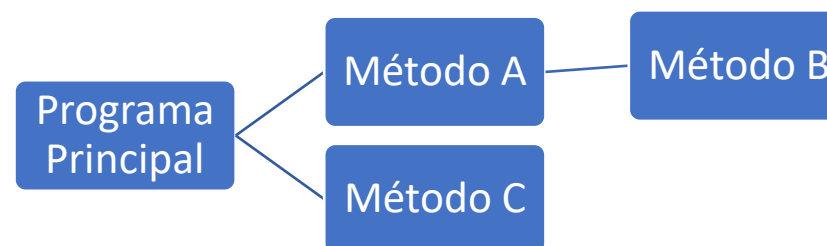
- O **paradigma procedimental** organiza o programa em termos de **algoritmos**
- O **paradigma OO** organiza o programa em termos de **objetos**



- Antes: um programa gigante



- Depois: vários programas menores



- Podemos criar programa pensando em termos de **objetos** ao invés de **algoritmos**?

~~Algoritmos~~Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos

~~Algoritmos~~Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.

~~Algoritmos~~Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.

AlgoritmosObjetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.
 - Uma universidade tem professores, alunos, disciplinas, etc.

~~Algoritmos~~Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.
 - Uma universidade tem professores, alunos, disciplinas, etc.
 - Uma rodoviária tem ônibus, passageiros, bagagens, etc.

~~Algoritmos~~Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.
 - Uma universidade tem professores, alunos, disciplinas, etc.
 - Uma rodoviária tem ônibus, passageiros, bagagens, etc.
- E se **criarmos programas** basicamente **criando objetos** equivalentes ao mundo real, e fazendo com que esses **objetos se comuniquem?**

AlgoritmosObjetos

Objetos?

Objetos

- Definição
 - Um objeto é a **representação computacional de um elemento ou processo do mundo real**

Objetos

- Definição
 - Um objeto é a **representação computacional de um elemento ou processo do mundo real**
 - Cada objeto possui suas **características** e seu **comportamento**

Objetos

- Definição
 - Um objeto é a **representação computacional de um elemento ou processo do mundo real**
 - Cada objeto possui suas **características** e seu **comportamento**

- Exemplos de Objetos

cadeira

mesa

caneta

lápis

carro

piloto

venda

mercadoria

cliente

aula

programa

computador

aluno

avião

Características de Objetos?

Características de Objetos

- Definição
 - Uma **característica** descreve uma **propriedade** de um objeto, ou seja, algum elemento que descreva o objeto.

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**
 - Cor

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**
 - Cor
 - Marca

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**
 - Cor
 - Marca
 - Número de portas

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**
 - Cor
 - Marca
 - Número de portas
 - Ano de fabricação

Características de Objetos

- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto

- Exemplo de características do objeto **carro**
 - Cor
 - Marca
 - Número de portas
 - Ano de fabricação
 - Tipo de combustível

Comportamento de Objetos?

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto
- Exemplos de comportamento para o objeto **carro**

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto
- Exemplos de comportamento para o objeto **carro**
 - Acelerar

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto

- Exemplos de comportamento para o objeto **carro**
 - Acelerar
 - Frear

Comportamento de Objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto

- Exemplos de comportamento para o objeto **carro**
 - Acelerar
 - Frear
 - Virar para direita

Comportamento de Objetos

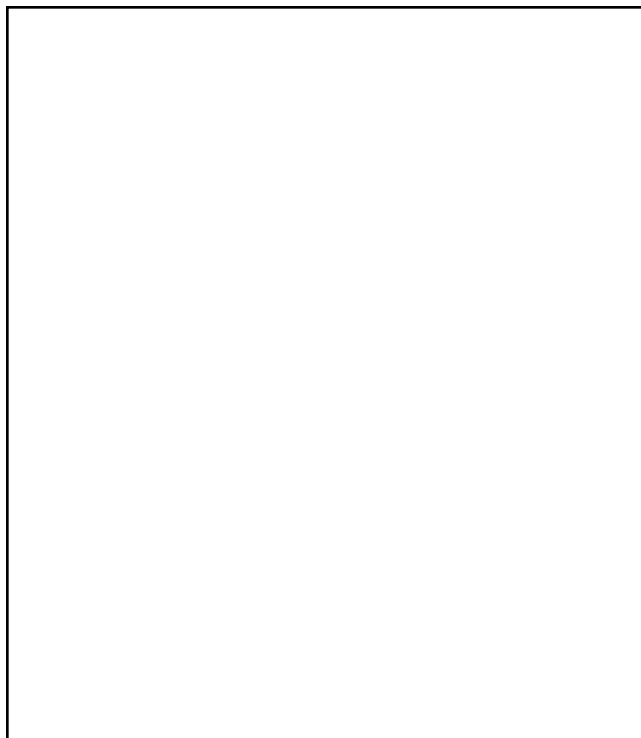
- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto

- Exemplos de comportamento para o objeto **carro**
 - Acelerar
 - Frear
 - Virar para direita
 - Virar para esquerda

Mapeamento de Objetos?

