

LISTA DE EXERCÍCIO – SEMANA 05

Notas: - Pode ser utilizado como ajuda os livros contidos na bibliografia da disciplina.
--

1) Use o conceito de classe abstrata para implementar o programa abaixo:

- Crie uma classe abstrata chamada **CartaoWeb**. Essa classe representa todos os tipos de cartões web contendo apenas um atributo: **destinatario** (String). Nessa classe você deverá também declarar o método abstrato *retornarMensagem(String remetente)*.
- Crie três classes filhas (sub-classes) da classe **CartaoWeb**:
 - a) **CartaoDiaDosNamorados**
 - b) **CartaoNatal**
 - c) **CartaoAniversario**.
- Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão. Cada classe também deve implementar o método *retornarMensagem(String remetente)*, retornando uma mensagem ao usuário com o nome do destinatário, um texto que seja específico para a data comemorativa do cartão, acrescido do remetente ao final da mensagem. Por exemplo, essa poderia ser uma mensagem de um cartão de dia dos namorados:

**“Querida Maria,
Feliz Dia dos Namorados!
Espero que esse tenha sido o único cartão do dia dos namorados que tenha ganhado nessa data!”
De todo meu coração,
João”**
- No método main da classe Principal, crie um único array de CartaoWeb. Insira de forma alternada, instâncias dos três tipos de cartões neste array. Em seguida, use um laço de repetição para exibir as mensagens dos cartões, chamando o método *retornarMensagem(String remetente)* dos objetos, utilizando como argumento o seu nome para remetente. Utilize os métodos da classe Scanner para interação com o usuário.

- 2) Uma companhia aérea deseja desenvolver um sistema para gerenciar o embarque de passageiros. Todos os passageiros possuem atributos comuns, como nome e número do voo. Durante o embarque, existem ações obrigatórias, como fazer check-in e passar pelo controle de segurança. Passageiros da classe econômica devem despachar uma mala pequena, enquanto passageiros da classe executiva podem ter privilégios como acesso à sala VIP e prioridade de embarque. Você deve modelar esse sistema em Java usando pelo menos uma classe abstrata e pelo menos uma interface, mas você deve decidir quais métodos e atributos colocar em cada uma. Caso a interface ou classe abstrata fique sem método, você deve sugerir um.
- Utilize o diagrama de classes da UML para modelar seu sistema (não é necessário incluir métodos get e set, construtor).
 - Implemente sua solução em Java.
- 3) Uma academia oferece as seguintes modalidades esportivas para seus clientes: *fitness* e lutas. Cada modalidade possui seus objetivos e regras. A modalidade *fitness* engloba as atividades de *spinning* e musculação e tem como objetivo a melhora do condicionamento físico do indivíduo. Uma regra comum nesse tipo de modalidade é o princípio da progressão, ou seja, aumentar gradualmente a intensidade e a complexidade do treinamento à medida que o condicionamento físico da pessoa melhora. Além disso, cada modalidade *fitness* possui seu tempo médio de duração (horas e minutos). Na modalidade de musculação é preciso definir a quantidade de peso nos exercícios, já na de *spinning* é preciso definir o nível de resistência da bicicleta (carga da bike). A modalidade de lutas, por sua vez, engloba as atividades de boxe e *muay thai* que possuem características similares e tem como objetivo a competição ou autodefesa. Uma regra comum nesse tipo de modalidade é o respeito e a segurança dos participantes. Cada modalidade de luta possui seus equipamentos (ex.: saco de pancada), vestuários (ex.: luvas, protetores bucais e de virilha, etc) e estratégias de luta.
- Utilize o diagrama de classes da UML para projetar um sistema de gerenciamento de modalidades esportivas de uma academia.
 - Quais conceitos da orientação a objetos foram usados na representação UML?
 - Adicione um método para as modalidades esportivas *fitness* e lutas.
 - Adicione na representação UML uma nova modalidade esportiva e um atributo.