



Lista 05

Professor: Edwar Saliba Júnior

Usando os componentes: **JLabel**, **TextField**, **Button**, **CheckBox**, **RadioButton**, **ComboBox**, **List**, **TextArea** e etc., resolva os exercícios a seguir. Faça um estudo sobre os **JTable** e também sobre **TableModel** (DefaultTableModel e TableModel Personalizado), pois ajudará na resolução dos problemas.

1. Crie um *software* para controle de venda de Veículos. Este *software* deverá possuir as seguintes classes (Figura 1):
 - a) Veículo, atributos:
 - i. marca (*String*),
 - ii. modelo (*String*),
 - iii. anoFabricacao (*int*) e
 - iv. anoModelo (*int*) e
 - v. preço (*double*);
 - b) Carro (que herda de Veículo), atributo:
 - i. quantidadePortas (*int*);
 - c) Caminhão (que também herda de Veículo), atributo:
 - i. capacidadeCarga (*float*);
 - d) Ônibus (que também herda de Veículo), atributos:
 - i. quantidadePassageiros (*int*) e
 - ii. quantidadeEixos (*int*);
 - e) Cliente, atributos:
 - i. nome (*String*),
 - ii. telefone (*String*) e
 - iii. cpf (*String*);
 - f) Venda, atributos:
 - i. veículo (Veículo),
 - ii. cliente (Cliente),
 - iii. data (*LocalDate*) e
 - iv. preço (*double*).

Crie um CRUD¹ para cadastro dos carros de todas as classes/objetos em classes “Fichário / Gerencia” diferentes. Porém deverá haver um único ArrayList para cadastro de objetos do tipo Veículo, e este deverá ser gerenciado com o uso de polimorfismo. Por fim, crie interfaces intuitivas e faça uso do componente JTable.

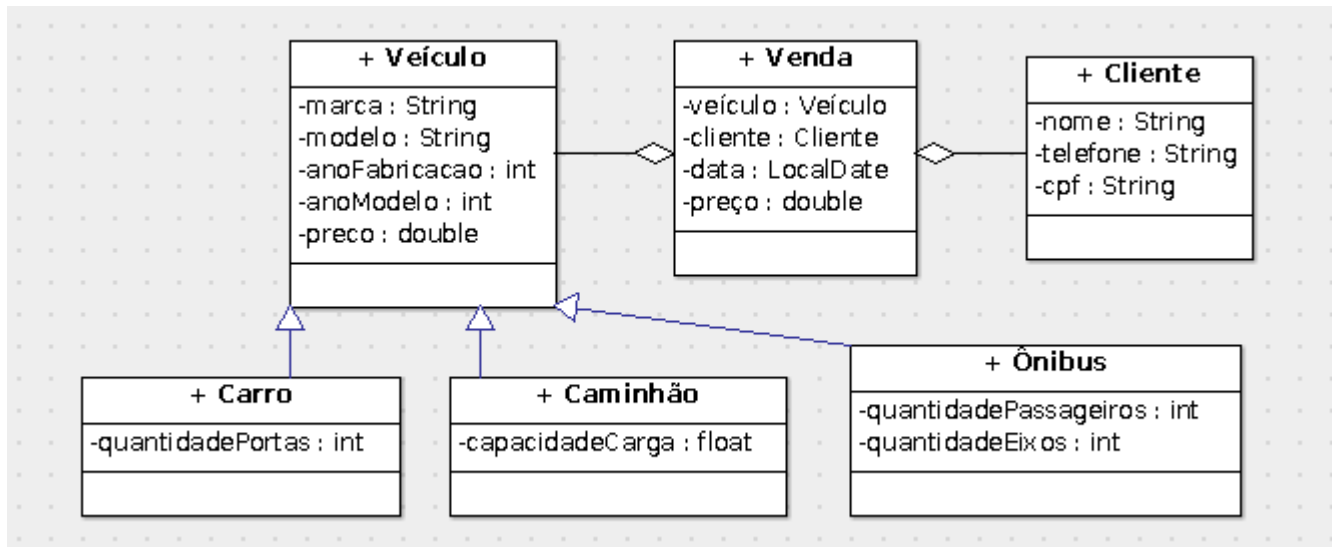


Figura 1: Sistema de venda de veículos.