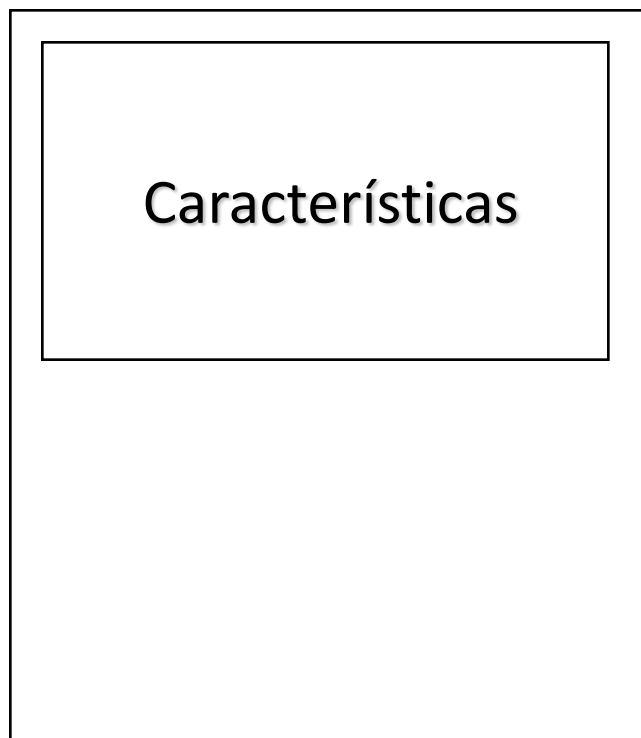
Mapeamento de Objetos

Objeto no Mundo Real



Mapeamento de Objetos

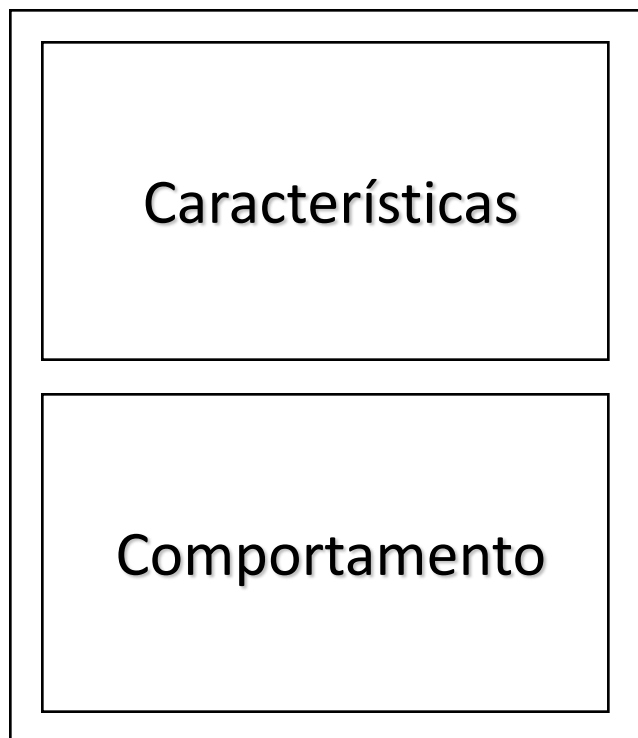
Objeto no Mundo Real



Cor
Marca
Número de portas
Ano de fabricação
Tipo de combustível

Mapeamento de Objetos

Objeto no Mundo Real

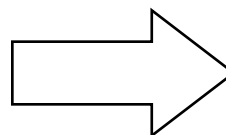
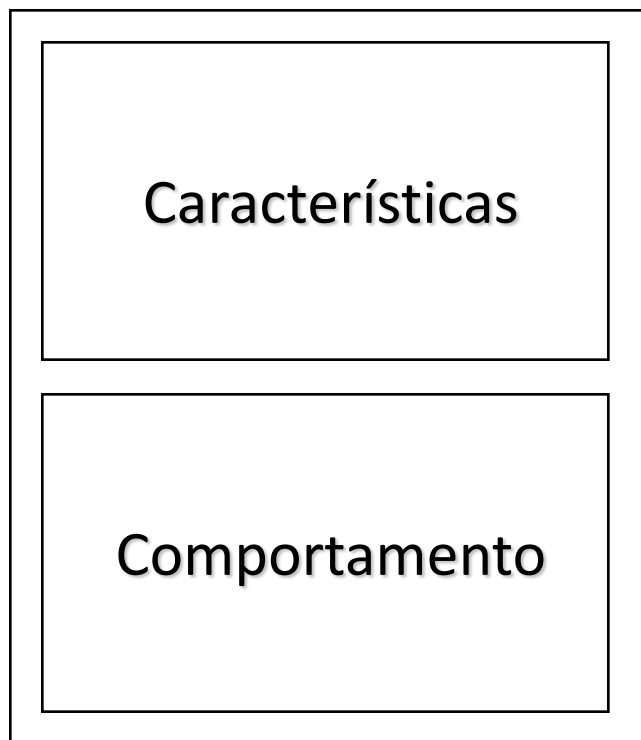


Cor
Marca
Número de portas
Ano de fabricação
Tipo de combustível

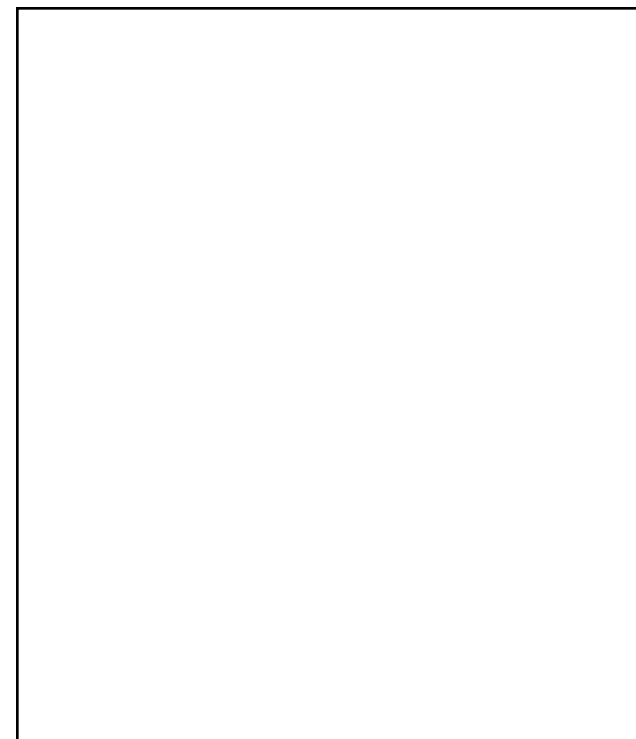
Acelerar
Frear
Virar para direita
Virar para esquerda

Mapeamento de Objetos

Objeto no Mundo Real



Objeto Computacional

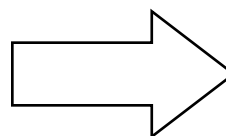
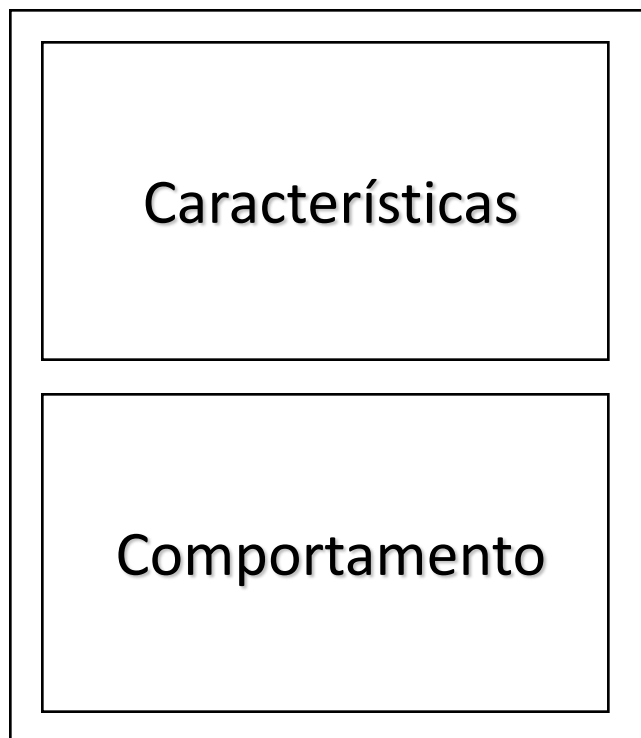


Cor
Marca
Número de portas
Ano de fabricação
Tipo de combustível

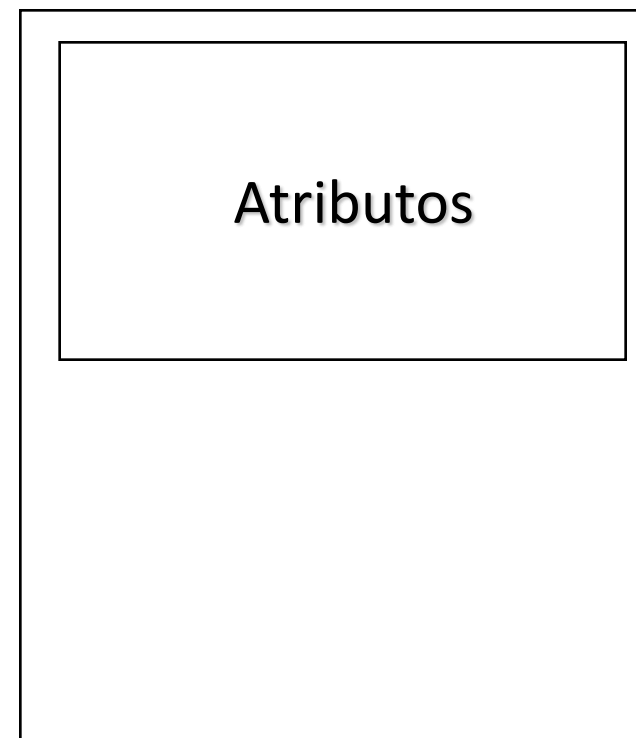
Acelerar
Frear
Virar para direita
Virar para esquerda

Mapeamento de Objetos

Objeto no Mundo Real



Objeto Computacional

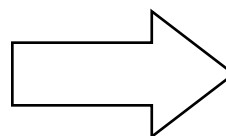
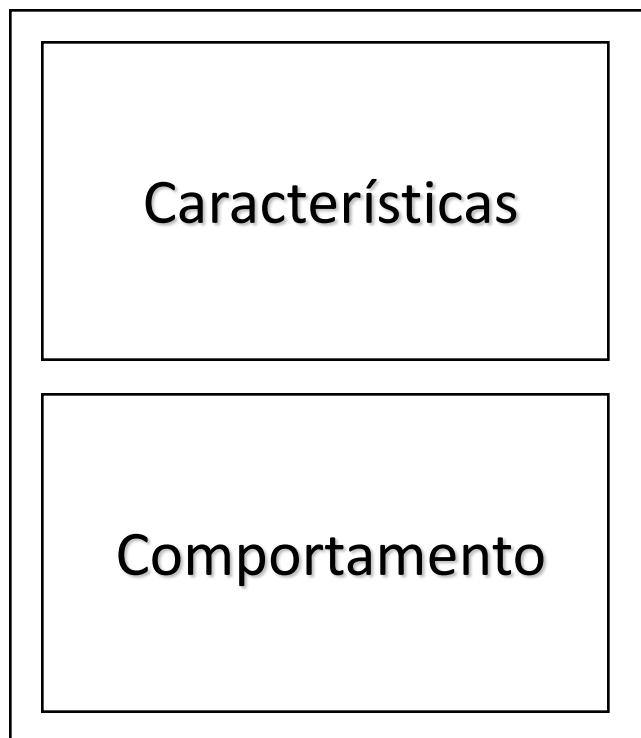


