

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

CURSO REFORÇO POO01

LUCAS FERNANDO
KELLY CRISTINA



INTERFACE



INTERFACE

- A interface é um recurso muito utilizado em Java;
- “obriga” um determinado grupo de classes a ter métodos ou propriedades em comum, onde são implementados em cada classe de uma maneira diferente.

INTERFACE

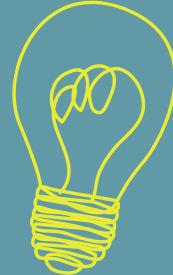
A classe que herdar uma interface é obrigada a implementar todos os métodos dessa classe que foi especificada;

Dentro da interface eu coloco assinaturas de métodos sem implementá-los;

INTERFACE

"Um contrato que quando assumido por uma classe deve ser implementado".

**Pode ser
entendida como
um manual de
implementação
para outras
classes;**



INTERFACE

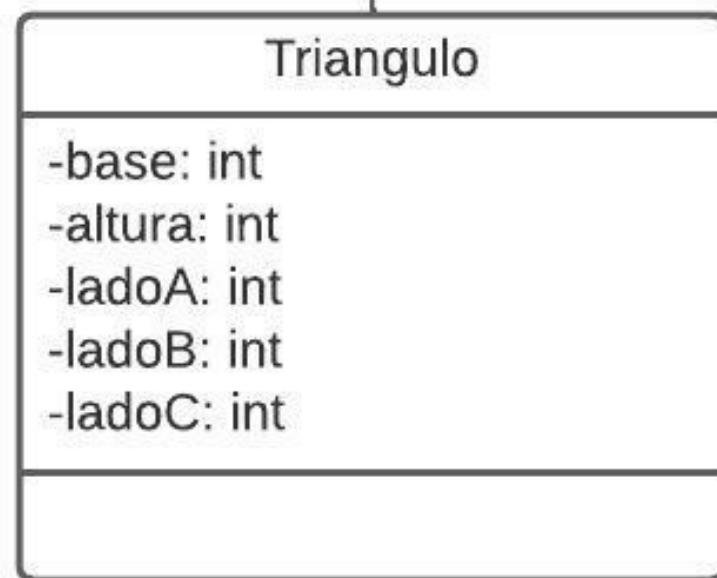
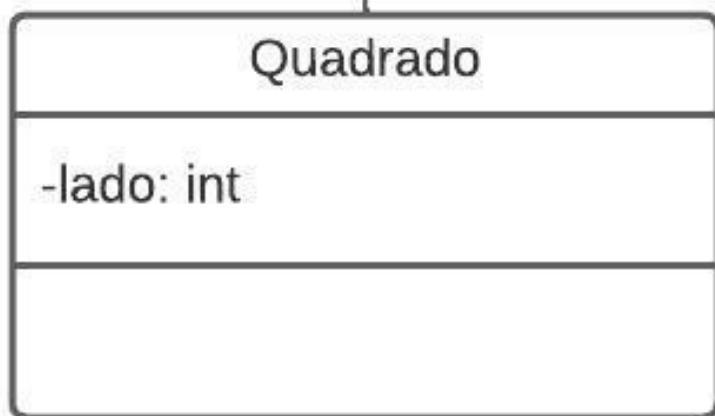


EXEMPLO



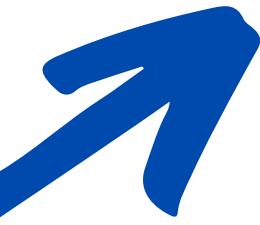
<<FiguraGeometrica>>

+ `getNomeFigura()` : string
+ `getArea()` : int
+ `getPerimetro()` : int



```
1 public interface FiguraGeometrica
2 {
3     public String getNomeFigura();
4     public int getArea();
5     public int getPerimetro();
6 }
```

PALAVRA RESERVADA



```
1 public interface FiguraGeometrica  
2 {  
3     public String getNomeFigura();  
4     public int getArea();  
5     public int getPerimetro();  
6 }
```

```
1 public interface FiguraGeometrica  
2 {  
3     public String getNomeFigura();  
4     public int getArea();  
5     public int getPerimetro();  
6 }
```

NÃO IMPLEMENTA OS MÉTODOS

CLASSE QUADRADO

```
public class Quadrado implements FigurasGeometricas {  
  
    private int lado;  
  
    public int getLado () {  
        return lado;  
    }  
  
    public void setLado (int lado) {  
        this.lado = lado;  
    }  
  
    @Override  
    public int getArea () {  
        int area = 0;  
        area = lado * lado;  
        return area;  
    }  
}
```

```
@Override  
public int getPerimetro() {  
    int perimetro = 0;  
  
    perimetro = lado * 4;  
    return perimetro;  
}  
  
@Override  
public String getNomeFigura() {  
    return "quadrado";  
}  
}
```