2. Em um Sistema de Gestão de Consultas, considere as classes:

- a) Pessoa, atributos:
 - i. nome (String),
 - ii. identidade (String),
 - iii. cpf (String) e
 - iv. telefone (String);
- b) Médico, atributos:
 - i. especialidade (String),
 - ii. crm (String) e
 - iii. ctps (int);
- c) Paciente, atributos:
 - i. nomeConvenio (String),
 - ii. numeroConvenio (int) e
 - iii. endereço[2] (Endereço);
- d) Endereço, atributos:
 - i. logradouro (String),
 - ii. número (int),

¹ CRUD – são as 4 operações básicas de um banco de dados (CREATE, RETRIEVE, UPDATE e DELETE), porém aqui se refere as operações de: inclusão, exclusão, alteração, consulta e relatório dos objetos que serão armazenados em um objeto do tipo Collection de Java.

- iii. complemento (String),
- iv. bairro (String),
- v. município (String),
- vi. cep (String) e
- vii.tipo (int); // Residencial ou comercial.

e) Consulta, atributos:

- i. médico (Médico),
- ii. paciente (Paciente),
- iii. data (LocalDate) e
- iv. hora (LocalTime).

Crie um CRUD para cadastro dos médicos, pacientes e seus respectivos endereços e consultas. Todas as classes/objetos deverão ficar em ArrayList's separados, com exceção dos objeto das classes Médico e Paciente que deverão ser armazenados e manipulados num único ArrayList, para tanto faça uso do recurso chamado de polimorfismo. Faça uso da classe de gerenciamento de objetos (Fichário). Por fim, crie interfaces intuitivas e faça uso do componente `JTable`.

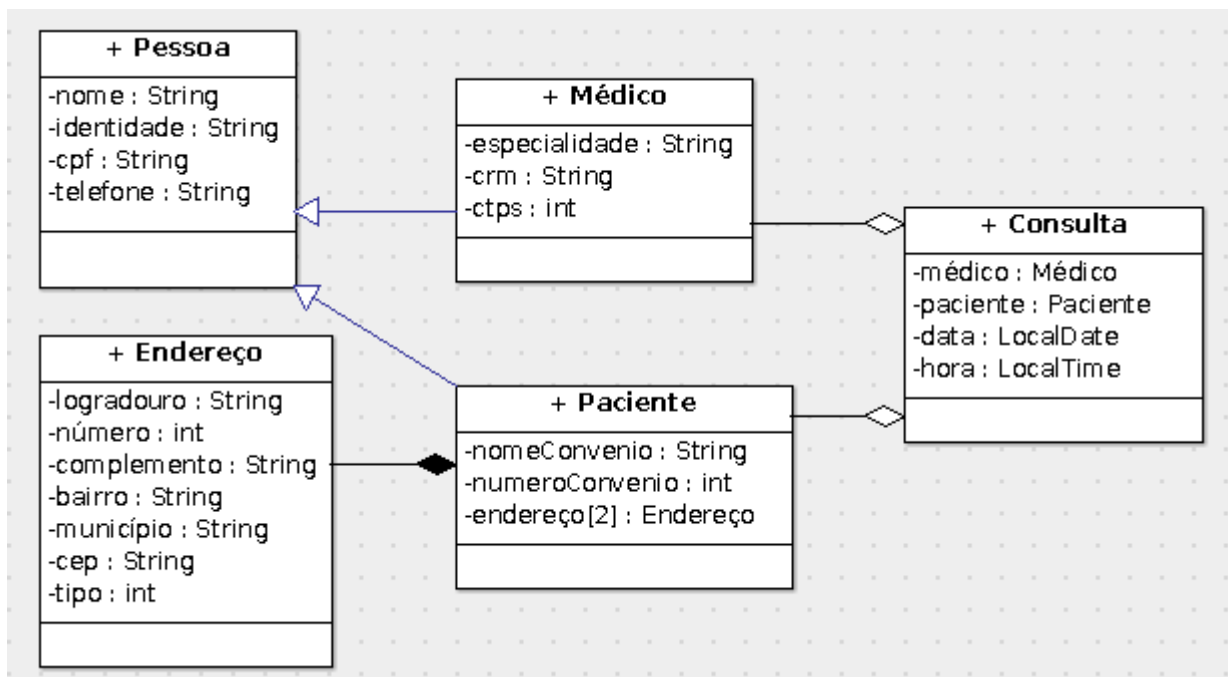


Figura 2: Sistema de gestão de consultas.