Cor
 Marca
 Número de portas
 Ano de fabricação
 Tipo de combustível

Acelerar
 Frear
 Virar para direita
 Virar para esquerda

Mapeamento de Objetos

Objeto no Mundo Real



Objeto Computacional



Cor
 Marca
 Número de portas
 Ano de fabricação
 Tipo de combustível

Acelerar
 Frear
 Virar para direita
 Virar para esquerda

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo Características

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo Características
 - Vetor de Contatos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome
 - Endereço
 - Telefone

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome
 - Endereço
 - Telefone
 - Métodos

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome
 - Endereço
 - Telefone
 - Métodos
 - Exibição de nome, endereço e telefone

Exemplo: Implementando agenda telefônica

Lista de nomes, com telefone e endereço

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo **Características**
 - Vetor de Contatos
 - Métodos **Comportamento**
 - Listar Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome
 - Endereço
 - Telefone
 - Métodos
 - Exibição de nome, endereço e telefone
 - Edição de nome, endereço e telefone

Exemplo: Total da compra

Lista de produtos e preços comprados

Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00

Quais são os objetos envolvidos?

Exemplo: Total da compra

Lista de produtos e preços comprados

Pedido: 12345
Cliente: João da Silva
Endereço: Rua dos Bobos, número zero

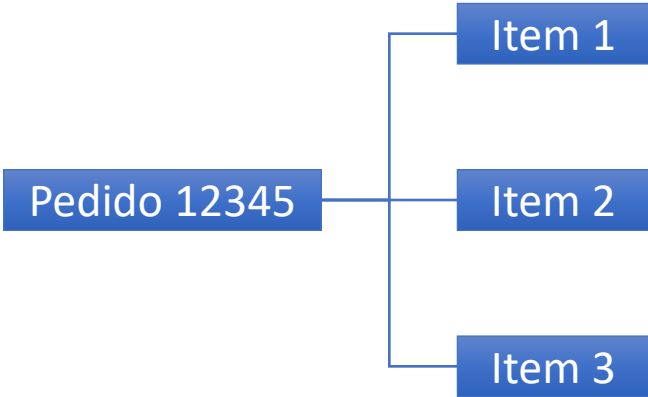
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
			TOTAL	R\$ 24,00

Pedido 12345

Exemplo: Total da compra

Lista de produtos e preços comprados

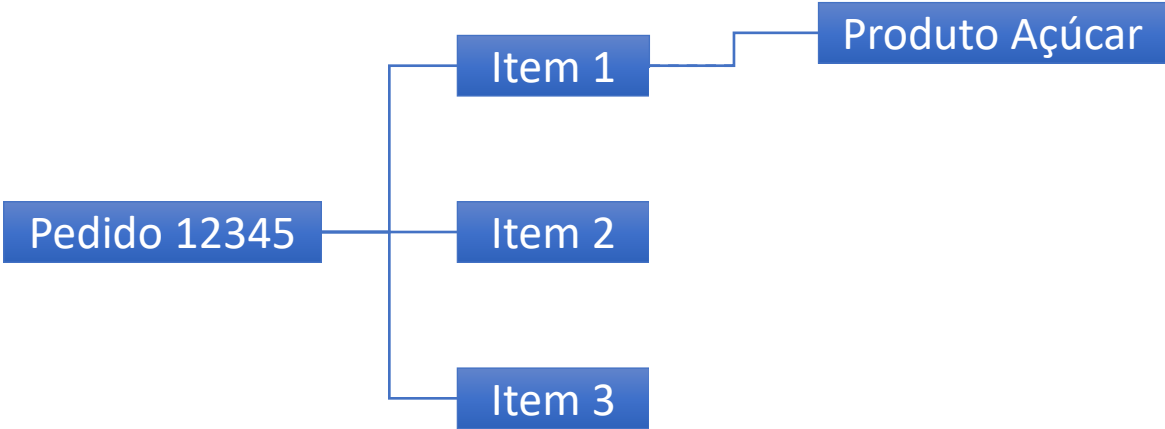
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

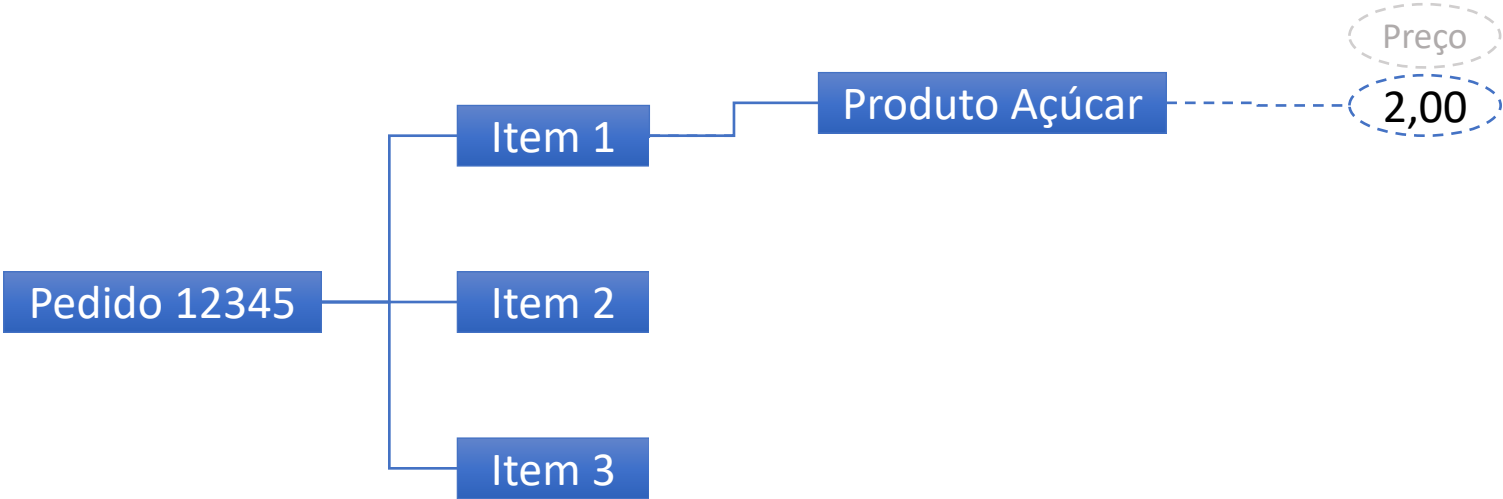
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

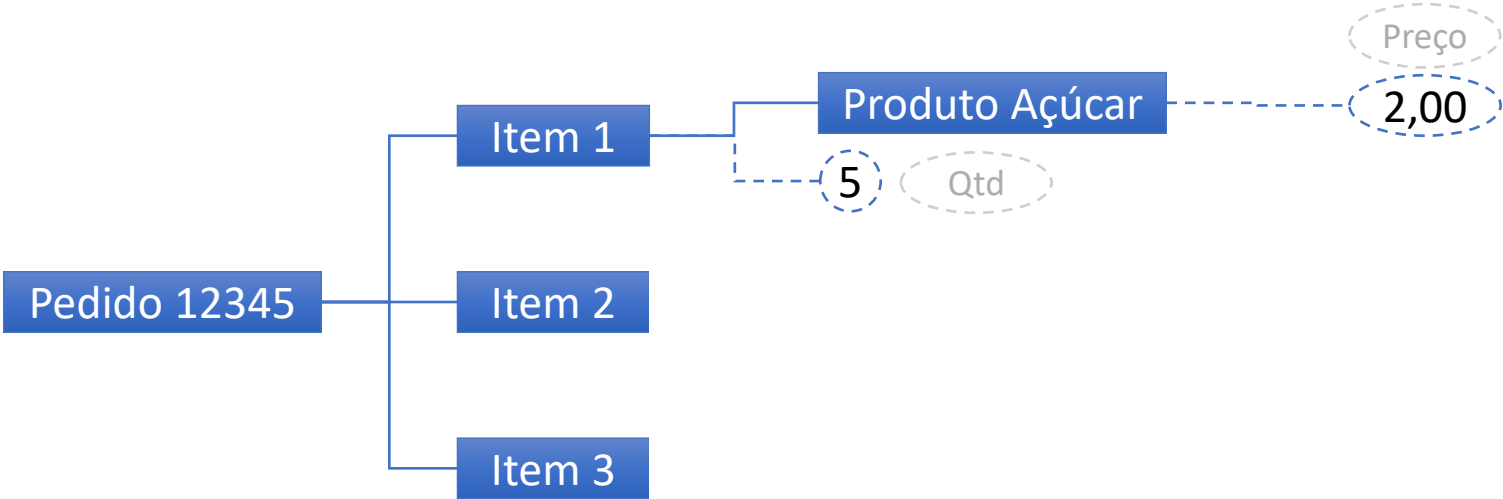
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