CLASSE TRIÂNGULO

```
public class Triangulo implements FigurasGeometricas{  
    private int base;  
    private int altura;  
    private int ladoA;  
    private int ladoB;  
    private int ladoC;
```

```
@Override  
public String getNomeFigura() {  
    return "Triangulo";  
}
```

```
@Override  
public int getArea() {  
    int area = 0;  
    area = (base * altura) / 2;  
    return area;  
}
```

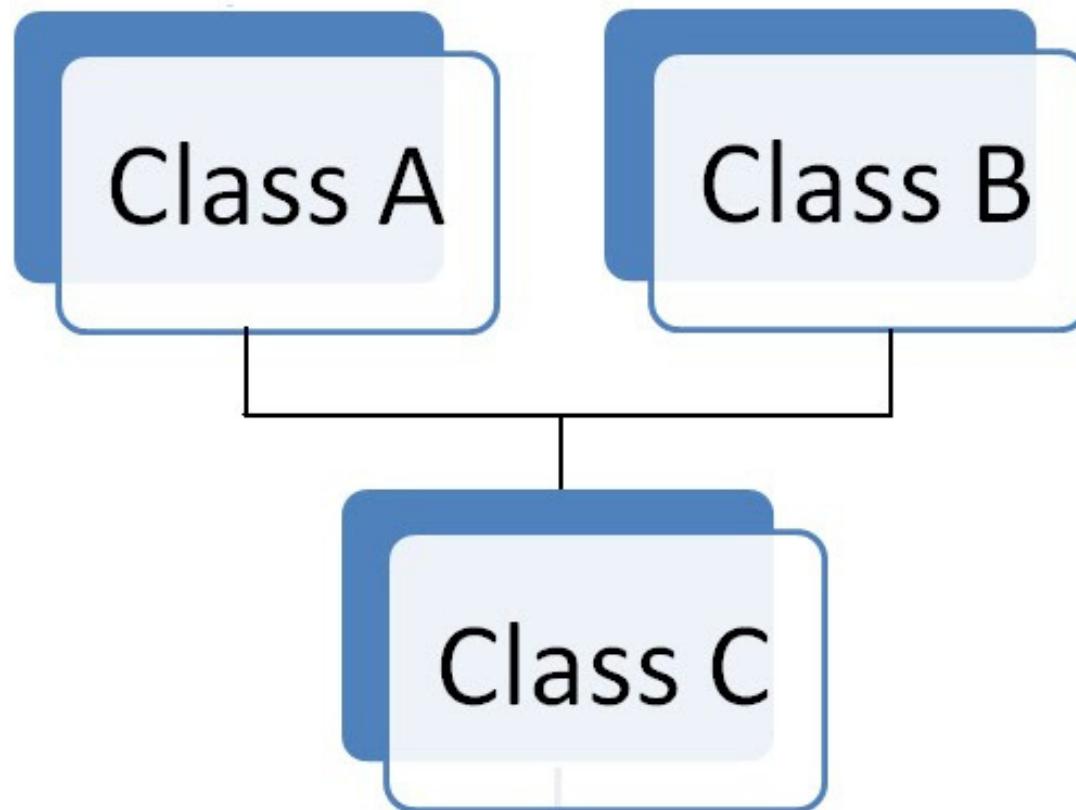
```
@Override  
public int getPerimetro() {  
    int perimetro = 0;  
    perimetro = ladoA + ladoB + ladoC;  
  
    return perimetro;  
}
```

- COMO É POSSÍVEL VER ACIMA, AMBAS AS CLASSES SEGUIRAM O CONTRATO DA INTERFACE FIGURAGEOMETRICA, PORÉM CADA UMA DELAS A IMPLEMENTOU DE MANEIRA DIFERENTE.
- AO CONTRÁRIO DA HERANÇA QUE LIMITA UMA CLASSE A HERDAR SOMENTE UMA CLASSE PAI POR VEZ, É POSSÍVEL QUE UMA CLASSE IMPLEMENTE VARIAS INTERFACES AO MESMO TEMPO.

