



INSTITUTO FEDERAL
Triângulo Mineiro
Campus Paracatu

Programação Visual

Exercícios

(componentes, herança, *collections*, eventos)

Lista 04

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 Veículos. Este *software* deverá possuir as seguintes classes: Veículo, com os seguintes atributos: marca (*String*), modelo (*String*), anoFabricacao (*int*) e anoModelo (*int*) e preço (*double*). Deverá ter também as classes: Carro (que herda de Veículo), com seguintes atributos: quantidadePortas (*int*) e a classe Caminhão (que também herda de Veículo) com o seguinte atributo capacidadeDeCarga (*float*). Crie um CRUD¹ para cadastro dos carros e caminhões em *ArrayList*'s separados. Faça uso da classe de gerenciamento de objetos (Fichário). Crie as interfaces de maneira que fiquem intuitivas para o usuário do *software*. Crie telas onde o usuário possa visualizar todos os objetos cadastrados por meio do componente *JTable*.
2. Em um Sistema de Gestão Hospitalar, considere as classes Médico e Paciente. As classes possuem características em comum, tais como: CPF, identidade, nome, endereço e telefone. As seguintes operações são comuns a ambas classes: cadastro, alteração e recuperação de dados. Entretanto, Médico e Paciente têm características diferentes. Por exemplo, Médico possui CRM e Especialidade, e paciente está associado a um convênio (Figura 1). Capriche nas interfaces e faça uso do *JTable*.

Implemente as classes do modelo proposto.

Na classe principal:

- crie uma estrutura de *menus* capaz de dar suporte, as operações de cadastro, alteração, exclusão e consulta, para toda esta estrutura que você criou;
- seu *software* deverá ser capaz de manipular quantos registros o usuário desejar cadastrar, para isto use *ArrayList* e
- não se esqueça de testar seu *software* para certificar-se de que tudo está funcionando.

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.

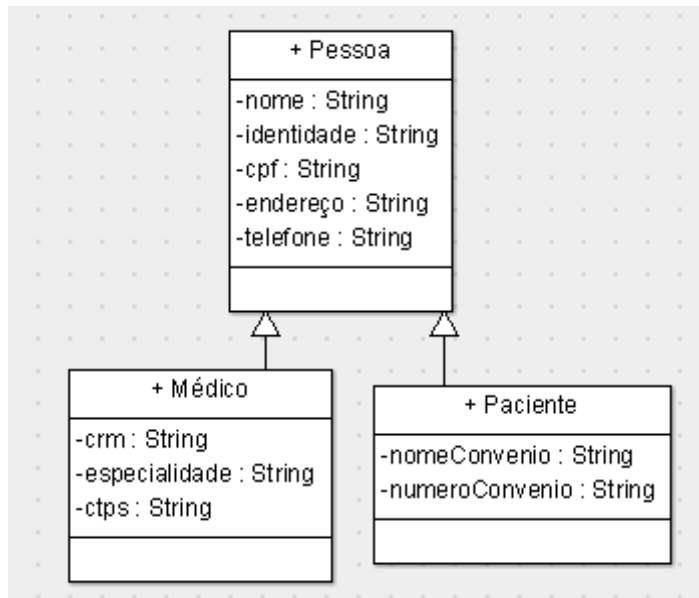


Figura 1: Gestão Hospitalar - Herança

3. Necessita-se de *software* para controle de funcionários e clientes. Este *software* deverá possuir as seguintes classes: Pessoa, com os seguintes atributos: cpf (*String*), nome (*String*), identidade (*String*). Funcionário (que herda de Pessoa), com seguintes atributos: salario (*float*), e ctps² (*String*). Também deverá possuir a classe Cliente (que herda de Pessoa), com os seguintes atributos: limiteDeCredito (*float*), valorDaCompra (*float*), dataDaCompra (*LocalDate*), dataLimiteParaPagamento (*LocalDate*), dataDePagamento (*LocalDate*). Crie um CRUD para cadastro dos clientes e funcionários em *ArrayList*'s separados. Faça uso da classe de gerenciamento de objetos (Fichário). Na classe Cliente, além dos métodos get's e set's, crie também um método "calculaJuros" terá o tipo de retorno *float*. Este método deverá fazer o seguinte cálculo e retorná-lo:

$$valorFinal = ((\text{"data de pagamento"} - \text{"data da compra"}) * 1.03) * valorDaCompra;$$

Nos menus que você criar para o CRUD de cliente, coloque também uma opção para a impressão do valor final a ser pago pelo cliente baseado nas informações contidas no cadastro do mesmo. Ao imprimir o valor final a ser pago, imprima também o "valor da compra", a "data da compra", a "data limite para pagamento" e a "data de pagamento" para que se possa averiguar se o cálculo ficou correto. Crie as interfaces de maneira que fiquem intuitivas para o usuário do *software* e faça uso do componente *JTable*.