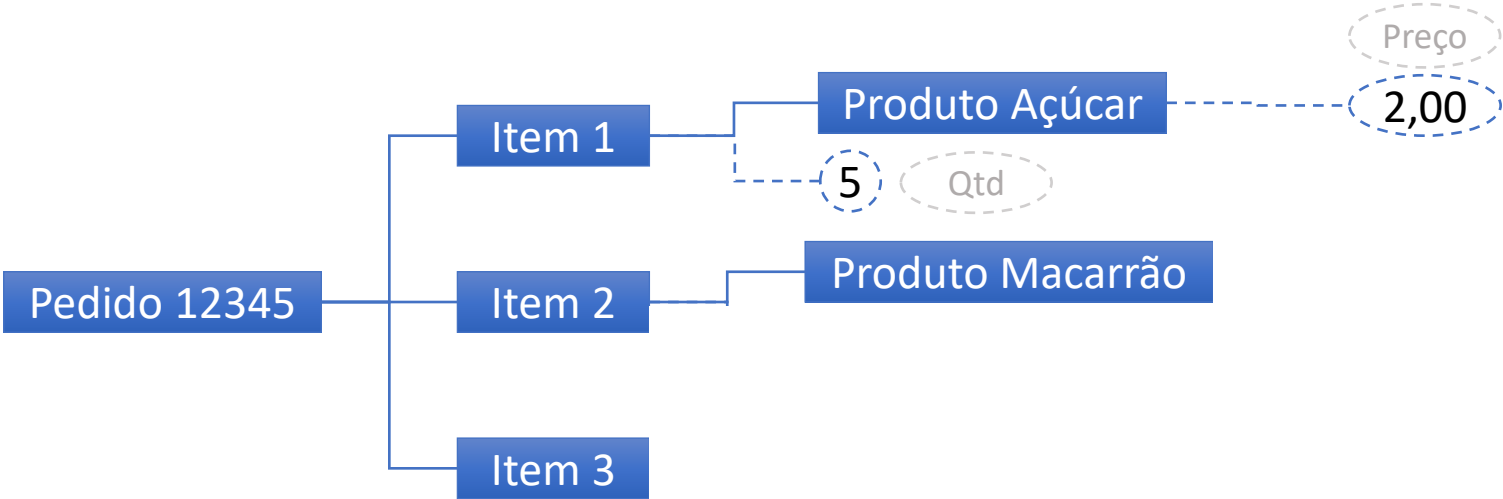
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

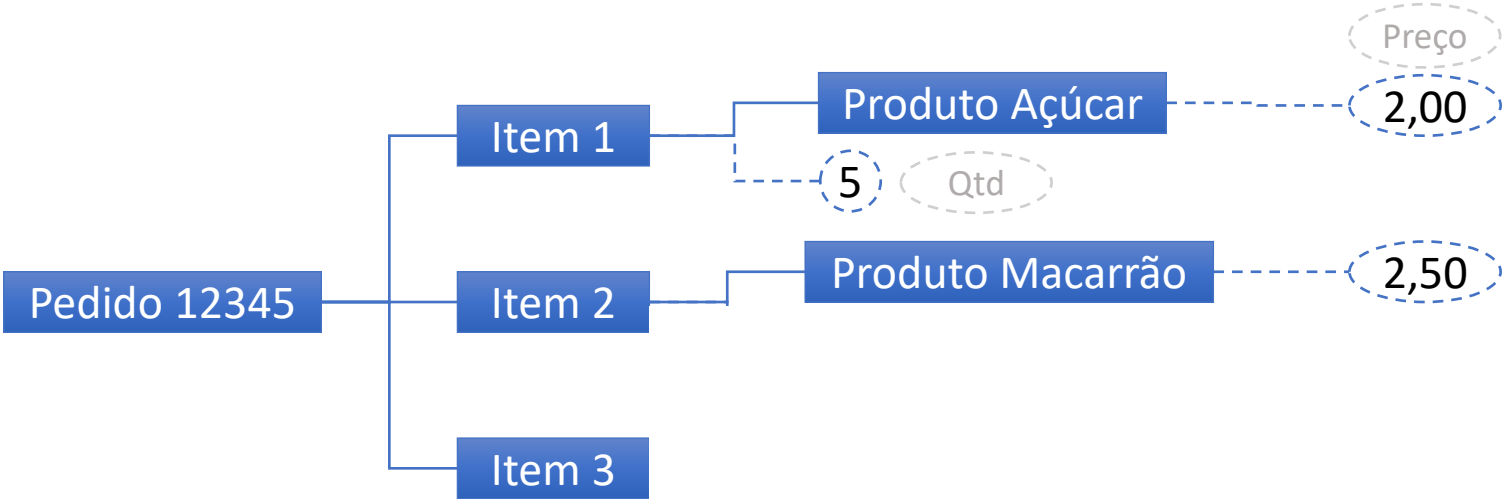
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

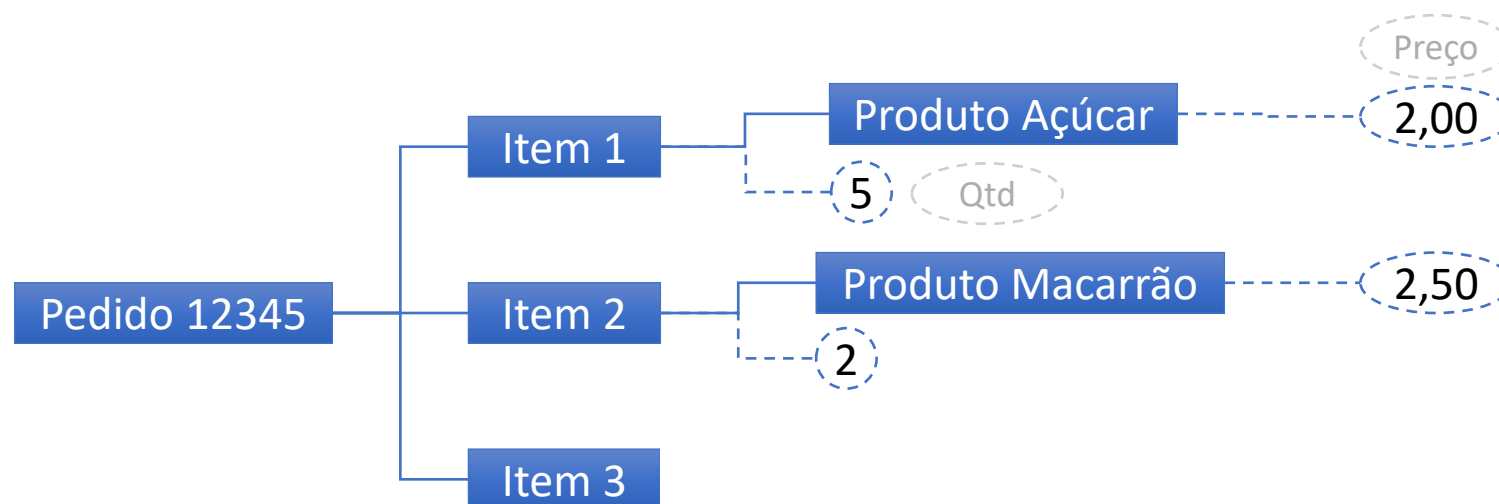
Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

