



Professor: Edwar Saliba Júnior

Exercício de Codificação:

1. Construa um diagrama de classes para um *software* de controle de Veículos. Este *software* deverá ser capaz de armazenar, em memória principal, os seguintes tipos de veículos: moto, carro, caminhão e ônibus. Faça uso do conceito de herança utilizando o conceito de classe abstrata. Crie um CRUD¹, também conhecido como classe “gerencia” ou “fichário” para o *software* em questão. Utilize interface para padronizar os principais métodos do CRUD. Seu diagrama deverá conter todas as classes do sistema, inclusive a classe Principal.
2. Crie um diagrama de classe para um *software* de controle de funcionários e clientes. Faça uso do mecanismo de herança, utilizando o conceito de classe abstrata. Os atributos de Cliente são os seguintes atributos: limiteDeCredito (*double*), valorDaCompra (*double*), dataDaCompra (*LocalDate*), dataLimiteParaPagamento (*LocalDate*), dataDePagamento (*LocalDate*). Crie CRUD’s para os diversos cadastros, padronizando os métodos principais através do uso de interfaces. Construa também um módulo de compras onde seja possível o cliente comprar a vista ou a prazo e também pagar parcelado e de diversas formas, inclusive cartão de crédito. Dê a possibilidade de se determinar a quantidade de parcelas e também a aplicação de juros no caso de atraso no pagamento. Seu diagrama deverá conter todas as classes do sistema, inclusive a classe Principal. Sempre que for possível faça uso do conceito de herança, para reduzir a implementação de código-fonte e também de interfaces, para padronização dos métodos utilizados.
3. Construa um diagrama de classes para um *software* de estacionamento. Neste estacionamento só serão admitidos 2 tipos de veículos: motos e carros. Este *software* deverá ser capaz de armazenar, em memória principal, uma determinada quantidade de veículos, de cada tipo, por pátio de estacionamento. O *software* também deverá ser capaz de controlar mais de um pátio de estacionamento de forma independente. E, por fim, seu *software* também deverá ser capaz de gerenciar as diárias pagas pelos veículos que usarem os estacionamentos (sendo que a única forma de cobrança será diária). Porém, os pátios poderão ter preços de diárias diferentes. Faça uso do conceito de herança utilizando o conceito de classe abstrata onde for possível. Crie um CRUD, também conhecido como classe “gerencia” ou “fichário” para o *software* em questão. Utilize interface para padronizar os principais métodos do CRUD. Seu diagrama deverá conter todas as classes do sistema, inclusive a classe Principal.

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.