Fabio de Moura Camargo Neto
github.com/bio353

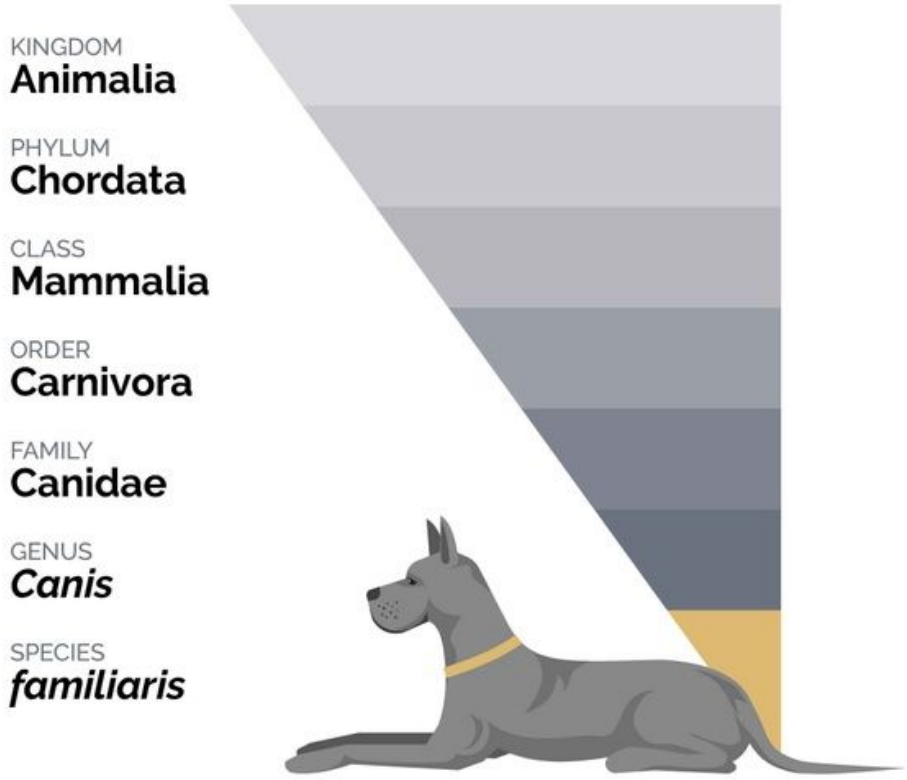
Classe Abstrata e Interface

Classe Abstrata

Definição

- Uma classe abstrata (`abstract`) é um tipo de classe que não pode ser instanciada, apenas herdada por outras classes.
    - Ou seja, não é possível criar objetos com classes abstratas, mas é possível herdá-las para utilizar seus atributos e métodos nas classes filhas.
- Essas classes são muito importantes quando não queremos criar um objeto a partir de uma classe “geral”, apenas de suas “subclasses”.

Exemplo



Um cão (*Canis familiaris*) é um **animal**, um **cordado**, um **mamífero**, um **carnívoro**, um **canídeo**. Pra simplificar, todo cão é um animal e um mamífero, mas esses conceitos não fazem sentido por si só.

```
public abstract class Animal {
    // Atributos e Métodos da Classe
}
```

```
public abstract class Mamifero extends Animal {
    // Atributos e Métodos da Classe

    public void amamentarFilhote(Mamifero filhote) {
      // Método de Exemplo
    }
}
```

```
public class Cachorro extends Mamifero {
    private String nome;

    public Cachorro(String nome) {
        this.nome = nome;
    }

    public void receberCarinho() {
```

```
        System.out.println("Ai, Salsicha, para com isso!")
    }
}
```

Se quisermos criar vários cachorros, podemos instanciar a classe cachorro e referenciar os objetos através de um vetor de `Animal` ou de `Mamifero`. Por exemplo:

```
public final class Main {
    public static void main(String[] args) {
        Cachorro c1 = new Cachorro("Scooby-Doo");
        Cachorro c2 = new Cachorro("Snoopy");
        Cachorro c3 = new Cachorro("Oddie");

        Animal[] animais = {c1, c2, c3};

        // Resto do Código ...
    }
}
```

Interface

Definição

- A interface é um recurso muito utilizado em Java e em outras linguagens orientadas a objeto para “obrigar” a um determinado grupo de classes a ter métodos ou propriedades em comum, ao mesmo tempo que obriga que todos os métodos sejam implementados em cada classe de uma maneira diferente.
    - Pode-se dizer, a grosso modo, que uma interface é uma espécie de contrato que, quando assumido por uma classe, deve ser implementado.
- Assim como as classes abstratas, não podem ser instanciadas.
- Uma vantagem da interface é que ela pode ser utilizada em conjunto com herança e, em Java, é possível que uma classe implemente diversas interfaces.