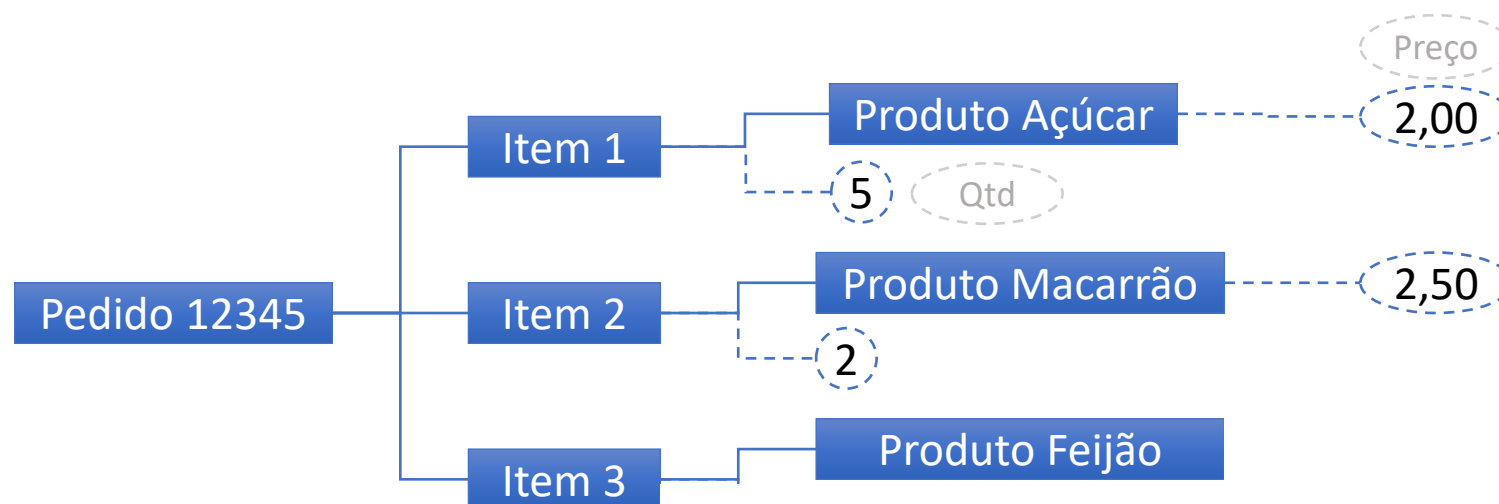
Pedido: 12345				
Cliente: João da Silva				
Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

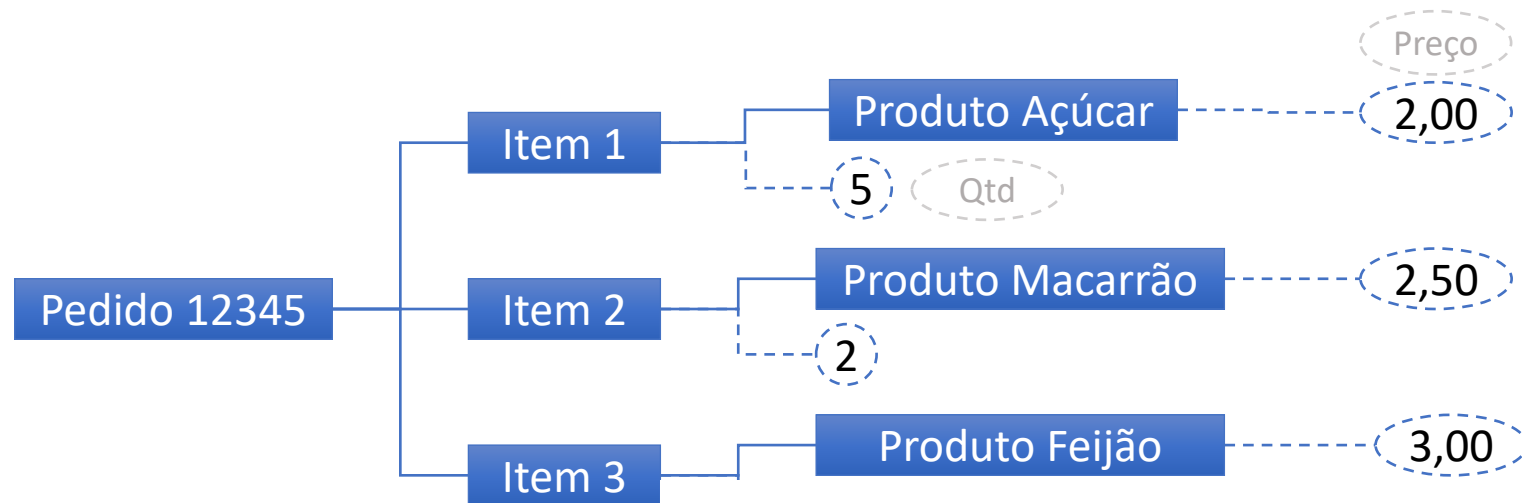
Pedido: 12345				
Cliente: João da Silva				
Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

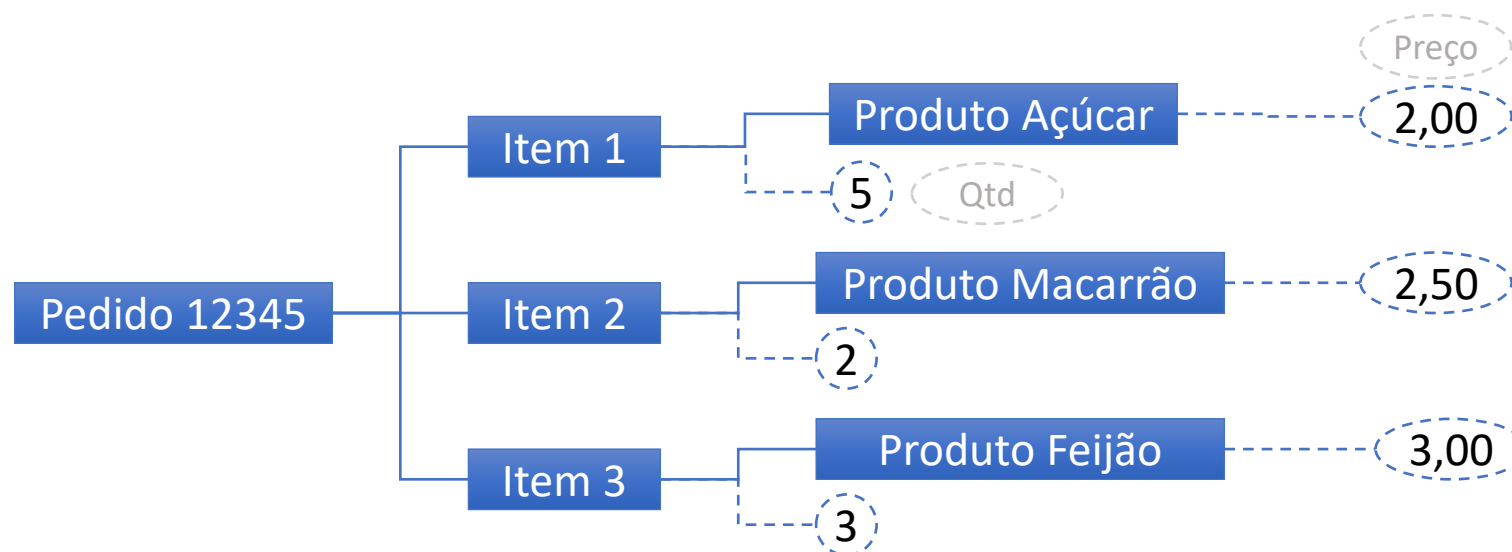
Pedido: 12345				
Cliente: João da Silva				
Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



Exemplo: Total da compra

Lista de produtos e preços comprados

Pedido: 12345				
Cliente: João da Silva				
Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00

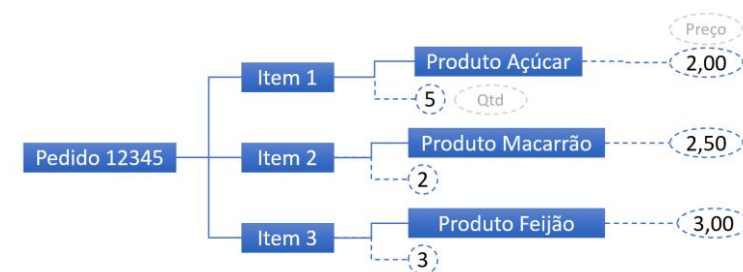


Exemplo: Total da compra

Lista de produtos e preços comprados

- Como obter o total da compra?

Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00

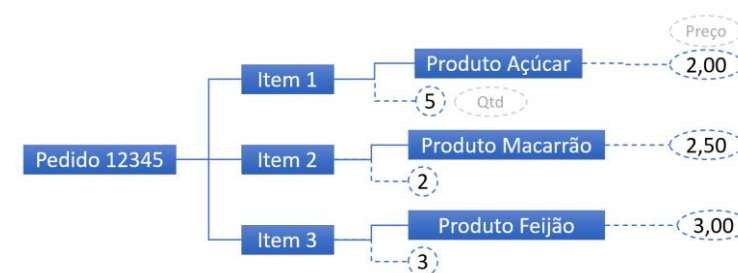


Exemplo: Total da compra

Lista de produtos e preços comprados

- Como obter o total da compra?
 1. O objeto **Caixa** pediria ao objeto **Pedido** seu valor total
 2. O objeto **Pedido**, por sua vez, percorreria todos os seus objetos **Item** perguntando o seu valor subtotal e somaria esses valores para responder ao objeto **Caixa**
 3. Cada objeto **Item** perguntaria ao objeto **Produto** o seu preço e multiplicaria esse preço pela quantidade que está sendo comprada, para responder ao objeto **Pedido**

Pedido: 12345 Cliente: João da Silva Endereço: Rua dos Bobos, número zero				
Item	Produto	Preço	Quantidade	Subtotal
1	Açúcar	R\$ 2,00	5	R\$ 10,00
2	Macarrão	R\$ 2,50	2	R\$ 5,00
3	Feijão	R\$ 3,00	3	R\$ 9,00
TOTAL				R\$ 24,00



CLASSES

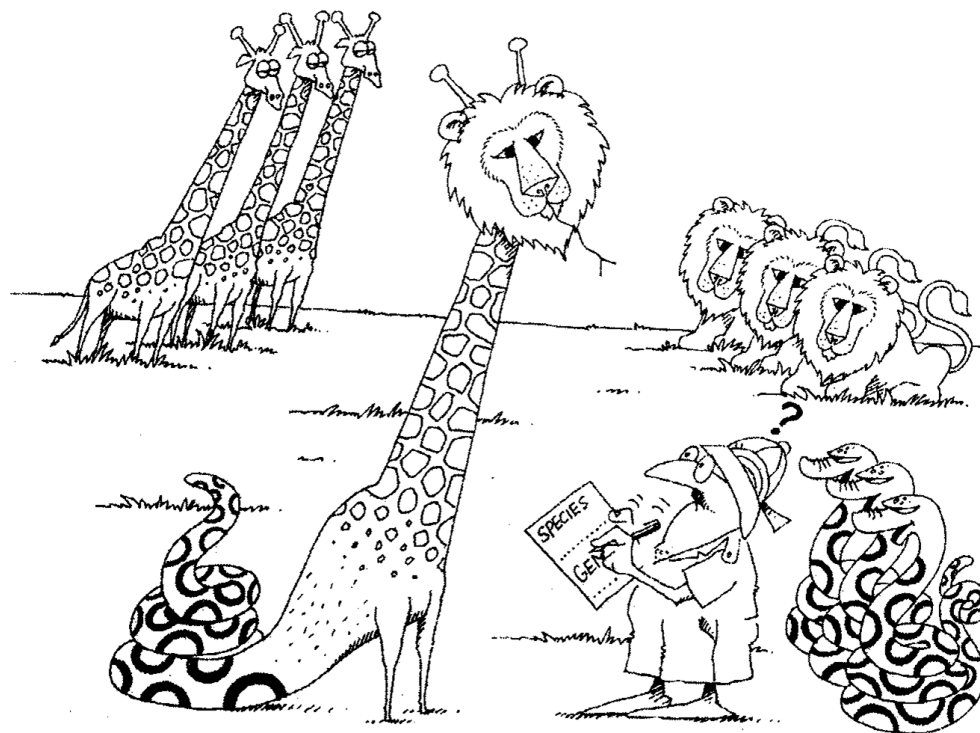
Um modelo para criar objetos

Classes versus Objetos

- A **Classe** é o **tipo do Objeto**

Classes versus Objetos

- A Classe é o tipo do Objeto



Fonte: livro “Object-Oriented Analysis and Design with Applications”

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe **Pessoa**

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe **Pessoa**
 - Fusca e Ferrari são da classe **Carro**

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe **Pessoa**
 - Fusca e Ferrari são da classe **Carro**
 - Flamengo e Fluminense são da classe **Time**

Classes versus Objetos

Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe **Pessoa**
 - Fusca e Ferrari são da classe **Carro**
 - Flamengo e Fluminense são da classe **Time**

Tipos Primitivos

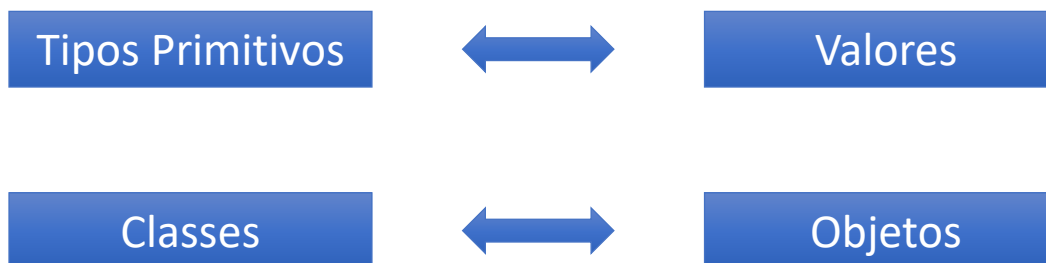


Valores

Classes versus Objetos

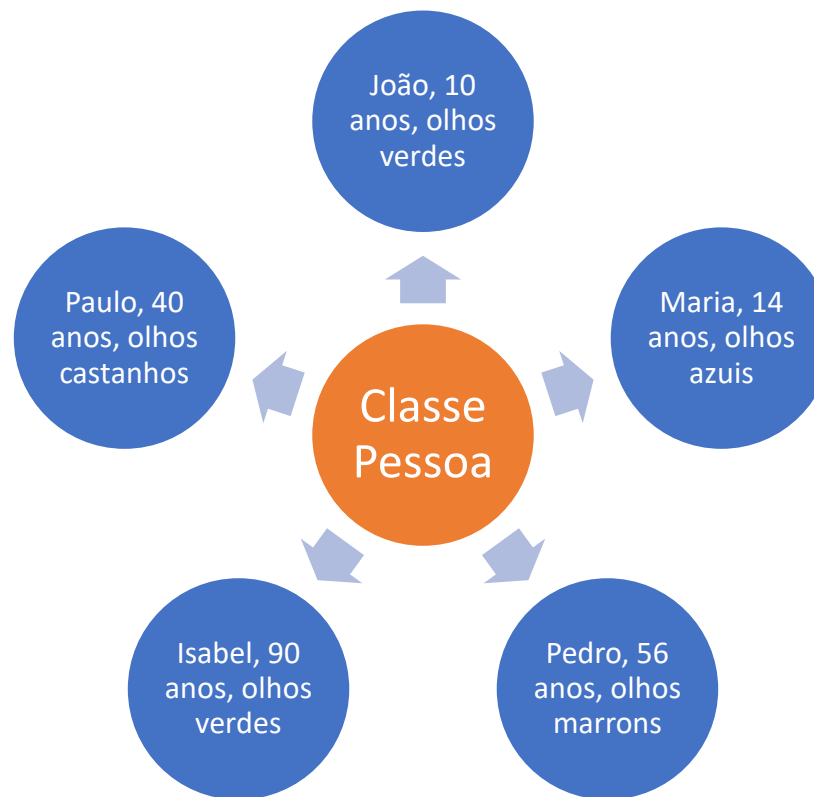
Vamos começar com analogia entre **tipos primitivos e valores**