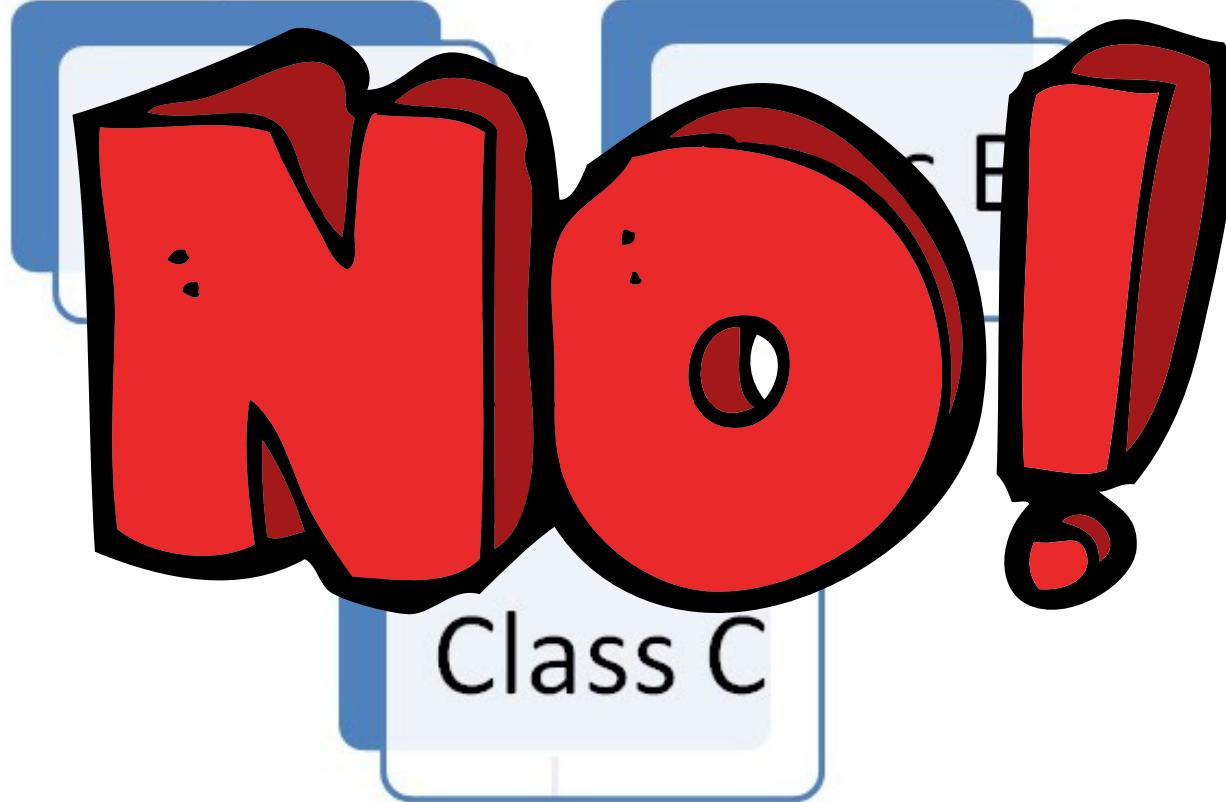
OBSERVAÇÃO



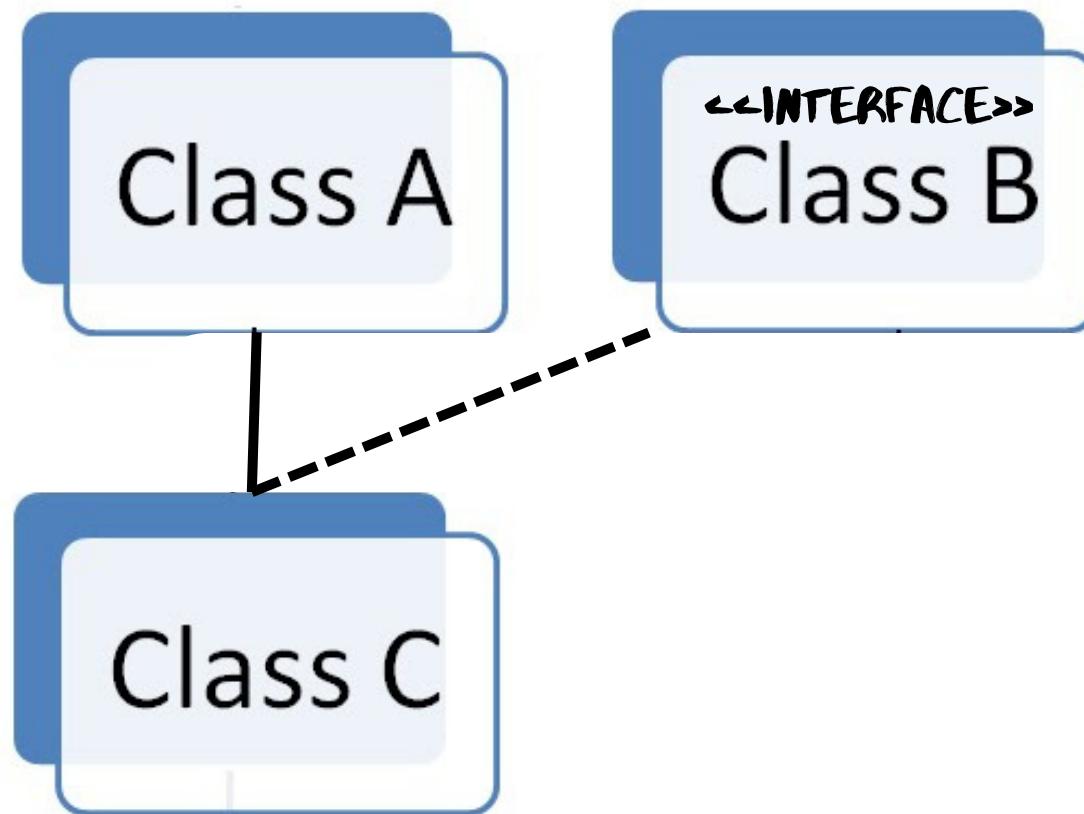
É POSSÍVEL FAZER ISSO NO JAVA?



