



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
MÓDULO IV - ENGENHARIA DE SOFTWARE III
PROF.: Ely Miranda
ALUNO: Lucas Gomes de Oliveira (20191ADS0185)**

1) Podemos instanciar classes abstratas? Justifique.

Na hierarquia de classes, as subclasses são progressivamente mais específicas, enquanto as superclasses são correspondentemente mais abrangentes. Às vezes, faz sentido fazer com que a superclasse seja tão geral que, por exemplo, o conteúdo de todos os métodos não seja especificado. Alguns métodos consistem apenas nas chamadas assinaturas estendidas (nome do método com parâmetros formais e seus tipos, delimitadores e tipo de retorno), e o conteúdo é deixado para ser especificado por subclasses.

Classes abstratas não podem ser instanciadas (isto é, realizar a criação de objetos), apesar de poderem ser herdadas. Sendo assim, a instanciação é possível somente através das classes concretas.

Se você estiver declarando uma classe derivada de uma classe abstrata, mas quiser criar objetos de um novo tipo, terá que fornecer definições para todos os métodos abstratos da classe base. Se isso não for feito, a classe derivada também permanece abstrata.

2) Explique o que é necessário para que a compilação da `ClasseConcreta` ocorra sem erros:

```
// @errors: 2511
abstract class Base {
  abstract getName(): string;

  printName() {
    console.log("Hello, " + this.getName());
  }
}
```

```
const b = new Base();
```

A classe não-abstrata “Classe Concreta” não implementa o membro abstrato herdado da classe “Classe Abstrata”. Sendo assim, é preciso alterar o modificador de acesso do método da classe abstrata, o método cujo função é a de imprimir algo no console.

3) Se uma classe que herda de uma abstrata e não implementa os seus métodos, o que ocorre?

Método abstrato é um método que não tem um corpo, ou seja, um método não implementado. Por exemplo, você pode criar um método abstrato para calcular a área de uma forma em classe abstrata. Todas as outras classes derivadas da classe principal já podem implementar seu código para o método finalizado.

Se uma subclasse é derivada de uma superclasse que contém um método abstrato e se esse método abstrato não for redefinido na subclasse, esse método permanece abstrato na subclasse. Com isso, a subclasse também deverá ser declarada explicitamente como abstract.

Para evitar descrever os dados várias vezes, às vezes é razoável criar uma superclasse onde alguns métodos consistem apenas nos chamados assinaturas estendidas e o conteúdo é deixado para ser especificado por subclasses.

4)

```
abstract class FiguraGeometrica {
    quantidadeLado: number = 1

    lado1: number = 1

    lado2: number = 1

    area: number = this.lado1 * this.lado2

    perimetro: number = this.lado1 + this.lado2
}

class Quadrado extends FiguraGeometrica {
    lado1 = 12
    lado2 = this.lado1

    calculaArea(): number {
        return this.area
    }

    calculaPerimetro(): number {
```

```

        return this.perimetro
    }
}

class Triangulo extends FiguraGeometrica {
    lado1 = 12
    lado2 = 6
    lado3: number = 8

    calculaArea(): number {
        return this.area / this.lado3
    }

    calculaPerimetro(): number {
        return this.perimetro + this.lado3
    }
}

```

6)

```

abstract class Funcionario {
    protected salario: number;

    constructor(salario: number){
        this.salario = salario;
    }

    abstract getBonificacao(): number;
}

class Gerente extends Funcionario {
    getBonificacao(): number {
        return this.salario * 0.4;
    }
}

class Diretor extends Funcionario {
    getBonificacao(): number {
        return this.salario * 0.6;
    }
}

```

```
class Presidente extends Funcionario {  
    getBonificacao(): number {  
        return this.salario + 1000;  
    }  
}  
  
let Possidonio: Gerente = new Gerente(2000);  
console.log(Possidonio.getBonificacao());
```