

INSTITUTO MAUÁ DE TECNOLOGIA



Linguagens I

Classes Abstratas

Prof. Tiago Sanches da Silva
Prof. Murilo Zanini de Carvalho

Pilares da Programação Orientada a Objetos

Abstração

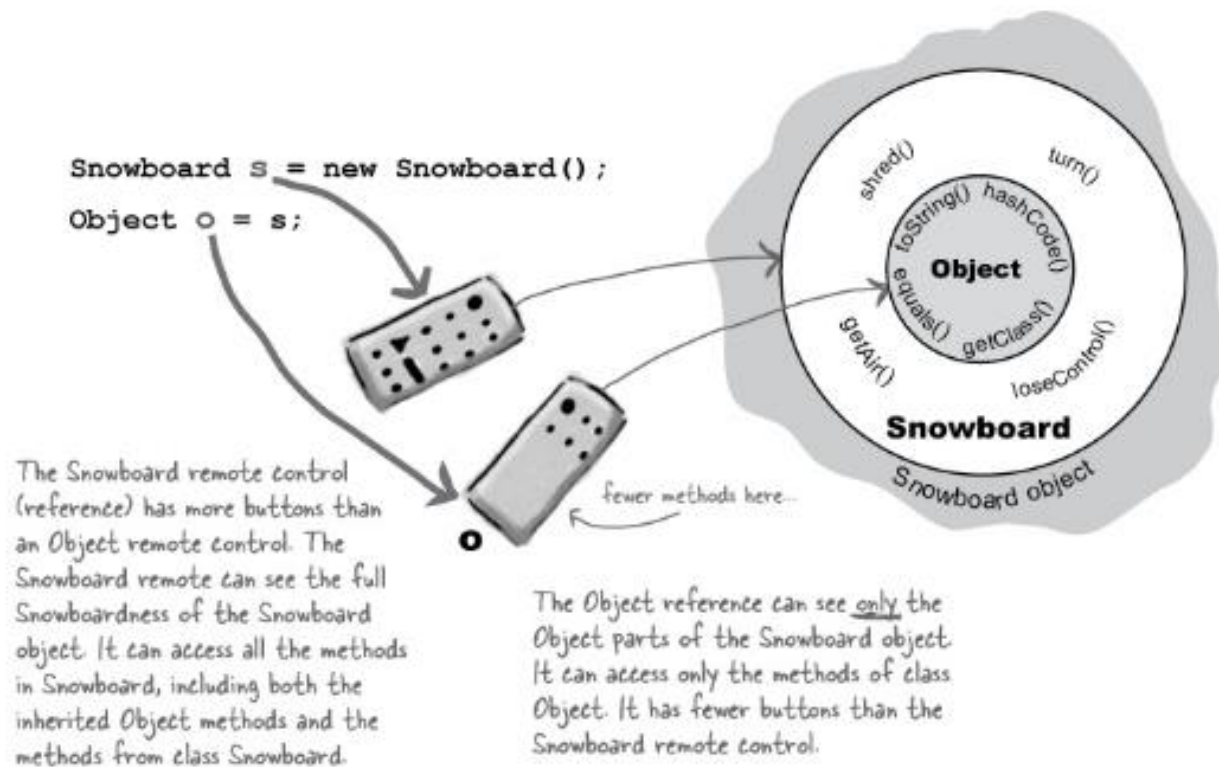
Encapsulamento

Herança

Polimorfismo

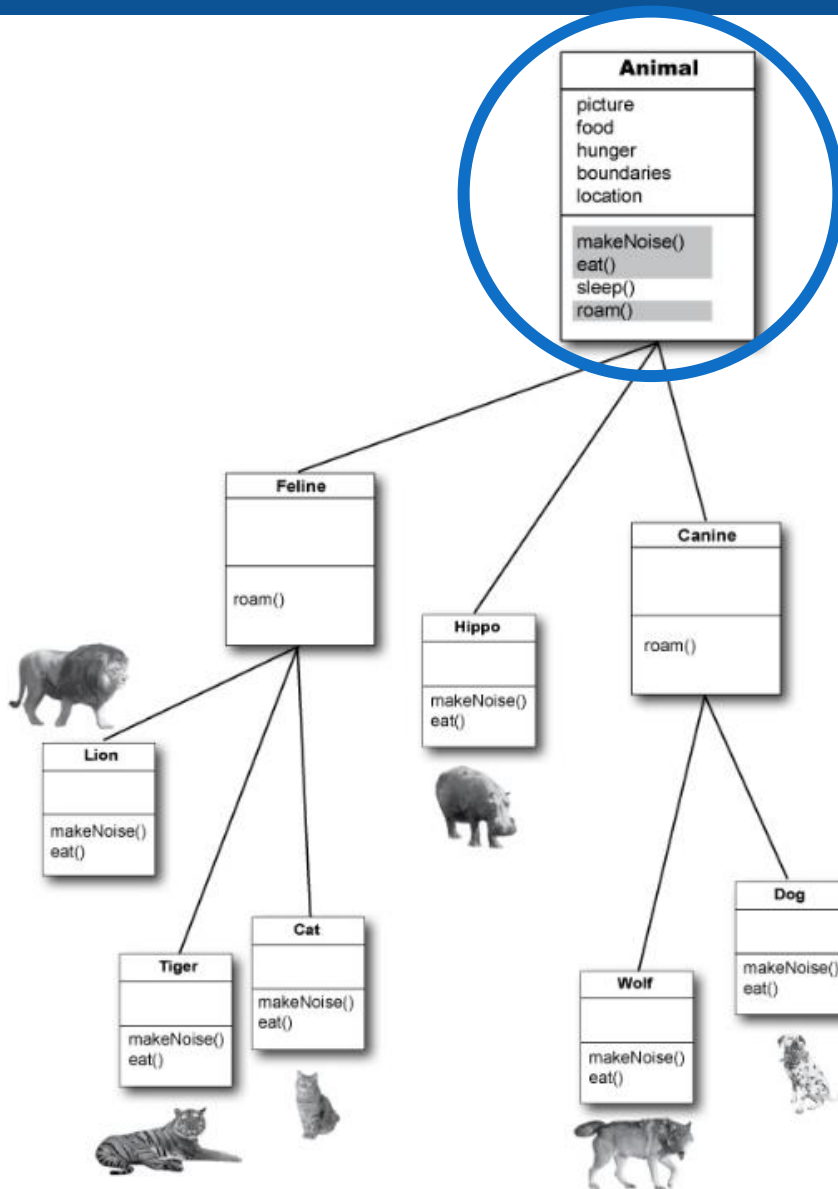
- Capacidade de representar elementos (físicos com ou não) por suas características e comportamentos.
