**Prova de Programação Orientada a Objeto**

**Prof. Francesco Artur Perrotti – Fatec Americana**

A biblioteca da faculdade decidiu que, além dos alunos, vai atender também outras pessoas, mas com algumas restrições. Para todos os usuários da biblioteca (alunos e não-alunos), o cadastro precisa registrar o código, nome e e-mail do usuário. Para os alunos também é preciso armazenar o RA. Para os que não são alunos é preciso armazenar o RG e o telefone para contato. Além disso, sempre que um usuário devolve um livro com atraso essa informação também fica registrada no cadastro, portanto é preciso ter um contador de atrasos no cadastro. Se o usuário atrasar mais que 3 vezes para não-alunos ou 5 vezes para alunos, fica impedido de retirar livros até que seja liberado pela administração da biblioteca. Quando o usuário é liberado o contador é zerado.

Faça uma hierarquia de classes para representar os usuários, alunos e não alunos e uma classe que manipule uma lista de usuários. Todos os atributos de todas as classes precisam ser privados e as classes devem fornecer os métodos adequados para a manipulação dos atributos.

O construtor da classe com a lista deve receber por parâmetro a capacidade máxima da lista. Além disso a classe precisa conter os seguintes métodos:

**int buscar(int cod)**: Procura o usuário com o código informado e retorna o índice do usuário na lista. Caso não encontre retorna -1.

**void cadastrar(Usuario us)**: adiciona um novo usuário na lista. Lança uma exceção se a lista está lotada ou se já existe um usuário com o mesmo código.

**void retirar(int cod)**: é chamado quando um usuário tenta retirar um livro. O método deve lançar uma exceção VERIFICADA caso o usuário esteja impedido de retirar livros.

**void devolver(int cod, boolean emAtraso)**: é chamado quando o usuário devolve um livro. O parâmetro emAtraso informa se o houve atraso na devolução. Caso o usuário tenha atingido o limite máximo de atrasos este método deve bloquear o usuário para novas retiradas. Lance uma exceção se o código não existe na lista de usuários.

**void liberar(int cod)** libera um usuário impedido de retirar livros. Lance uma exceção se o código não existe na lista de usuários.

**void listarAlunos()** Lista todos os alunos cadastrados.

**void listarUsuarios()** Lista todos os usuários cadastrados. No final da listagem informe quantos usuários estão impedidos por atraso.

**Importante:**

1. **Teste os métodos implementados e capture todas as exceções no método main.**
2. **Não utilize as classes List<>, ArrayList<> ou qualquer classe de manipulação de listas já pronta do java.**
3. **Ponha nome completo e RA em TODOS os arquivos do projeto.**
4. **É PRA PÔR O NOME COMPLETO.**