- Valores pertencem a tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe **Pessoa**
 - Fusca e Ferrari são da classe **Carro**
 - Flamengo e Fluminense são da classe **Time**



Classes versus Objetos

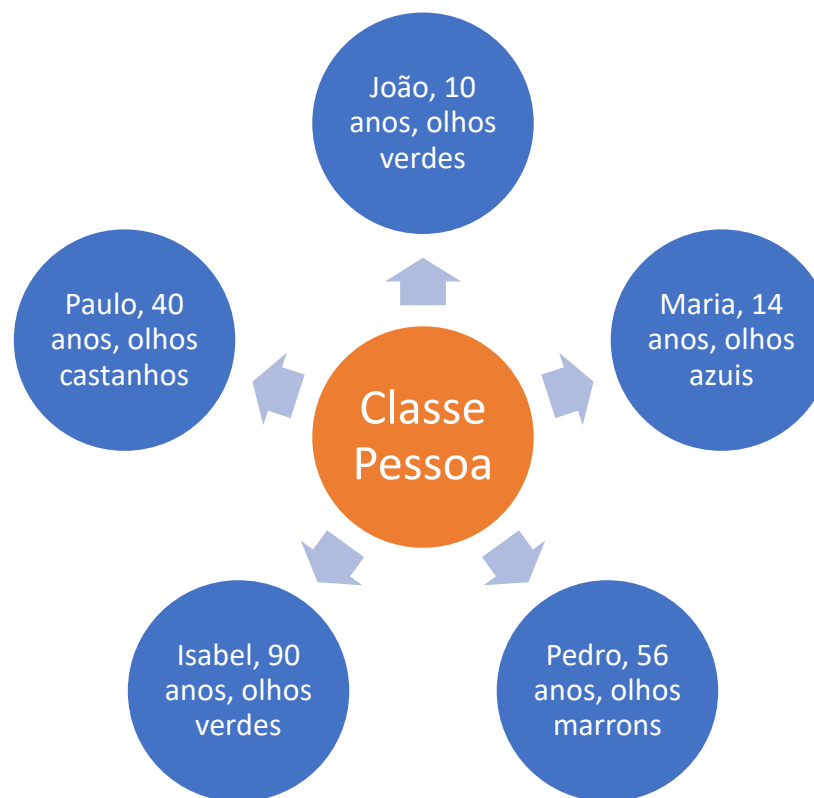
- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores criam classes**, as **classes instanciam objetos**



Classes versus Objetos

- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores criam classes**, as **classes instanciam objetos**

uma **classe** é um modelo ou estrutura que define quais características (**atributos**) e quais ações (**métodos**) um conjunto de objetos terá

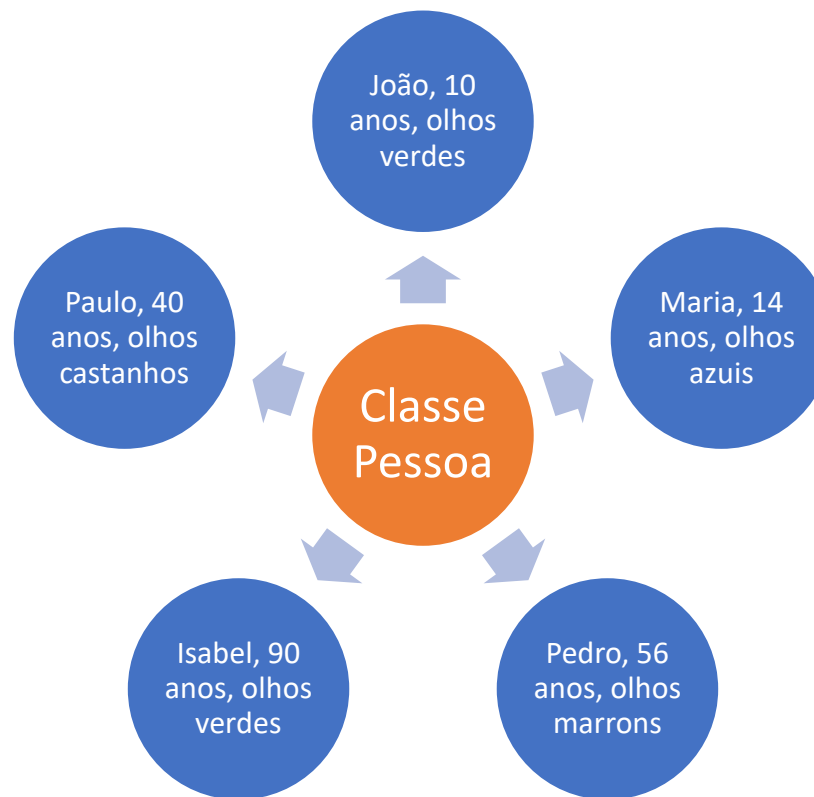


Classes versus Objetos

- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores criam classes**, as **classes instanciam objetos**

Em **Java**, um **objeto** é uma **instância** de uma **classe**.

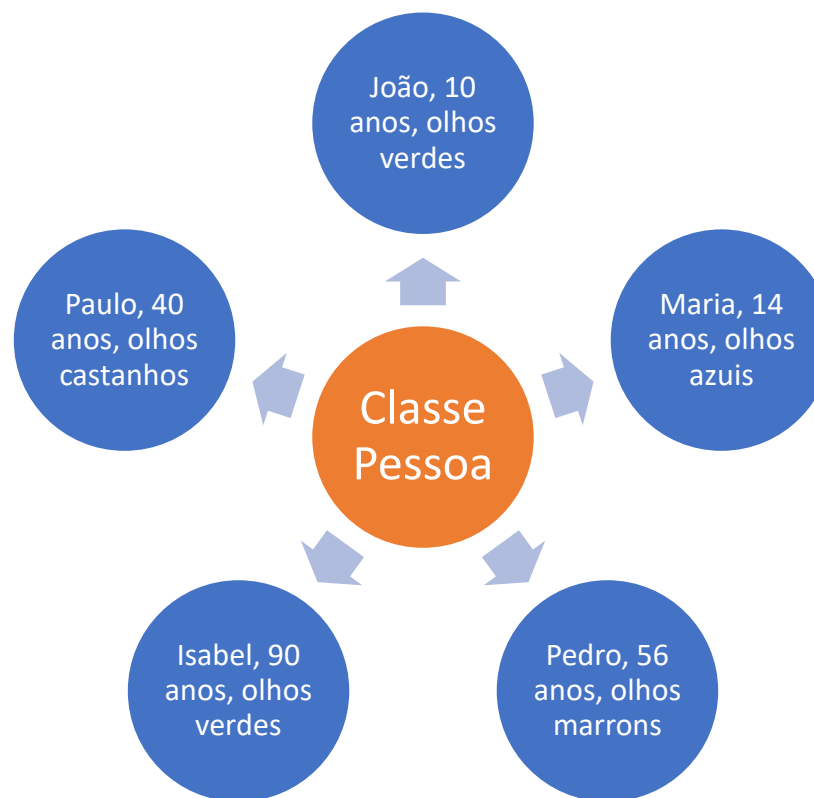
uma **classe** é um modelo ou estrutura que define quais características (**atributos**) e quais ações (**métodos**) um conjunto de objetos terá



Classes versus Objetos

- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores criam classes**, as **classes instanciam objetos**

uma **classe** é um modelo ou estrutura que define quais características (**atributos**) e quais ações (**métodos**) um conjunto de objetos terá