É POSSÍVEL FAZER ISSO NO JAVA?



SOLUÇÃO:



JAVA NÃO PERMITE
HERANÇA MÚLTIPLA



SOLUÇÃO:

INTERFACES



- JAVA RESTRINGE HERANÇA MÚLTIPLA PARA EVITAR PROBLEMAS CAUSADOS POR AMBIGUIDADE QUE PODE AUMENTAR SE DUAS CLASSES PAI TIVEREM UM MÉTODO COM A MESMA ASSINATURA;
- O COMPILADOR NÃO PODE DECIDIR QUAL MÉTODO CHAMAR NESSA SITUAÇÃO.
- ISSO É RESOLVIDO POR INTERFACES;



1

EXEMPLO

```
① interface Employee {  
②     void getData(int empId, String empName);  
③ }  
④  
⑤ class DemoClass implements Employee {  
⑥     public void getData(int empId, String empName) {  
⑦         System.out.println("Employee ID: " + empId);  
⑧         System.out.println("Employee Name: " + empName);  
⑨     }  
⑩ }  
⑪  
⑫     public static void main(String[] args) {  
⑬         DemoClass object = new DemoClass();  
⑭         object.getData(12, "Joe");  
⑮     }  
⑯ }
```

Interface

Class



Output - DemoClass (run) ×



run:

Employee ID: 12

Employee Name: Joe

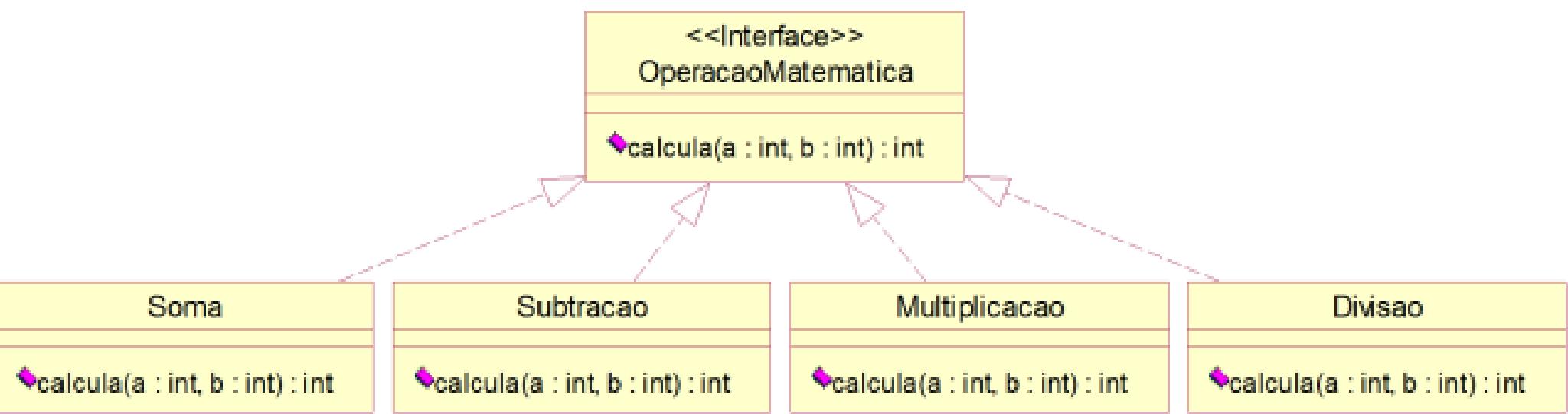
Output

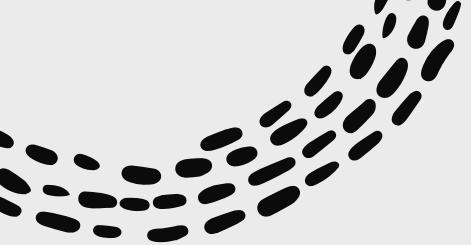
BUILD SUCCESSFUL (total time: 0 seconds)

PRÁTICA



EXERCÍCIO





REFERÊNCIAS



GUANABARA (CURSO EM VÍDEO)



DEV MEDIA



FIM

DÚVIDAS OU PERGUNTAS?

