Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Programação Orientada a Objetos

2

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROF. LUCIO AGOSTINHO ROCHA

AULA 11:
INTERFACES E CLASSES ABSTRATAS

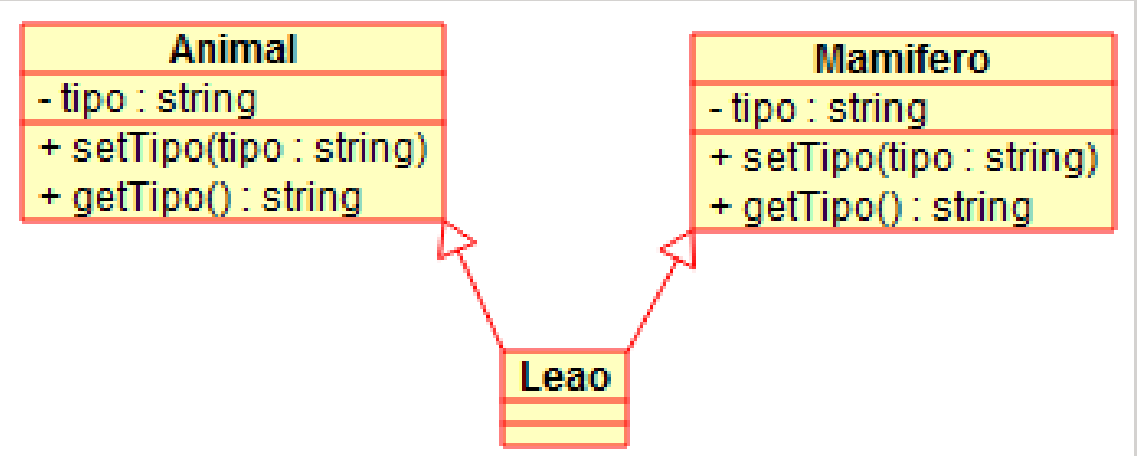
1º.SEMESTRE 2023

Interfaces

Interfaces

- Interfaces:
  - Java não suporta herança múltipla, mas admite múltiplas Interfaces.
  - Classes implementam Interfaces
    - A Interface garante que as classes implementem os métodos.
    - Métodos na interface devem ser declarados ‘public abstract’
  - Interfaces permitem que métodos sejam implementados em Interfaces diferentes, e não todos em uma única classe.
  - Ao implementar uma interface a classe explicitamente deve definir qual método será implementado.

Interfaces



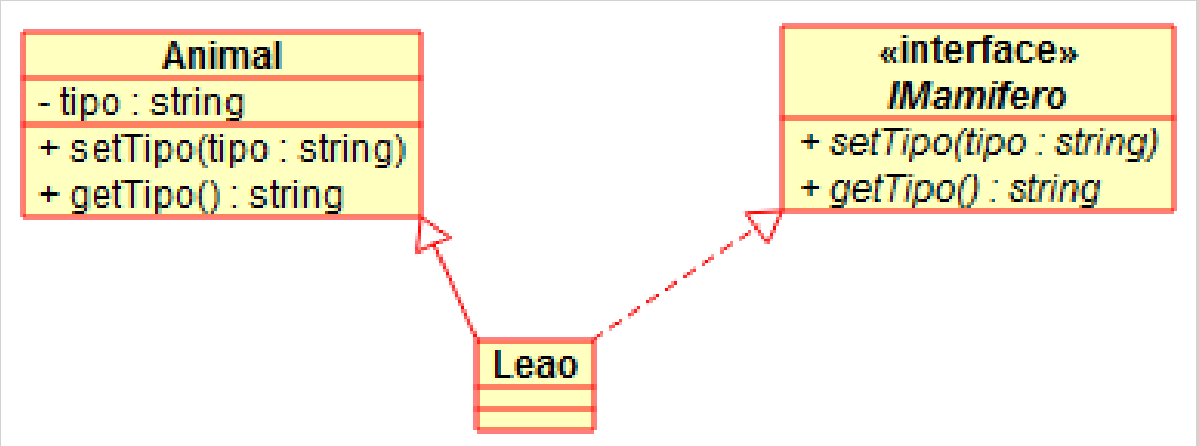
- Figura: Diagrama de Classes com Herança Múltipla.

Interfaces


```
 1 #include <iostream>
 2 #include <string>
 3
 4 #include "Leao.h"
 5
 6 using namespace std;
 7
 8 int main() {
 9
10     Leao leao;
11     cout << leao.Animal::getTipo() << endl;
12     cout << leao.Mamifero::getTipo() << endl;
13
14
15     return 0;
16 }
17
```

- C++ explicitamente informa a superclasse na chamada do método.

Interfaces



- Figura: Diagrama de Classes com Interface.

Interfaces


```
1
2  public interface IMamifero {
3
4      public final String tipo="mamifero";
5
6      public void setTipo(String tipo);
7
8      public String getTipo();
9  }
10
```

- Java: 1) Declaração dos métodos da interface.

Interfaces


```
1
2  public class Leao extends Animal implements IMamifero{
3
4      public Leao(){
5
6      }
7
8      public String toString(){
9          return this.getTipo();
10     }
11
12 }
13
```

- Java: 2) métodos da interface devem ser implementados ou sobrescritos.

