

# Estrutura de Dados – Prof. Luiz Mário L. Pascoal

## Trabalho 01 – Exercícios de Revisão de IP e POO

1. Altere o programa feito em sala de aula que contém uma classe **Aluno** com os seguintes atributos: **nome**, **n° de matricula**, **notaBimestre1**, **notaBimestre2**. Em seguida insira o construtor (que deverá conter os 4 primeiros atributos supracitados), também o construtor vazio, métodos **Getters e Setters**, o método **toString()** para impressão dos dados e por fim o método **calculaMédia()** que deverá retornar a média aritmética entre **notaBimestre1** e **notaBimestre2**.

Em seguida, crie a classe Principal que deverá conter o método **Main**. A classe **Principal** deverá ler os 4 atributos, e para cada leitura completa deverá criar um objeto do tipo **Aluno**, onde ao final estes objetos **Aluno** deverão ser armazenados em um **ArrayList** “listaAlunos” do tipo **Aluno**, que representa uma lista de alunos.

Após a criação da listaAlunos, construa os métodos que implementem as seguintes operações apresentadas via menu para o usuário:

**0. Sair:** Encerra o programa.

**1. Impressão da lista de Alunos Cadastrados:** Este método deverá imprimir todas as informações de cada aluno presente na listaAluno.

**2. Impressão da lista de Alunos Aprovados:** Este método deverá imprimir o nome e a média dos alunos que estão aprovados com média 6.0.

**3. Busca de Aluno pelo nome ou matrícula:** Este método deverá perguntar ao usuário se ele deseja buscar um aluno pelo seu nome ou o seu número de matrícula e em seguida procurar por este aluno dentro do **ArrayList**. Caso não encontre, o sistema deverá mostrar uma mensagem para o usuário.

**4. Imprimir a lista de Alunos Cadastrados ordenados pela média final:** Este método deverá ordenar e mostrar o nome e a média da listaAlunos em ordem decrescente (da maior para a menor) de acordo com a média.

Dica: Percorra este **ArrayList** utilizando um “**for-each**” **for(Aluno aluno : listaAlunos)**

2. Escreva uma classe que represente um país. Um país tem como atributos o seu nome, o nome da capital, sua dimensão em Km2 e uma lista de países com os quais ele faz fronteira. Represente a classe e forneça os seguintes construtores e método:

**a)** Construtor que inicialize o nome, capital e a dimensão do país;

**b)** Métodos de acesso (obter/get) e modificar (set) para as propriedades indicadas no item (a);

c) Um método que permita verificar se dois países são iguais. Dois países são iguais se tiverem o mesmo nome e a mesma capital. A assinatura deste método deve ser:

```
public boolean equals(final Pais outro);
```

d) Um método que define quais outros países fazem fronteira (note que um país não pode fazer fronteira com ele mesmo);

e) Um método que retorne a lista de países que fazem fronteira;

f) Um método que receba um outro país como parâmetro e retorne uma lista de vizinhos comuns aos dois países.

3. Crie uma classe **calculadora**. Esta classe deve ser abstrata e implementar as operações básicas (soma, subtração, divisão e multiplicação). Utilizando o conceito de herança crie uma classe chamada **calculadora científica** que implementa os seguintes cálculos: raiz quadrada e a potência. Dica utilize a classe Math do pacote java.lang.

4. De forma incremental, traduza o seguinte conjunto de classes em um programa Java. Importante: Não são permitidas chamadas a System.in, System.out ou similares de dentro das classes criadas.

a) Classe: Porta Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ Métodos: void abre(), void fecha(), void pinta(String s), boolean estaAberta() Para testar, crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método estaAberta para verificar se ela está aberta.

b) Classe: Casa Atributos: cor, porta1, porta2, porta3 Método: void pinta(String s), int quantasPortasEstaoAbertas(), int totalDePortas() Para testar, crie uma casa e pinte-a. Crie três portas e coloque-as na casa; abra e feche as mesmas como desejar. Utilize o método quantasPortasEstaoAbertas para imprimir o número de portas abertas.

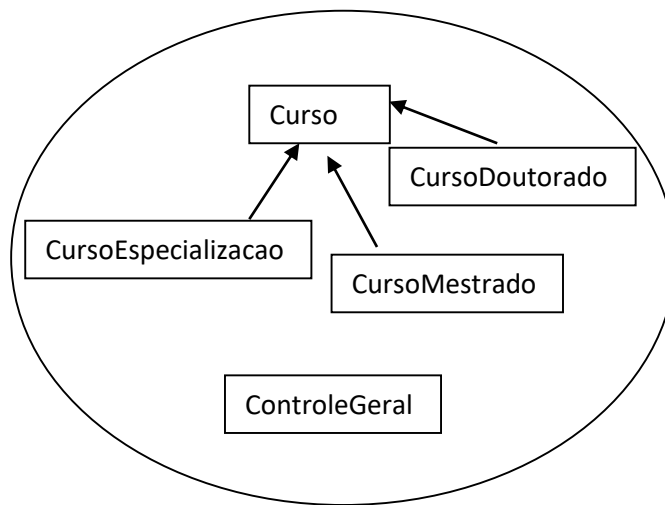
c) Classe: Edificio Atributos: cor, totalDePortas, totalDeAndares, portas[] Métodos: void pinta(String s), int quantasPortasEstaoAbertas(), void adicionaPorta(Porta p), int totalDePortas(), void adicionarAndar(), int totalDeAndares()

Para testar, crie um edificio, pinte-o. Crie seis portas e coloque-as no edificio através do método adicionaPorta, abra e feche-as como desejar. Utilize o método quantasPortasEstaoAbertas para imprimir o número de portas abertas e o método totalDePortas para imprimir o total de portas em seu edificio. Cria alguns andares utilizando o método adicionarAndar e retorne o número total de andares utilizando o método totalDeAndares.