- Isolar a forma como a abstração é realizada, fornecendo uma interface de acesso a ela.
- Transmissão de características de um elemento geral para elementos mais específicos.
- Possibilidade de executar o método correto de acordo com a situação/referência utilizada.

Herança

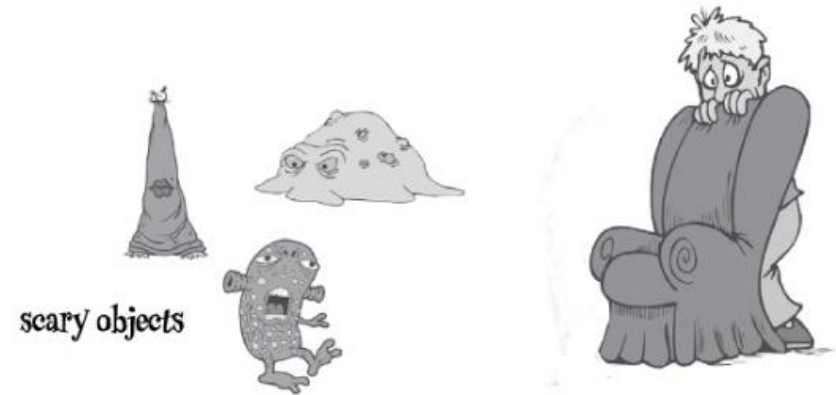


Retirado de: "Head First! Java"

