

LISTA L36

PROF. ME. ALEXANDRE GARCIA

1. ORIENTAÇÃO A OBJETOS - PARTE 3

Exercício 1.1. *(Obrigatório) Crie a classe Biblioteca que possui nome e varias Estantes. Uma Estante possui um nome, uma categoria (Ciência, Literatura ou Filosofia) e vários livros. Um livro é composto por nome, autor e ano de publicação. Todas as classes possuirão get's para seus atributos e construtores. Faça os métodos (é requisito de avaliação a correta implementação dos métodos abaixo em sua respectiva classe):*

- (1) `void inserirEstante(Estante estante):` *Insere uma Estante em uma Biblioteca*
- (2) `void inserirLivro(Livro livro):` *Insere um Livro em uma Estante*
- (3) `Livro[] listarFilosofia():` *Retorna uma lista com todos os livros de filosofia*
- (4) `int contarCiencia():` *Retorna a quantidade de livros de Ciência na biblioteca*
- (5) `void listarTudo():` *Mostra na tela todos os livros de todas as estantes*
- (6) `void removerLivro(Livro livro):` *Remove de uma estante o livro em questão.*
- (7) `ArrayList<String> listarAutores(Categoria cat):` *Retorna um lista de autores de acordo com a categoria desejada.*
- (8) *Crie dois métodos que achar necessário que difram dos listados acima.*
- (9) *Crie um menu para o controle da biblioteca.*

Exercício 1.2. *Sabe-se que VIP, Premium e Regular são tipos de Cliente. Em cada cliente, há a necessidade de controle de conta corrente. Em uma conta, há dados como nome, e saldo. Para cada tipo de conta uma tarifa deve ser calculada: 1% do saldo para Regular, 2% para Premium e 4% para VIP. Use Herança. Aqui, fica proibido usar conceitos que não foram vistos em aula.*

- *Esboce um diagrama de classes.*
- *Implemente as classes e seus métodos.*
- *Crie um pequeno menu para cadastrar clientes e verificar o valor das tarifas.*

Exercício 1.3. *Uma Estrela é composta por nome, temperatura e cor. Uma Constelação possui um nome e tem várias Estrelas. Toda Estrela possui um método `mostrar()` que mostra na tela todas as informações sobre ela. Neste sistema, é possível incluir estrelas em uma constelação usando o método `void adicionar(Estrela e)`, mostrar as informações de todas as estrelas com o método `void infoEstrelas()` e exibir a temperatura total somando-se todas as estrelas da constelação usando o método `double tempConstelacao()`. Implemente as classes descritas.*

Exercício 1.4. *(Obrigatório) Em Desenvolvimento Web, autenticação e autorização são conceitos fundamentais. Em um sistema, um Guest, Regular, Group e Root são tipos de Usuário. Um usuário possui uma string que representa seu login. Cada um desses vai possuir métodos de despedida e descrição da permissão (o que cada um deles fazem. Aqui você é livre para criar). Os métodos descritos devem possuir a convenção `acaoUsuario` (por exemplo, `despedidaRegular()`). Um regular terá um método chamado `void dashboard()` (imitação de uma área de usuário cadastrado), o usuário de grupo terá um `void groupBoard()` que avisará que ele poderá apagar usuários de seu grupo (apenas a mensagem). O super usuário terá um método `void excluirTodos()` que avisa que ele poderá excluir todos os usuários (apenas mensagem). Sabe-se que todos esses usuários mostrarão uma mensagem de boas-vindas ao início do sistema. A mensagem será: Bem-vindo <seuloginaqui>. Implemente as classes envolvidas e *esboce um diagrama de classes. Crie um menu de testes a partir da descrição do problema acima.**