Segunda lista de exercícios
Herança, Classes Abstratas, Polimorfismo

Exercício 1 (Unidade 2 - Semana 8):

Considere a seguinte situação de uma empresa:

- Todos os Funcionários tem como atributos : nome e sobrenome;
- Funcionários podem ser gerentes, administradores, comissionados e horistas;
- Um administrador ganha um salário mensal, sendo um atributo complementar da classe Administrador;
- Um Gerente ganha um salário mensal mais uma bonificação, sendo dois atributo a mais da classe Gerente;
- Comissionados ganham um salário mensal mais uma comissão em cima do total de vendas, tendo como atributos complementares : salario mensal, total de vendas e percentual de comissão;
- Horistas ganham em cima da quantidade de horas trabalhadas tendo como atributos próprios: quantidade de horas trabalhadas e valor da hora

1. Criar as classes necessárias para implementar a situação acima, usando Herança e polimorfismo;
2. Criar os métodos gets e sets para os atributos;
3. Criar métodos para mostrar os dados dos funcionários;
4. Criar métodos em cada classe para mostrar os ganhos mensais de acordo com a forma de pagamento de cada funcionário;

5. Na classe principal:

    - instanciar 10 funcionários sendo 2 horistas, 4 administradores, 3 comissionados e 1 gerente;
    - Armazenar os funcionários em um vetor;
    - Percorrer o vetor mostrando os dados dos funcionários e quanto cada um ganha por mês.

Exercício 2

1 – Criar uma classe Abstrata para Loja com identificador, cnpj, razão social, aberta (boolean).

2 – Criar um construtor da classe Loja para criar o objeto já com cnpj, razão social e o atributo aberta = false.

3 - Criar um método para retornar a razão social, cnpj e o status da loja (se aberta ou fechada)

4 – Criar uma classe concreta para Loja com endereço e gerente.

5 - Criar um método construtor para a Loja concreta recebendo os dados dos atributos, chamando o construtor da classe pai e atualizando os atributos.

6 – Criar um método para mostrar a Loja com todos os dados.

7 - Criar 3 instâncias da Loja (1 Matriz e 2 filiais), cada uma com seu cnpj, razão social, identificador, endereço e gerente. Mostrar as lojas criadas, com todos os dados.

8 – Criar uma interface Registro com os métodos registra_abertura_dia e registra_fechamento_dia.

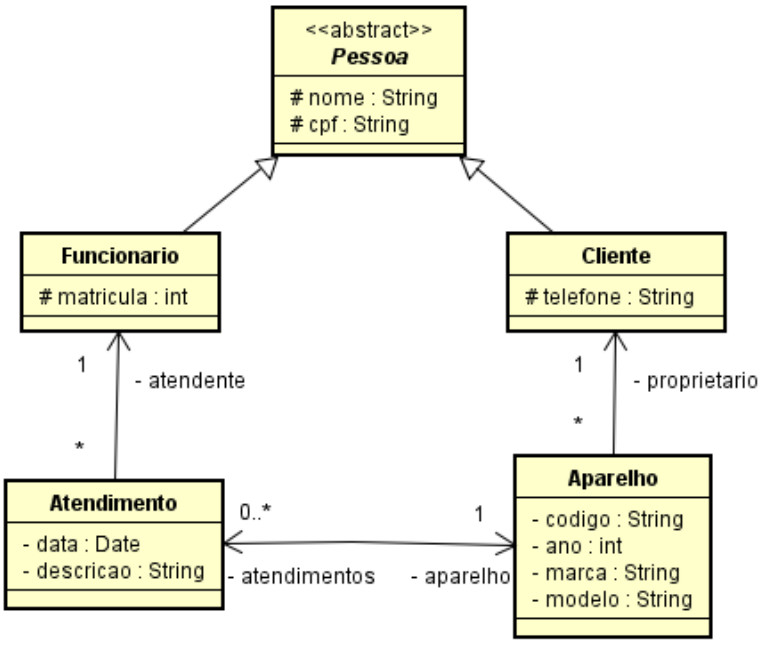
9 – Fazer a Loja utilizar a interface Registro e implementar os métodos atualizando o atributo **aberta** para *true* quando chamar o método registra_abertura_dia , ou para false quando invocado o método registra_fechamento_dia. Uma loja aberta não pode ser aberta novamente e uma loja só pode ser fechada se estiver aberta.

10 – Na classe principal, as lojas criadas estarão fechadas (aberta = false). Usar os métodos de Abrir e Fechar mostrando os dados novamente para ver o funcionamento dos métodos.

Exercício 3

Considerando que uma empresa deseja controlar os atendimentos, de assistência técnica, realizados por seus funcionários, mantendo os dados referentes aos aparelhos, seus proprietários e o histórico dos atendimentos de cada aparelho, registrando quando foi realizado, quem realizou e uma breve descrição. Para atender a esses requisitos, foi modelado o diagrama de classes abaixo.

- Implemente as classes, seus atributos e relacionamentos de acordo com a representação do diagrama.
- Crie construtores para todas as classes, atualizando todos os dados (atributos).
- Crie também um método para adicionar um atendimento a um aparelho.
- Na classe principal, o programa deve criar 2 aparelhos, cada um relacionado a um cliente e para cada aparelho, devem ser cadastrados 2 atendimentos distintos realizados por funcionários diferentes.
- Os funcionários e clientes cadastrados devem ser armazenados em uma única lista. Ao final o programa deve percorrer a lista e mostrar todos os clientes e funcionários cadastrados, com seus dados.
- O programa deverá também mostrar (usando o método toString()) os dados dos 2 aparelhos (todos os dados) incluindo os atendimentos que foram realizados e os funcionários que realizaram.



Exercício 4

1 – Utilizando os exemplos dado em sala de aula do padrão de projeto Factory Method, implemente uma fábrica de criação de documentos.

Os documentos têm um número, o remetente e um tipo. Sendo que cada tipo é uma subclasse da classe Documento, que podem ser Cartas, Telegramas, Notificações.

Dependendo do tipo informado para o método de criação, ele deve criar a instância equivalente.

Classes a serem criadas:

- Documento (abstrata)

- Carta (concreta)

- Notificacao (concreta)

- Telegrama (concreta)

- FabricaDocumentos (concreta)

- Fabrica (abstrata)

- CadastroDocumentos (classe principal) – que, dependendo do tipo informado, vai solicitar à fabrica a criação de um documento.