Herança



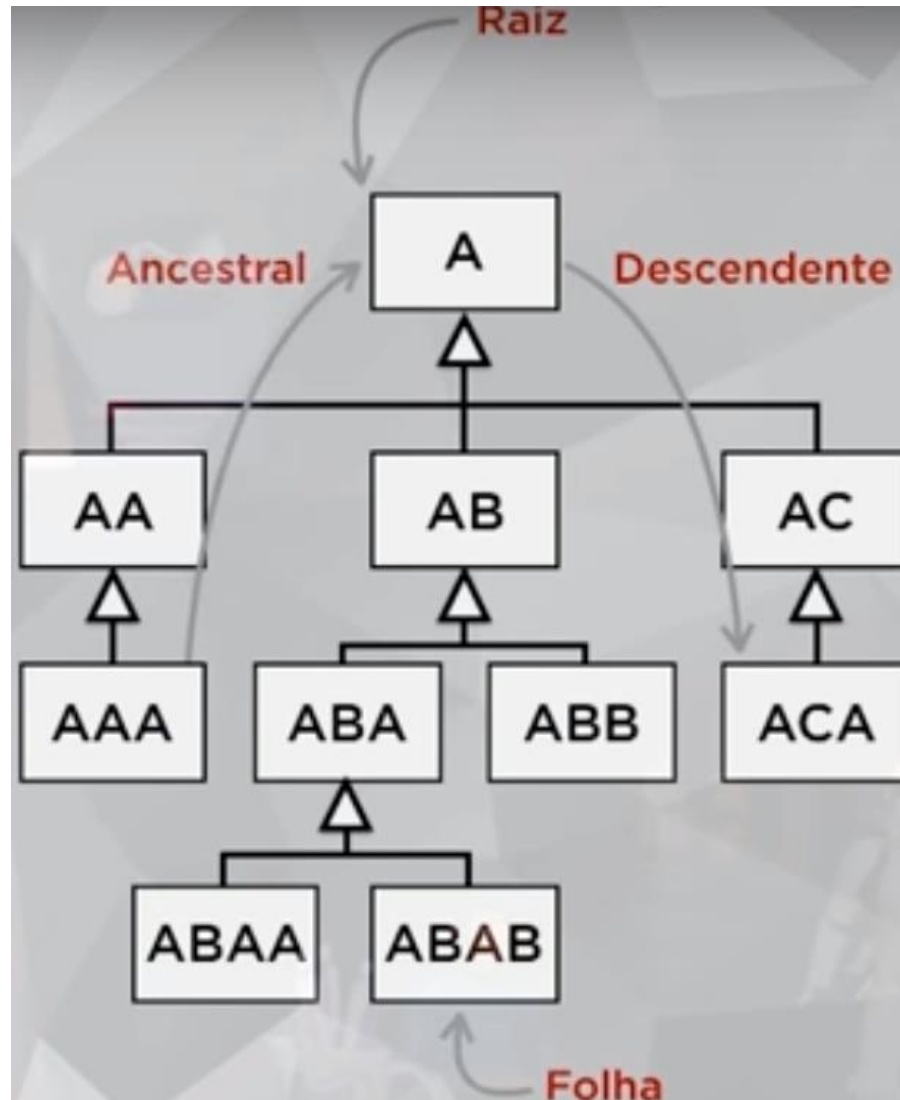
What does a new `Animal()` object *look* like?



Retirado de: "Head First! Java"

Classes abstratas

Herança – Conceitos gerais



Definição

Uma Classe Abstrata é desenvolvida para representar entidades e conceitos abstratos. A classe abstrata é sempre uma superclasse que não possui instâncias (ou melhor, não pode ser instanciada).

Ela define um modelo para determinada funcionalidade e geralmente fornece uma implementação incompleta – a parte genérica – dessa funcionalidade.

Definição

Cada uma das classes derivadas da classe abstrata, completa a funcionalidade dessa classe abstrata, adicionando um comportamento específico (especialização).

