 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</p> <p>Curso: Análise e Desenvolvimento de Sistemas</p> <p>Disciplina: Programação Orientada a Objetos</p> <p>Professor: Ely</p>
--	--

Exercício 09

Classes Abstratas

- Podemos instanciar classes abstratas? Justifique.
- Explique o que é necessário para que a compilação da ClasseConcreta ocorra sem erros:

<pre>abstract class ClasseAbstrata { abstract imprimaAlgo(): void ; }</pre>	<pre>class ClasseConcreta extends ClasseAbstrata { }</pre>
---	--

- Se uma classe que herda de uma abstrata e não implementa os seus métodos, o que ocorre?
- Imagine que você deve modelar várias figuras geométricas em TypeScript e que cada uma tem sua forma específica de calcular área e perímetro. Proponha e implemente uma hierarquia de classes usando uma classe abstrata chamada FiguraGeometrica e outras concretas: Quadrado, Triangulo, etc.
- Não podemos aplicar o operador new em FiguraGeometrica, mas porque então podemos realizar o seguinte código de instanciação:

```
abstract class FiguraGeometrica {
    //...
}

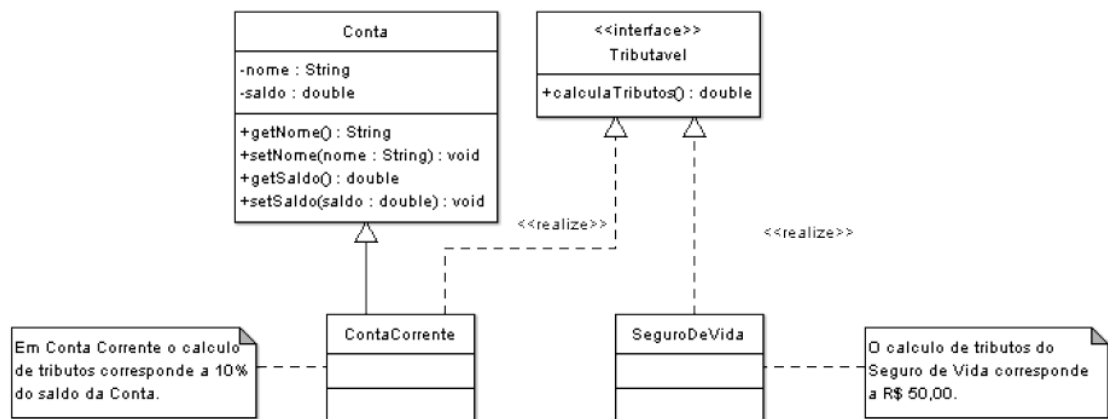
let figuras: FiguraGeometrica[] = new Array();
```

- Implemente as classes Funcionario, Gerente e Diretor conforme o diagrama exposto em sala:
 - A classe funcionário deve ser abstrata e o método getBonificacao() abstrato;
 - Na classe gerente o método bonificação deve retornar 40% do salário;

- c. Em Diretor a bonificação deve ser 60% do salário.
- d. Por fim, na classe presidente o método deve retornar 100% do salário + R\$ 1.000,00.

Interfaces

7. Refaça a questão 04 do exercício usando interfaces com os métodos propostos em vez de herança. Crie também um script que instancie e teste diferentes formas geométricas.
8. Crie uma interface chamada IComparavel com um método chamado comparar que receba uma forma geométrica como parâmetro e retorna um inteiro como resultado. Implemente em cada uma das classes do exemplo anterior a interface retornando -1, 0 e 1 caso a área da forma seja menor, igual ou maior que a passada via parâmetro.
9. Crie uma classe para testar os exemplos anteriores. Instancie várias formas diferentes. Pegue duas formas chame em uma delas o método comparar passando a outra como parâmetro e exiba o resultado. Repita para outras formas.
10. Implemente o diagrama de classes abaixo:



11. Crie uma classe chamada **AuditoriaInterna** que tenha dois métodos que tenham um array de **Tributaveis** e os métodos:
 - a. `adicionar(Tributável);`
 - b. `calcularTributos();` retorna um `double` que representa a soma de todos os cálculos dos tributos de todos os tributáveis;
 - c. Crie uma classe de testes que instancie várias classes **ContaCorrente** e **SeguroDeVida**, adicione-as na classe **AuditoriaInterna** e exiba o resultado

do método `calculaTributos`. Perceba que a classe de auditoria não se preocupa que tipo de classe está sendo passada.

12. Altere a aplicação feita sobre redes sociais para:

- a. Tratar erros e capturar exceções;
- b. Utilizar um mecanismo de persistência independente:
 - i. Ter interfaces chamadas `IRepositorioPerfis` e `IRepositorioPostagens` com métodos de consulta e inclusão relacionada à persistência de dados;
 - ii. Usar um mecanismo de persistência alternativo que implemente as interfaces. Sugestão: banco de dados SQL;
 - iii. Na hora de inicializar o App, você deve escolher o mecanismo de persistência e instanciar as implementações das interfaces (array, banco de dados, arquivo);
 - iv. A classe rede social deve ter, em vez de repositórios comuns, interfaces e funcionar sem alterações independente da implementação.