Exemplo

Consideramos uma figura geométrica. Que características em comum uma figura tem com a outra? Algumas nem se quer tem lados, como o círculo, e cada uma tem maneiras totalmente diferentes de calcular área e perímetro.

Por isso, consideremos o código a seguir:

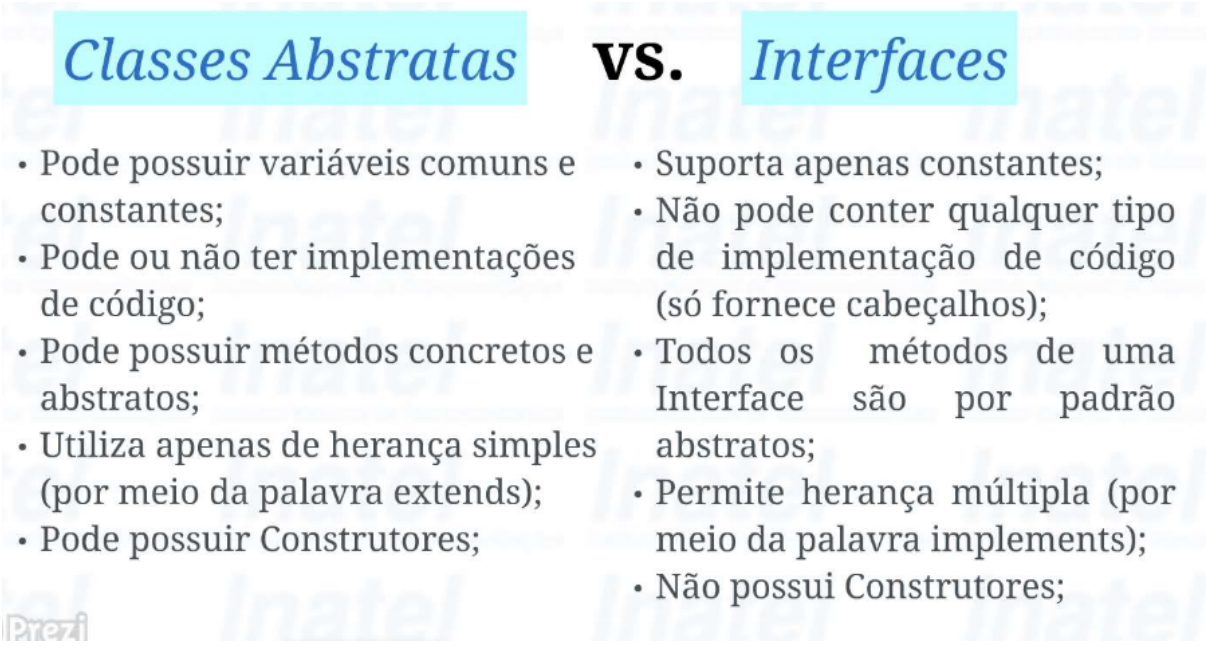
```
public interface FiguraGeometrica
{
    public double calculaArea();
    public double calculaPerimetro();
}
```

Para implementarmos uma interface, devemos utilizar a palavra reservada `implements`.

```
import java.lang.Math;

public class Circulo implements FiguraGeometrica {
    private final static double PI = 3.141592;
    private double raio;

    public Circulo(double raio) {
        this.raio = raio;
    }

    @Override
    public double calculaArea() {
        return PI * Math.pow(raio, 2);
    }

    @Override
    public double calculaPerimetro() {
        return 2 * PI * raio;
    }
}
```

Observe que todos os métodos foram obrigatoriamente sobreescritos.

Diferenças



Direto do material do RenZo porque estou sem tempo pra nada... 😕

Exercícios Práticos

Implemente o seguinte diagrama de classes: