



**COM220 – Computação Orientada a Objetos I**  
**Trabalho 8**

Nome: \_\_\_\_\_  
Nro Matrícula: \_\_\_\_\_

Considere as classes *Disciplina* e *Estudante*, com os atributos descritos a seguir:

*Estudante*

Atributos: *nroMatric* (int), *nome* (String)

*Disciplina*

Atributos: *codigo* (int), *nome* (String)

Implemente essas classes com construtores contendo dois atributos. Em seguida, utilize o código a seguir para popular dois objetos tipo *Vector* com instâncias de estudantes e disciplinas.

```
ArrayList<Disciplina> listaDisciplina = new ArrayList<>();  
listaDisciplina.add(new Disciplina(110, "Estrutura de Dados"));  
listaDisciplina.add(new Disciplina(120, "Sistemas Operacionais"));  
listaDisciplina.add(new Disciplina(130, "Programação OO"));  
ArrayList<Estudante> listaEstudante = new ArrayList<>();  
listaEstudante.add(new Estudante(1001, "José da Silva"));  
listaEstudante.add(new Estudante(1002, "João de Souza"));  
listaEstudante.add(new Estudante(1003, "Marcelo dos Santos"));  
// inserir mais 7 alunos, totalizando 10 no ArrayList
```

**Questão 1 – Criação de turmas** [valor: 60 pts]

Elabore uma aplicação *Swing* para matrícula de alunos em disciplinas. A interface da aplicação deve conter um *JComboBox*, o qual conterá o código e o nome das disciplinas cadastradas, por exemplo:

110 – Estrutura de Dados

Abaixo do combo box deve haver um *textField* para a digitação do código dos alunos que se deseja matricular. Após a digitação de um número de matrícula, o programa deve verificar se o mesmo é um número de matrícula válido e, se for, deve inseri-lo em um *ArrayList* chamado *listaEstMatric*. Caso o código seja inválido, deve-se dar uma mensagem em um *JOptionPane*. O procedimento de matrícula termina quando o usuário digita uma matrícula igual a 0. Neste momento, deve-se gerar um objeto da classe *Turma*, contendo a instância da disciplina escolhida no combo box e o *ArrayList* *listaEstMatric*. O objeto gerado deve ser armazenado num *ArrayList* denominado *listaTurma*. O código deve salvar o *ArrayList* *listaTurma* num arquivo, de modo que os dados das matrículas não sejam perdidos. Os dados de disciplinas e estudantes não precisam ser salvos, pois o código fornecido irá construir as listas de disciplinas e estudantes toda vez que o programa for executado. Para simplificar a implementação, poderá haver apenas 1 turma para cada disciplina. Assim, as turmas poderão ser identificadas pelo código da disciplina.

Importante: Nesta questão você deve utilizar pelo menos um gerenciador de layout a sua escolha. Não serão consideradas as implementações que utilizarem unicamente *JOptionPane*.

## Questão 2 – Impressão de Turmas [valor: 40 pts]

Implemente uma aplicação Swing que proveja uma janela para digitação de um código de disciplina e a exibição dos alunos matriculados na referida disciplina. A tela deve conter um *JTextField* para a leitura do código da disciplina e uma *JTextArea* para a exibição dos nomes dos alunos matriculados. Após a digitação do código da disciplina, deve-se checar se a mesma existe, em seguida deve-se verificar se existe alguma turma daquela disciplina.

Se o código da disciplina for inexistente, deve-se dar a seguinte mensagem na *JTextArea*: “Este código de disciplina não existe”. Se a disciplina existir, mas não houver turma daquela disciplina, deve-se dar a mensagem: “Não existem turmas dessa disciplina”. Caso exista uma turma da disciplina, deve-se exibir na *JTextArea* o **nome** de todos os alunos matriculados na disciplina.

Sugestão: Você pode criar um *JFrame* contendo um menu com as opções:

- Criação de turmas
- Impressão de turmas

---

Trabalho 08 – versão 2

Para 16/06

Implemente as classes *LimiteEstudante* e *LimiteDisciplina* para permitir a leitura dos dados de estudantes e disciplinas. Faça a persistência de todos os dados lidos em arquivo.