

ATIVIDADE

Assunto:

Classes abstratas.

Orientações:

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

Regras de criação dos programas:

Crie um novo projeto Java denominado **AtividadeClassesAbstratas**. As classes devem possuir os nomes informados no texto. Ao final, o projeto deve ser exportado para um arquivo em formato ZIP.

Nome completo:

Yan Pedro Façanha Brasileiro

1. Quais as diferenças entre classes abstratas e classes concretas? Explique.

Classes abstratas são aquelas que representam conceitos genéricos e incompletos, não podendo ser instanciadas diretamente. Elas servem como modelo para outras classes e podem conter métodos abstratos (sem implementação) e métodos concretos (com implementação).

Já as classes concretas são completas e podem ser instanciadas. Elas implementam todos os métodos necessários, inclusive os herdados de classes abstratas, se for o caso.

2. Classes abstratas podem ter métodos concretos? Explique.

Sim, classes abstratas podem ter métodos concretos. Isso ocorre porque elas podem fornecer uma implementação padrão para determinados comportamentos que serão compartilhados por todas as subclasses. Esses métodos concretos ajudam a evitar repetição de código e permitem que apenas os comportamentos realmente específicos sejam definidos nas classes filhas.

3. Em quais situações as classes abstratas devem ser utilizadas?

Classes abstratas devem ser utilizadas quando se deseja criar uma estrutura comum para várias classes relacionadas, mas sem permitir a criação de objetos diretamente a partir dessa estrutura. Elas são úteis quando existe um comportamento padrão a ser compartilhado, mas também há comportamentos que precisam ser definidos de forma diferente por cada subclasses. Também são usadas quando se quer forçar a implementação de certos métodos nas subclasses, através de métodos abstratos.

4. Se uma classe abstrata não pode ser instanciada, explique porque o código-fonte a seguir funciona:

```
Poligono[] p = new Poligono[10];
```

Esse código funciona porque ele não está criando objetos da classe Poligono, e sim um vetor que pode armazenar referências de objetos do tipo Poligono ou de qualquer subclasses concreta que herde de Poligono. Isso é possível porque em Java é permitido criar vetores ou variáveis de tipos abstratos, desde que os objetos realmente instanciados sejam de classes concretas que estendem a classe abstrata.

5. Demonstre, através de um código-fonte simplificado, o uso de classes abstratas e concretas em uma hierarquia de herança. Devem ser inseridos ao menos 2 métodos abstratos e um construtor na classe abstrata.

```
abstract class Poligono {
    protected int lados;

    public Poligono(int lados) {
        this.lados = lados;
    }

    public abstract double calcularArea();
    public abstract double calcularPerimetro();

    public void mostrarInfo() {
        System.out.println("Número de lados: " + lados);
    }
}

class Triangulo extends Poligono {
    private double base;
    private double altura;

    public Triangulo(double base, double altura) {
        super(3);
        this.base = base;
        this.altura = altura;
    }

    @Override
    public double calcularArea() {
        return (base * altura) / 2;
    }

    @Override
    public double calcularPerimetro() {
        return base * 3;
    }
}

class Quadrado extends Poligono {
    private double lado;

    public Quadrado(double lado) {
        super(4);
        this.lado = lado;
    }

    @Override
    public double calcularArea() {
        return lado * lado;
    }

    @Override
    public double calcularPerimetro() {
        return 4 * lado;
    }
}
```

Boa sorte!

Prof. Igor.