

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA TRIÂNGULO MINEIRO</p>	<p>MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA TRIÂNGULO MINEIRO - Campus Uberlândia Centro</p>
<p>Curso: Sistemas para Internet Disciplina: Programação Orientada a Objetos 2 Atividade – Interface, Classe Abstrata, Polimorfismo Professora: Cíbele Mara Fonseca</p>	

1. (classe abstrata, herança, sobrescrita, polimorfismo) Crie uma hierarquia de classes para representar os diferentes tipos de funcionários de um escritório que tem os seguintes cargos: gerente, assistente, vendedor. Escreva uma classe base abstrata chamada `Funcionario` que declara um método **abstrato**:

Assinatura	<code>double calculaSalario()</code>
------------	--------------------------------------

Esta classe também deve definir os seguintes atributos: `nome` (tipo `String`), `matricula` (tipo `String`) e `salario_base` (tipo `double`). Use encapsulamento e forneça um construtor que recebe os valores correspondentes a serem armazenados nos respectivos atributos. Esta classe abstrata deverá ser estendida pelas outras classes representativas dos tipos de funcionários, portanto devem ser escritas as classes `Gerente`, `Assistente` e `Vendedor`. Em cada classe deve-se sobrescrever o método `calculaSalario` de forma que cálculo do salário é feito assim: O gerente recebe duas vezes o `salario_base`, o assistente recebe o `salario_base` e o vendedor recebe o `salario_base` mais uma comissão definida no construtor de sua classe. Crie uma classe `Teste` com um método `main` que cria um objeto de cada tipo e os armazena em uma lista e depois calcula a folha salarial dos três funcionários e imprime o valor total. Indique quais conceitos de POO você usou e como foi usado.

2. (classe abstrata, sobrescrita, polimorfismo) Escreva uma classe abstrata chamada `CartaoWeb`. Essa classe representa todos os tipos de cartões web e conterá apenas um atributo: `destinatario` (tipo `String`). Nessa classe você deverá também declarar o método `public abstract void showMessage()`. Crie classes filhas da classe `CartaoWeb`: `DiaDosNamorados`, `Natal`, `Aniversario`. Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão. Cada classe também deve implementar o método `showMessage()`, mostrando uma mensagem ao usuário com seu nome e que seja específica para a data de comemorativa do cartão. Escreva uma classe com um método `main` crie uma lista de `CartaoWeb`. Insira instâncias dos 3 tipos de cartões nesta lista. Após, use um laço `for` para exibir as mensagens deste cartão chamando o método `showMessage()`. Em que linha(s) acontece polimorfismo nesse código?
3. (interface, polimorfismo) Escreva uma interface chamada `FormaGeometrica`, que declara os seguintes métodos:

Assinatura	<code>double calcularPerimetro()</code>
------------	---

Assinatura	<code>double calcularArea()</code>
------------	------------------------------------

Escreva uma classe para representar retângulos que implementa a interface `FormaGeometrica`, com os atributos `base` (tipo `double`) e `altura` (tipo `double`). O construtor recebe o tamanho da base e da altura e armazena nos respectivos campos. Esta classe deve implementar os métodos declarados na interface, usando as fórmulas: $\text{perímetro} = 2 \cdot \text{base} + 2 \cdot \text{altura}$ e $\text{área} = \text{base} \cdot \text{altura}$. Escreva uma classe para representar círculos que implementa a interface `FormaGeometrica`, com o atributo `raio` (tipo `double`). O construtor recebe o tamanho do raio e armazena no campo. Use sempre encapsulamento. Escreva um programa com um método `main` que cria um retângulo com base 3 e altura 4 e um círculo com raio 5 e armazene estes objetos em uma lista. Em seguida, imprima os perímetros e as áreas de todas as formas usando um laço `for`.

4. Considere a interface `Relacao` que define o relacionamento entre dois objetos de acordo com a "ordem natural" dos objetos:

```
interface Relacao {  
    boolean eMaior(Object a, Object b);  
    boolean eMenor(Object a, Object b);  
    boolean eIgual(Object a, Object b);  
}
```

Escreva uma classe chamada `Linha` que implementa a interface `Relacao`. A classe `Linha` possui quatro atributos nomeados `X1,Y1,X2,Y2` do tipo `int` para representar as coordenadas dos dois pontos extremos da linha no plano cartesiano. Escreva o método:

Assinatura	<code>double getTamanho()</code>
Efeito	Retorna o tamanho da linha. Use a fórmula $\text{Math.sqrt}((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1))$, sendo <code>X1,Y1</code> coordenadas do primeiro ponto e <code>X2,Y2</code> coordenadas do segundo ponto

Use *casting* para implementar os métodos da interface, pois os atributos dos métodos são declarados do tipo `Object` mas você precisa tratar de objetos do tipo `Linha`.

5. Escreva uma classe que representa retângulos com os atributos `base` e `altura` do tipo `int` e que implementa a interface do exercício 3. Use a área do retângulo como fator de comparação entre retângulos. Escreva um programa com um método `main` que cria 2 retângulos. Em seguida, usando os métodos da interface `Relacao`, imprima os atributos do retângulo com maior área.