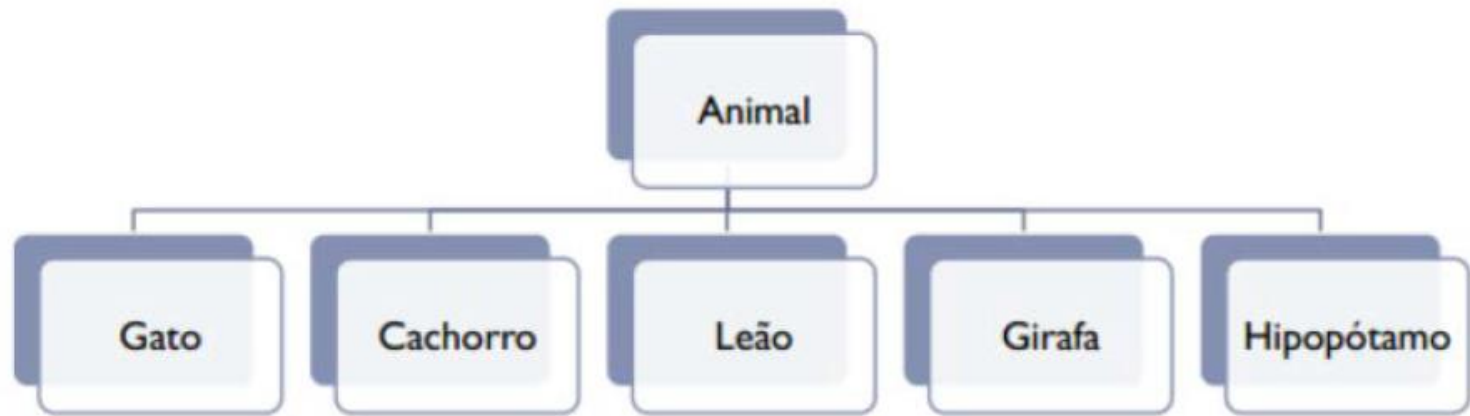
Nomenclaturas

Classes Abstratas: Conceitos e ideias, classes que não possuem instâncias diretas, porém suas descendentes podem ter.

Classes Concretas: Classes que possuem uma implementação completa e podem ser instanciadas.



Classes Abstratas



Não existe instância da classe *Animal*, apenas das classes especializadas.

Classes Abstratas



Métodos Abstratos

Métodos abstratos

Podemos definir um método como abstrato, isso implica, **obrigatoriamente**, que esse método deverá ser implementado nas classes descendentes concretas.

Em Java, podemos declarar um método e uma classe como abstratos utilizando a palavra chave ***abstract***.

Exemplo de implementação - Java

```
abstract class Animal {  
    public abstract void comer();  
}  
  
class Lobo extends Animal {  
    @Override  
    public void comer() {  
        System.out.println("Eu me alimento como um lobo!");  
    }  
}  
  
class Peixe extends Animal {  
    @Override  
    public void comer() {  
        System.out.println("Eu me alimento como um peixe!");  
    }  
}  
  
public class Teste {  
    public static void main(String[] args) {  
        Animal a = new Lobo();  
        Animal b = new Peixe();  
  
        a.comer();  
        b.comer();  
    }  
}
```

Métodos abstratos

Uma classe abstrata pode possuir métodos concretos e abstratos.

Se uma classe possui ao menos um método abstrato, então a classe precisa ser abstrata!

Classes abstratas

Classe Abstrata

Não pode ser instanciada.
Só pode servir como
progenitora.

Método Abstrato

Declarado, mas não
implementado na
progenitora.

Classe Final

Não pode ser herdada por
outra classe.
Obrigatoriamente folha.