- d) As classes Casa e edifício ficaram muito parecidas. Crie a classe Imovel e coloque nela tudo o Casa e Edificio tem em comum. Faça Imovel superclasse de Casa e Edificio. Note que alguns métodos em comum não poderão ser implementados por Imovel (e.g., quantasPortasEstaoAbertas e totalDePortas). Logo, esses deverão ser declarados como métodos abstratos.

5. **(Com interface gráfica GUI).** Desenvolva um sistema para cadastro de Curso com seus Alunos de acordo com o diagrama abaixo:

Classes envolvidas



Classe Curso:

Atributos: int código, cargaHoraria, sala  
String nome, nomeProfessor;  
double valorCurso;  
int quantidadeAlunos;

Métodos: public Curso(int cod, int ch, int sala, String n, String nP, double v, ArrayList alunos)

Getters and Setters

Classe CursoEspecializacao extends Curso:

Atributos: boolean latoSensu;  
double adicionalDiploma;

Métodos: public CursoEspecializacao (int cod, int ch, int sala, String n, String nP, double v, int qntAlunos, boolean lS)  
public void setAdicionalDiploma(); // método que seta o adicional diploma.

- O adicional diploma deverá ser calculado da seguinte forma:
  - Se a turma tiver até 10 alunos → R\$ 25,00 por aluno
  - Se a turma tiver até 30 alunos → R\$ 20,00 por aluno
  - Acima de 30 alunos → R\$ 18,00 por aluno

Classe CursoMestrado extends Curso:

Atributos: boolean strictoSensu;  
double adicionalDiploma;

Métodos: public CursoMestrado (int cod, int ch, int sala, String n, String nP, double v, int qntAlunos, boolean lS)

public void setAdicionalDiploma(); // método que seta o adicional diploma.

- O adicional diploma deverá ser calculado da seguinte forma:
  - Se a turma tiver até 5 alunos → R\$ 45,00 por aluno
  - Se a turma tiver até 15 alunos → R\$ 43,00 por aluno
  - Se a turma tiver até 30 alunos → R\$ 40,00 por aluno
  - Acima de 30 alunos → R\$ 36,00 por aluno

Classe CursoDoutorado extends Curso:

Atributos: boolean strictoSensu;

int mesesParaDefesa;

double adicionalDiploma;

Métodos: public CursoMestrado (int cod, int ch, int sala, String n, String nP, double v, int qntAlunos, boolean lS)

public void setAdicionalDiploma(); // método que seta o adicional diploma.

- O adicional diploma deverá ser calculado da seguinte forma:
  - Se faltam 24 meses para defesa → R\$ 1500,00
  - Se faltam 12 meses para a defesa → R\$ 2000,00
  - Se faltam 6 meses para a defesa → R\$ 3000,00
  - Caso contrário → R\$ 4000,00.
- Além disso, o adicional diploma terá desconto com reajuste percentual de acordo com a quantidade de alunos presentes no programa da seguinte forma:
  - 7% se a turma tiver até 3 alunos.
  - 12% se turma tiver até 7 alunos
  - 15% se a turma tiver mais de 7 alunos.

Classe Controle Geral:

Responsável em ler todas as informações para o cadastramento dos CursosEspecializacao CursoMestrado e CursoDoutorado e crie um ArrayList para cada tipo de Curso, colocando em um ArrayList do tipo de cada classe implementada. Obs. Pelo menos 2 objetos de cada tipo (Utilize o **showInputDialog** para ler os atributos normais e o **showInputDialog com opções** para selecionar o tipo de Curso “Especializacao” “Mestrado” “Doutorado” que está sendo cadastrado. Repita este processo enquanto o usuário desejar continuar, controle isso usando o **showConfirmDialog**. Ao final, apresente uma caixa de diálogo usando o **showOptionDialog** usando as opções de escolha definidas acima para escolher qual dos ArrayLists será impresso pelo **showMessageDialog** contendo todos os dados armazenados naquele ArrayList.

6. Dada a classe Interface FormaGeometrica abaixo, desenvolva o código Java das classes: Círculo, Quadrado e Retângulo que implementam os métodos de FormaGeometrica de acordo com as regras de interface.

```
public interface FiguraGeometrica
{
    public double getArea();
}
```

```
        public double getPerimetro();  
  
        public void exibirDados(); //método que mostra todas  
as informações da classe, com a área e perímetro calculados.  
    }
```

Sabendo que:

- **Quadrado** =>  $\text{Area} = \text{lado} * \text{lado}$ ;  $\text{Perimetro} = \text{lado} * 4$ ;
- **Círculo** =>  $\text{Area} = (\text{PI} * \text{raio}^2)$ ;  $\text{Perimetro} = (2 * \text{PI} * \text{raio})$ ;
- **Retângulo** =>  $\text{Area} = (\text{base} * \text{altura})$ ;  $\text{Perimetro} = (2 * \text{base}) + (2 * \text{altura})$