

EXERCÍCIO INCREMENTAL – ETAPA 2

1)

Relembrando que a finalidade do exercício incremental deste semestre, cujas etapas vão sendo definidas e publicadas no *Moodle* periodicamente, é criar ao final um pequeno aplicativo para controle de movimento para consumo em bares, restaurantes e similares. A ideia é que esse aplicativo seja desenvolvido por etapas, de acordo com a evolução do conhecimento do aluno em POO.

IMPORTANTE: Todos os atributos das classes descritas devem ser criados como *private* e todos eles devem possuir métodos acessores (*get* e *set*).

Para esta segunda etapa você deve aproveitar as **classes construídas na primeira etapa** e fazer o seguinte.

- Adicionar uma classe **Bar**, que deverá conter como atributo ESTÁTICO uma coleção de objetos da classe *Movimento*. Crie métodos ESTÁTICOS públicos para adicionar movimento, tendo como parâmetro apenas o objeto da classe *Movimento* que se deseja adicionar, e também para remover movimento, tendo como parâmetro o número da mesa vinculada ao movimento. **ATENÇÃO:** Não é permitida a criação de objetos de *Movimento* nesta coleção de objetos com um número de mesa já existente ou, caso já exista, com o status diferente de 9 (encerrada). Ou seja, para criar é preciso ter certeza que não existe nenhum movimento para àquela mesa e, se existir, já esteja encerrado.
- Esta mesma classe Bar deverá conter também como atributo ESTÁTICO uma coleção de objetos da classe *Produto*. Crie métodos ESTÁTICOS públicos para adicionar produtos, tendo como parâmetro apenas o objeto da classe *Produto* que se deseja adicionar, e também para remover produtos, tendo como parâmetro apenas seu código.
- Crie uma classe **Cliente** que deverá conter como atributo os campos *ID* (int), *nome* (String), *tipo* (int: 0 – PF ou 1 – PJ) e *categoria* (int: 0 – NORMAL, 1 – ESPECIAL ou 2 – VIP).
- Crie também uma classe **Especial**, herdeira de *Cliente*, destinado a informações complementares para clientes da categoria especial. Esta classe deverá ter os campos adicionais *endereco* (String), *telefone* (String), *email* (String) e *sexo* (char: M – Masculino e F – Feminino).
- Crie adicionalmente uma classe **VIP**, herdeira de *Especial*, destinado a informações complementares para clientes da categoria VIP. Esta classe deverá ter os campos adicionais *limite* (double) e *gerente* (String).
- Adicione à classe *Movimento* uma coleção de objetos de *Cliente*, juntamente com os métodos de adição e remoção de clientes para um determinado movimento. Para adicionar cliente ao movimento, passar como parâmetro o objeto cliente. Para remover cliente do movimento, passar apenas seu ID.
- Todas as classes criadas até então devem ser colocadas no pacote **bar**.

ATENÇÃO: Os nomes dos atributos e classes devem ser obedecidos conforme descritos e deve ser utilizada a convenção de nomenclatura da comunidade Java, conforme descrito no exercício anterior.

2)

Para testar se as classes deste exercício estão ok, faça um método *main* em outra classe (em outro pacote que não o pacote *bar*) que crie até 5 movimentos para a classe *Bar*. Para cada movimento adicione pelo menos um objeto *Cliente* (pode ser mais de um). Os dados a serem entrados para o cliente vão depender da categoria do cliente. Por exemplo, para cliente normal, apenas ID, nome e tipo. Para cliente especial, além dessas informações, pedir também endereço, telefone e e-mail, e assim por diante.

Todos os dados deverão ser fornecidos pelo teclado e capturado com a classe *java.util.Scanner*. Para abertura dos movimentos, considere a data de abertura como sendo a data atual.

Ao final da alimentação de dados, o programa deverá exibir os dados de cada movimento, incluindo a listagem de todos os clientes por movimento, com todas as informações recebidas do cliente (de acordo com sua categoria).