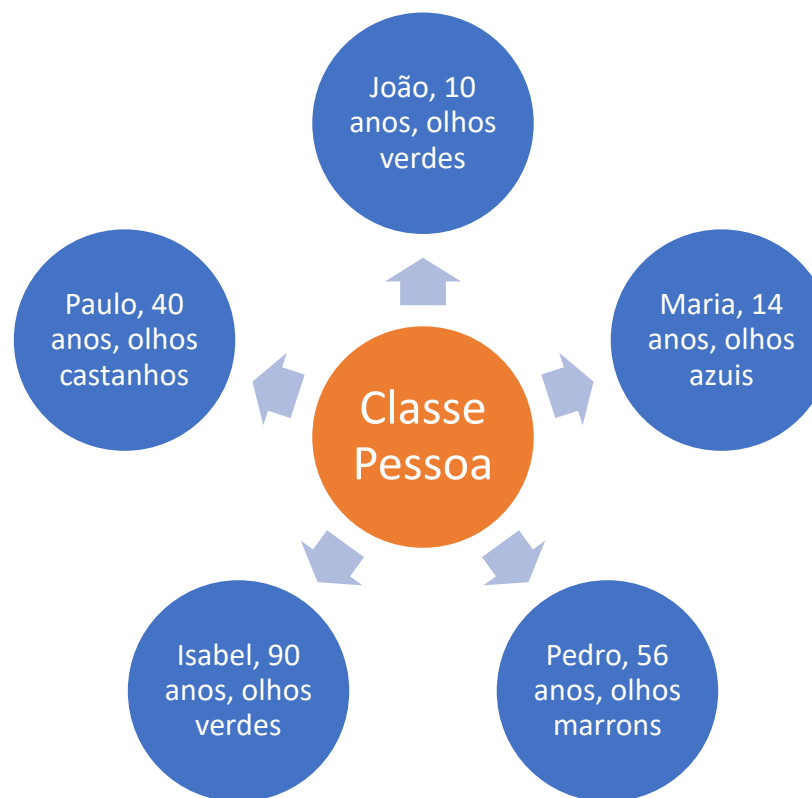
Em **Java**, um **objeto** é uma **instância** de uma **classe**.

Quando criamos um objeto, estamos gerando uma cópia específica daquele modelo (classe).

Classes versus Objetos

- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores criam classes**, as **classes instanciam objetos**

uma **classe** é um modelo ou estrutura que define quais características (**atributos**) e quais ações (**métodos**) um conjunto de objetos terá



Em **Java**, um **objeto** é uma **instância** de uma **classe**.

Quando criamos um objeto, estamos gerando uma cópia específica daquele modelo (classe).

Instanciar um objeto significa criar um novo objeto baseado no modelo da classe, atribuindo a ele valores específicos.

Classes

- A classe descreve as características e comportamento de um conjunto de objetos

Classes

- A classe descreve as características e comportamento de um conjunto de objetos
 - Em Java, **cada objeto pertence a uma única classe**

Classes

- A classe descreve as características e comportamento de um conjunto de objetos
 - Em Java, **cada objeto pertence a uma única classe**
 - O objeto possuirá os atributos e métodos definidos na classe

Classes

- A classe descreve as características e comportamento de um conjunto de objetos
 - Em Java, **cada objeto pertence a uma única classe**
 - O objeto possuirá os atributos e métodos definidos na classe
 - O objeto é chamado de instância de sua classe

Classes

- A classe descreve as características e comportamento de um conjunto de objetos
 - Em Java, **cada objeto pertence a uma única classe**
 - O objeto possuirá os atributos e métodos definidos na classe
 - O objeto é chamado de instância de sua classe
 - A classe é o bloco básico para a construção de programas OO

Exemplo de Classe em Java

```
public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
```

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

```

```

    public void acelera() {
        velocidade++;
    }

```

```

    public void freia() {
        velocidade--;
    }
}

```

Definição da classe

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
  
```

Definição da classe

*A palavra-chave **class** define que estamos criando uma classe chamada **Carro**.*

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
    
```

Definição da classe

*A palavra-chave **class** define que estamos criando uma classe chamada **Carro**.*

*Essa classe servirá como modelo para criar objetos do tipo **Carro**.*

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
  
```

Atributo (características)

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
  
```

Atributo (características)

*A variável **velocidade** representa um atributo da classe*

Exemplo de Classe em Java

```

public class Carro {
    private int velocidade;

    public void acelera() {
        velocidade++;
    }

    public void freia() {
        velocidade--;
    }
}
  
```

Atributo (características)

*A variável **velocidade** representa um atributo da classe*

***private** significa que esse atributo só pode ser acessado dentro da própria classe*

Exemplo de Classe em Java

```
public class Carro {  
    private int velocidade;
```

```
    public void acelera() {  
        velocidade++;  
    }
```

```
    public void freia() {  
        velocidade--;  
    }
```

```
}
```

Métodos (comportamentos)



Exemplo de Classe em Java

```
public class Carro {
    private int velocidade;
```

```
    public void acelera() {
        velocidade++;
    }
```

```
    public void freia() {
        velocidade--;
    }
}
```

Métodos (comportamentos)

Definindo o comportamento de aumentar ou diminuir a velocidade

2

Módulo 2: Exercício 1

Módulo 2

Exercício

Execução do Programa Pessoa + CtrlPrograma (Módulo 2)

Este exercício tem como objetivo praticar a organização e execução de programas em Java no Eclipse, entendendo a relação entre pacotes, classes e o método main.

Você receberá 2 arquivos .java:

- Pessoa.java (classe de modelo – pacote model)
- CtrlPrograma.java (classe controladora – pacote controller, contém o main)

Exercício

Execução do Programa Pessoa + CtrlPrograma (Módulo 2)

Passo a passo obrigatório

1. Criar uma nova área de trabalho (Workspace)
 - Ao abrir o Eclipse, escolha uma nova pasta para o workspace.
 - Sugestão: C:\WrkModulo02

2. Criar um novo Projeto Java
 - Menu: [File] [New] [Java Project]
 - Nome: prjModulo02
 - Desmarque **Create module-info.java** se aparecer.
 - Clique em Finish.

Exercício

Execução do Programa Pessoa + CtrlPrograma (Módulo 2)

Passo a passo obrigatório

1. Criar uma nova área de trabalho (Workspace)
 - Ao abrir o Eclipse, escolha uma nova pasta para o workspace.
 - Sugestão: C:\WrkModulo02

2. Criar um novo Projeto Java
 - Menu: [File] [New] [Java Project]
 - Nome: prjModulo02
 - Desmarque **Create module-info.java** se aparecer.
 - Clique em Finish.

Exercício

Execução do Programa Pessoa + CtrlPrograma (Módulo 2)

3. Adicionar as classes

- No pacote model, crie a classe Pessoa e cole o conteúdo do arquivo Pessoa.java fornecido.
- No pacote controller, crie a classe CtrlPrograma e cole o conteúdo do arquivo CtrlPrograma.java fornecido.

4. Conferir o código

- No topo de Pessoa.java, deve estar: package model;
- No topo de CtrlPrograma.java, deve estar: package controller; import model.Pessoa;

5. Executar o programa

- Abra CtrlPrograma.java.
- Clique no botão verde Run.
- O programa deve compilar e rodar sem erros, mostrando a saída no console.

Exercício

Execução do Programa Pessoa + CtrlPrograma (Módulo 2)

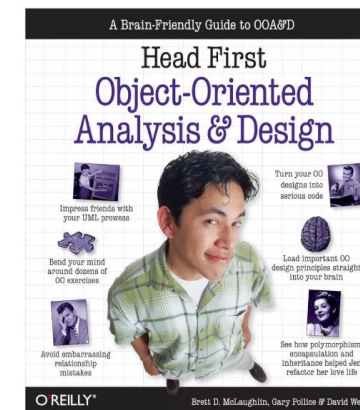
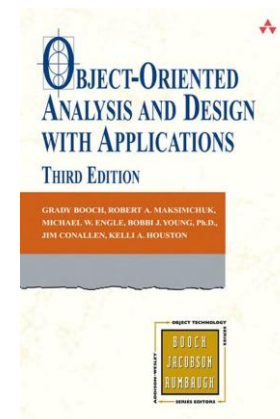
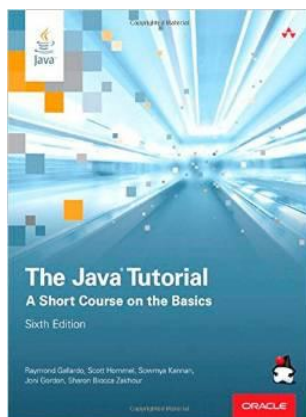
6. Entrega

Resumo com suas palavras explicando o que o programa faz, e

Print da tela do Eclipse com:

- O código de CtrlPrograma.java aberto, mostrando a estrutura dos pacotes
- O console mostrando a execução sem erros e a saída gerada pelo programa

Bibliografia



<http://docs.oracle.com/javase/tutorial>