LP1A3 – Exercícios

Aula 5 – POO - Composição e Herança

Instruções para entrega das listas de exercícios:

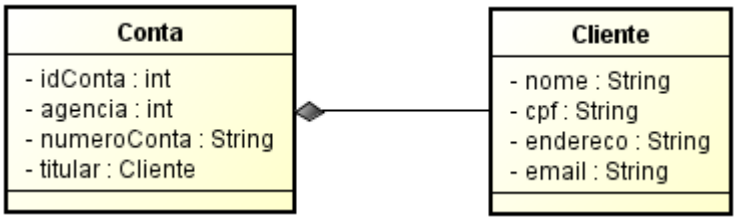
Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (http://eadcampus.spo.ifsp.edu.br).

Forma de Entrega: Para exercícios com uma única classe, deve ser entregue o arquivo da classe (extensão JAVA) referente ao exercício. Por exemplo: Exercicio3.java. Para exercícios com mais de uma classe, cada exercício deve ter uma pasta, na qual serão colocados os arquivos JAVA referentes ao exercício. Por exemplo: Para o Exercício 4, deve existir uma pasta "Exercicio4" contendo todos os arquivos JAVA deste exercício. **Entregue apenas os arquivos JAVA**. Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaoSilva.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. Faça um programa console em Java que implemente o seguinte relacionamento entre as classes Conta e Cliente. Use os métodos **set** e **get** para receber e imprimir os dados (não é necessário ler os dados do usuário). Por fim, faça testes para verificar interdependência entre as duas classes.



2. Faça um programa console em Java que implemente um relacionamento de **composição** entre duas classes. Use os métodos **set** e **get** para receber e imprimir os dados (não é necessário ler os dados do usuário). Lembre-se que, na composição, a existência do objeto-parte só faz sentido se ele estiver associado a um objeto-principal.

3. **(Atividade assíncrona referente ao dia 22/08/2020)** Faça um programa em Java que apresente ao usuário as opções a seguir, enquanto ele não digitar a opção 0 (zero). De acordo com o número da opção informada, o programa deverá efetuar a operação, solicitando as informações ao usuário quando necessário. Os dados dos clientes devem ser armazenados em dois vetores de objetos, um do tipo PessoaFisica[10], que armazenará objetos contendo código, endereço, telefone, nome, CPF dos clientes, e um vetor do tipo PessoaJuridica[10], que armazenará objetos contendo código, endereço, telefone, razaoSocial e CNPJ dos clientes. Seguem as opções:
    1) **Inserir cliente** – Primeiramente, pergunte ao usuário se o cliente é pessoa física (PF) ou jurídica (PJ). Para pessoa física, solicite o nome, o CPF, o endereço e o telefone. Para pessoa jurídica, solicite a razão social, o CNPJ, o endereço e o telefone. O código do cliente não deve ser informado pelo usuário, deve ser gerado automaticamente a partir do número 1. Além disso, a sequência de códigos deve ser única para pessoas físicas e jurídicas, ou seja, não devem existir clientes PF e PJ com um mesmo código. Informe ao usuário se houve sucesso ou não na inserção.

2) **Remover cliente** – Primeiramente, pergunte ao usuário se o cliente é pessoa física (PF) ou jurídica (PJ). Solicite então o código do cliente. Na remoção, deve ser atribuído o valor "null" à posição do vetor relacionada ao cliente informado. Informe ao usuário se houve sucesso ou não na remoção (por exemplo: código inexistente).
3) **Consultar clientes** – A consulta deve apresentar ao usuário os dados de todos os clientes cadastrados, separados em pessoas físicas e pessoas jurídicas.

A aplicação deve possuir as seguintes classes:

- **Cliente –** Classe que contém os atributos privados int codigo, String endereco e String telefone, além dos métodos protegidos **get** e **set** necessários.
- **PessoaFisica –** Classe filha da classe **Cliente**. Contém os atributos privados String nome e String cpf, além dos métodos públicos **get** e **set** necessários.
- **PessoaJuridica –** Classe filha da classe **Cliente**. Contém os atributos privados String razaoSocial e String cnpj, além dos métodos públicos **get** e **set** necessários.
- **Inicio –** Classe que contém o método **Main**, onde são instanciadas as classes **PessoaFisica** e **PessoaJuridica**. Na opção 1, devem ser usados os métodos **set** para criar um objeto contendo os dados da pessoa física ou da pessoa jurídica, dependendo do tipo de cliente informado. Este objeto deve ser inserido então no vetor de pessoas físicas ou pessoas jurídicas. Por fim, o código deve ser incrementado para inserção do próximo cliente. Na opção 2, deve-se usar o método **getCodigo** para percorrer os vetores de pessoas físicas ou pessoas jurídicas e buscar o cliente que tem o código informado para remoção. Na opção 3, devem ser chamados os métodos **get** para recuperar os dados de cada objeto armazenado nos vetores e apresentá-los ao usuário.