Interfaces


```
1
2  public class Principal {
3
4      public static void main(String[] args) {
5
6          Leao leao = new Leao();
7
8          System.out.println(leao);       //Animal
9          System.out.println(leao.tipo); //Mamifero
10     }
11
12 }
13
```

- Java: 3) Declaração e Instanciação do objeto.

Classes Abstratas

Classes Abstratas

- Classe Abstrata:
  - Permitir que todas as classes herdem umas das outras é um risco de segurança.
  - Classe abstrata: fornece uma superclasse para a qual outras classes podem herdar.
  - Não instanciam objetos.
  - Subclasses devem implementar todos os métodos abstratos. Se não, a subclasse se torna abstrata.
- Classe Concreta:
  - Classes que permitem instanciar objetos.
  - Fornece modelo para instanciar objetos específicos.
    - Ex.: Circulo, Quadrado, Triangulo, Rosa, Margarida, Samambaia.

Classes Abstratas

Estudo de Caso com Classe Abstrata:

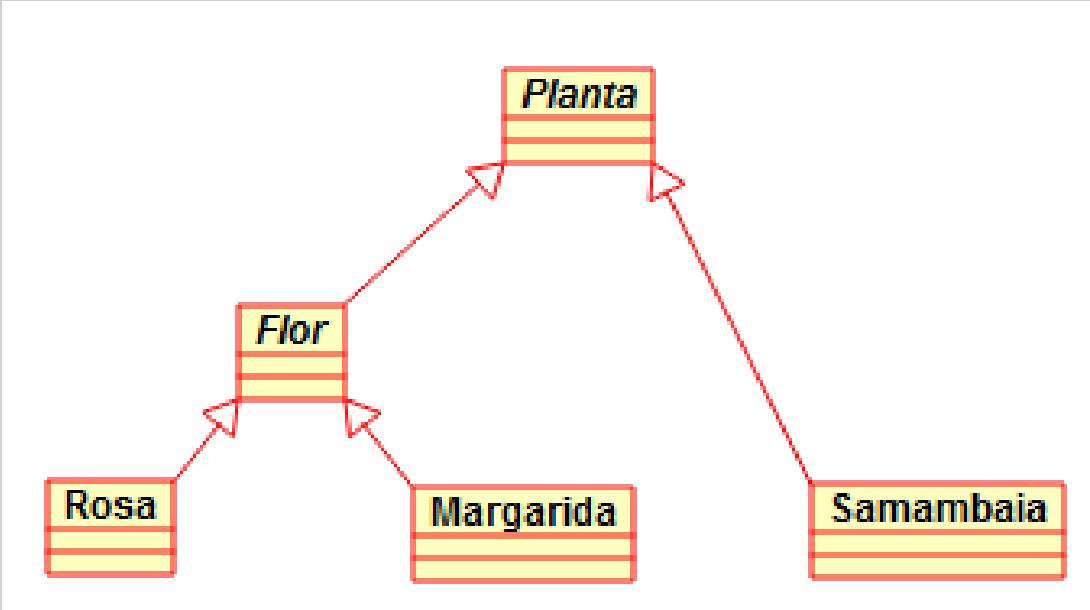


Figura: Classe Planta e Classe Flor são classes Abstract. Subclasses folha não deveriam ser herdadas (final).

Classes Abstratas

Estudo de Caso com Classe Abstrata:

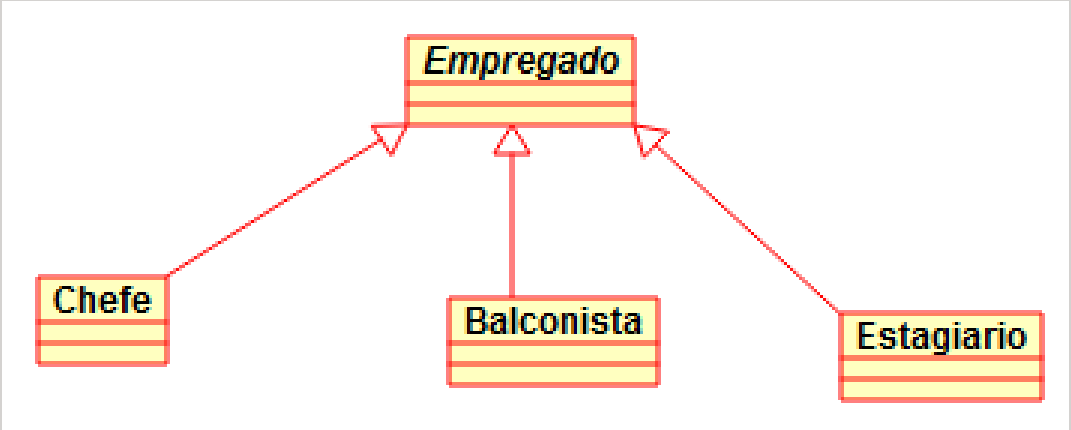


Figura: Classe Empregado é classe Abstract.
Subclasses folha não deveriam ser herdadas (final).

Classes Abstratas

- Modificador de acesso ‘final’:
  - Classes ‘final’ não podem ser herdadas.
  - Métodos ‘final’ não podem ser sobrescritos.
  - Variáveis de instância ‘final’ são herdadas, mas não podem ser modificadas.

Revisão

Revisão

- Interfaces
- Classes Abstratas

Exercícios

<Ver conteúdo na plataforma de ensino>

Referências

- Referências bibliográficas da disciplina.