

Terceira Lista de Exercícios

Herança, Polimorfismo, Classes Abstratas e Interfaces

Questão 1 – No GitHub Classroom: <https://classroom.github.com/a/UVnPG3Hk>

Considere a seguinte situação de uma empresa:

- Todos os Funcionários têm, 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 atributos a mais na classe Gerente;
 - Comissionados ganham um salário mensal mais uma comissão em cima do total de vendas, tendo como atributos complementares : salário 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 *get* e *set* 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 as instâncias de funcionários em um vetor;
 - Percorrer o vetor mostrando os dados dos funcionários e quanto cada um ganha por mês.

Questão 2 – No GitHub Classroom: <https://classroom.github.com/a/yD83vQnD>

- 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.