

## 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:

<b>Mac Myller da Silva Carlos</b>
-----------------------------------

1. Quais as diferenças entre classes abstratas e classes concretas? Explique.  
Uma classe abstrata normalmente possui métodos abstratos. Esses métodos são implementados nas suas classes derivadas concretas com o objetivo de definir o comportamento específico. O método abstrato define apenas a assinatura do método e, portanto, não contém código.
2. Classes abstratas podem ter métodos concretos? Explique.  
Classes abstratas podem sim ter métodos concretos mas classes concretas não podem ter métodos abstratos, pois para ter métodos abstratos a classe obrigatoriamente deve ser abstrata
3. Em quais situações as classes abstratas devem ser utilizadas?  
Classes abstratas podem descrever “como deve ser feito” e são utilizadas quando se conhece parte do comportamento do objeto, apenas assinando e mantendo métodos desconhecidos como abstratos.
4. Se uma classe abstrata não pode ser instanciada, explique porque o código-fonte a seguir funciona:

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

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 Carro{
    String modelo;
    public boolean portaAberta = false;
    Carro(String modelo){
        this.modelo = modelo;
    }
    public abstract void abrirPorta();
    public abstract void fecharPorta();
    public abstract boolean getPorta();
}

class CarroEsportivo extends Carro{
    CarroEsportivo(String modelo){
        super(modelo);
    }
    public void abrirPorta(){
        portaAberta = true;
    }
    public void fecharPorta(){
        portaAberta = false;
    }
    public boolean getPorta(){
        return portaAberta;
    }
}
```

Boa sorte!

Prof. Igor.