Método Final

Não pode ser sobrescrito
pelas suas sub-classes.
Obrigatoriamente herdado

Regras

- Toda classe que possui métodos abstratos é automaticamente abstrata e deve ser declarada como tal;
- Uma classe pode ser declarada abstrata mesmo se ela não possuir métodos abstratos;
- Uma classe abstrata não pode ser instanciada;
- Uma classe que é descendente direta de uma classe abstrata pode ser instanciada somente se:
 - 1. Sobrescrever e fornecer implementação para cada um dos métodos abstratos de sua superclasse;
- **Se uma subclasse de uma classe abstrata não implementar todos os métodos abstratos que ela herda, então a subclasse também é abstrata.**

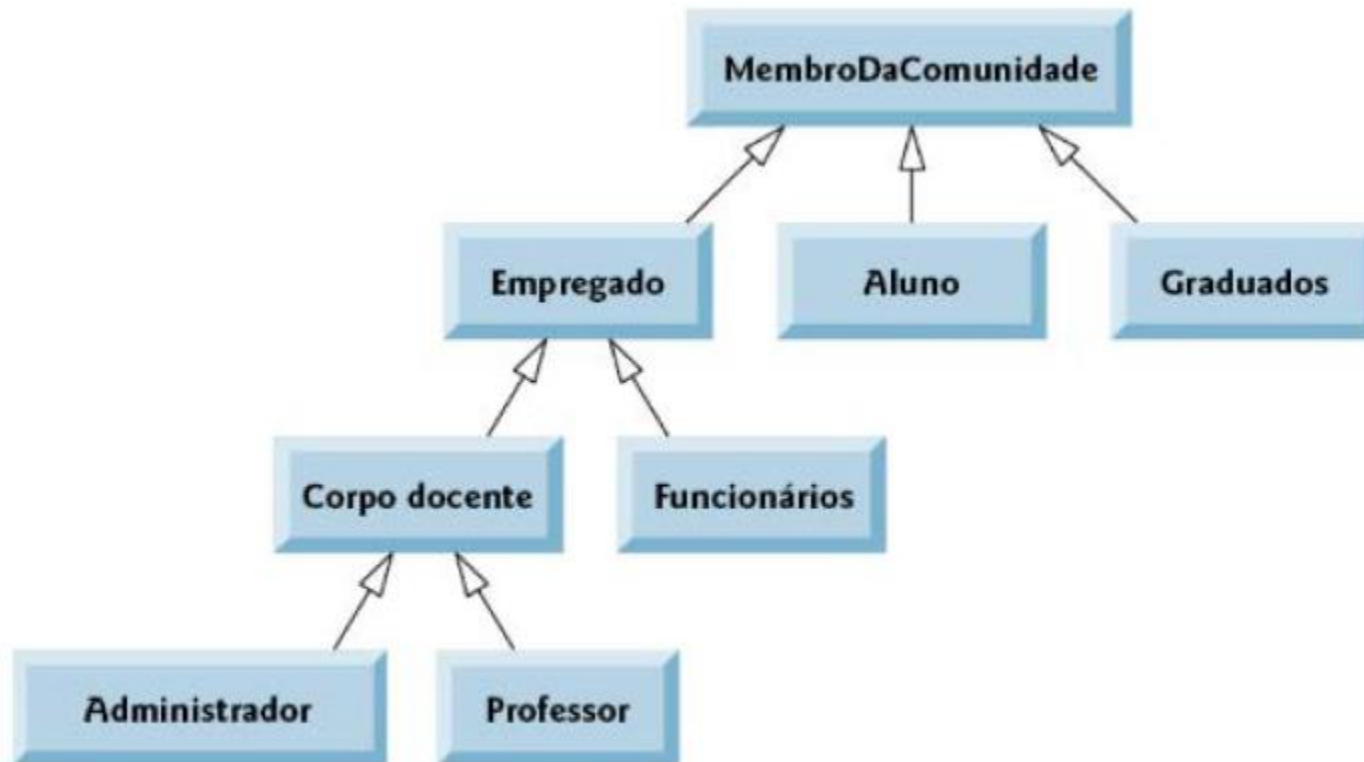
Por que não utilizar herança comum?

A única diferença é que não podemos instanciar a classe abstrata, e se tratando de conceitos abstratos, utilizar classes abstratas adiciona coerência e consistência ao sistema.

A decisão de transformar ou não uma classe em abstrata depende do domínio do seu problema!



O que é o que?



Perguntas?