UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Programação Orientada a Objetos

1

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROF. LUCIO AGOSTINHO ROCHA

AULA 7: HERANÇA

2º.SEMESTRE 2023

Herança

Herança

- Herança:
  - Reusabilidade de Software
  - Novas classes de objetos aproveitam características de classes já existentes.
    - Aproveitamento de Atributos e Métodos
    - Adição de novas funcionalidades
    - Exemplo: Classe Sapo herda da Classe Animal

Herança

- Herança:
  - Generalização x Especialização
  - Subclasse herda da Superclasse
    - Subclasse: adiciona novas variáveis de instância e métodos
  - Java não suporta herança múltipla, porém admite múltiplas Interfaces
  - Herança: define um relacionamento do tipo “é um”:
    - Ex.: Calculadora é um Equipamento
  - Composição: define um relacionamento do tipo “tem um”:
    - Ex.: Calculadora tem um Visor.

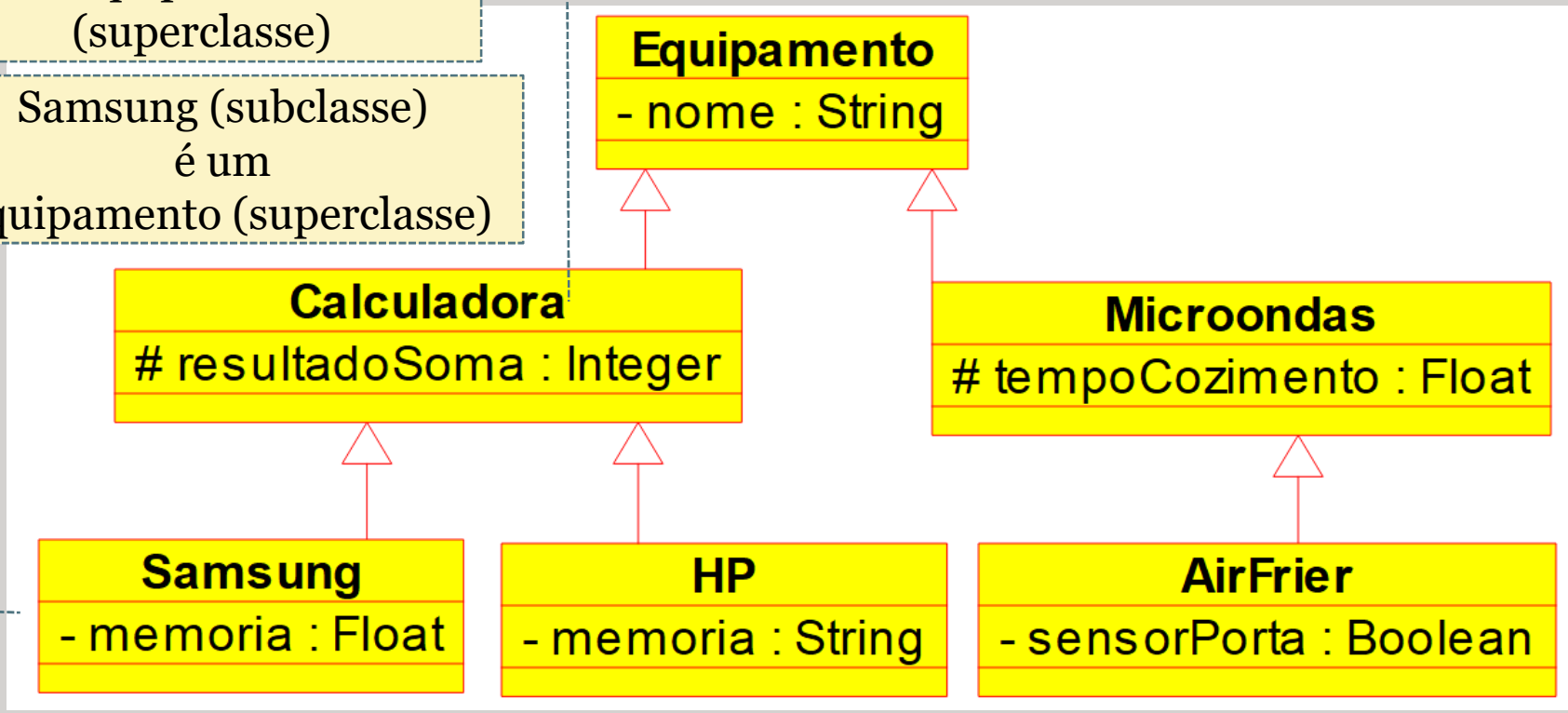
Herança

- Herança: relacionamento “é um”:
  - Objeto “é um” objeto de outra classe
  - Herança define uma hierarquia de árvore.
    - Nota: Herança múltipla é fonte de problemas.
    - Nota: Hierarquia em árvore remove o problema de loops sintáticos e semânticos.

Herança

Calculadora (subclasse)
é um
Equipamento
(superclasse)

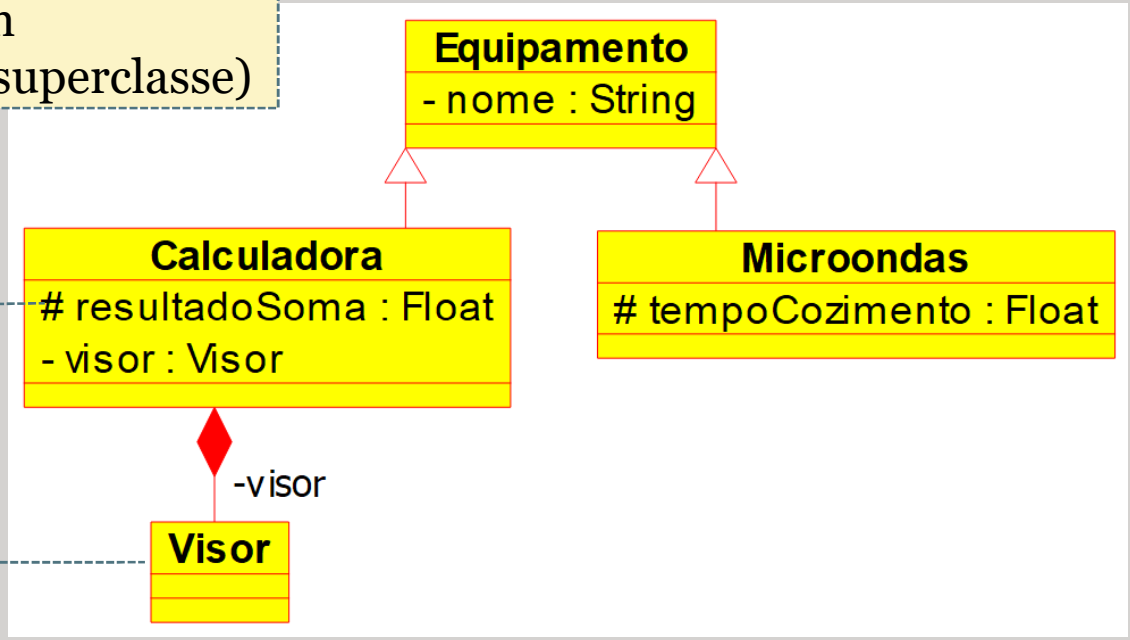
Samsung (subclasse)
é um
Equipamento (superclasse)



- Figura - Exemplo de Relacionamento de Herança.

Herança

Calculadora (subclasse)
é um
Equipamento (superclasse)



- Figura - Exemplo de Relacionamento de Herança.

Calculadora (classe)
tem um
Visor (classe)

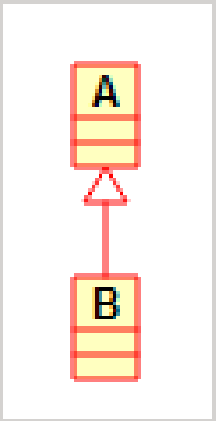
Herança

- Herança:
  - Ocultamento de informação:
    - Objetos não sabem como outros objetos são implementados.
      - Ex.: é possível usar uma calculadora sem saber como ele calcula os números.
  - Abstração:
    - Pensar em termos de propriedades comuns a vários objetos.
      - Calculadora (Classe) é um tipo de Equipamento (Classe)
      - AirFrier (Classe) é um tipo de Equipamento (Classe)
      - Celular (Classe) é um tipo de Equipamentp (Classe)

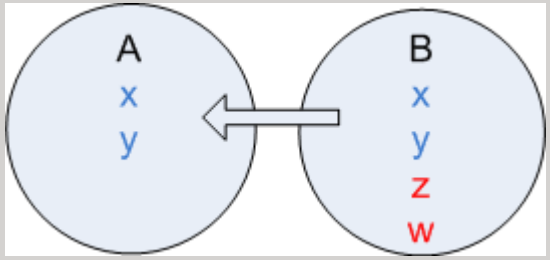
Herança

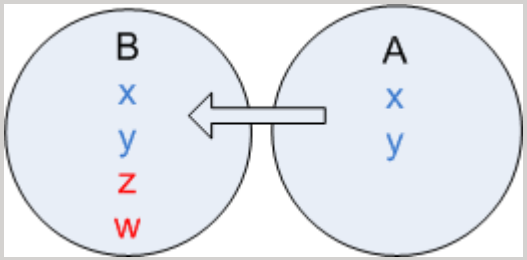
- Objeto da Subclasse (mais especializado) pode ser tratado como um objeto da Superclasse (menos especializado)
  - Subclasse contém todos os Membros não-privados da Superclasse
  - Ex.: Samsung é uma Calculadora E Calculadora é um Equipamento
- Objeto da Superclasse (menos especializado) não podem ser diretamente tratados como um objeto da Subclasse (mais especializado).
  - Há perda de informação.
  - Ex.:
    - Equipamento é uma Calculadora X
    - Equipamento é uma HP X

Herança



- B é um A

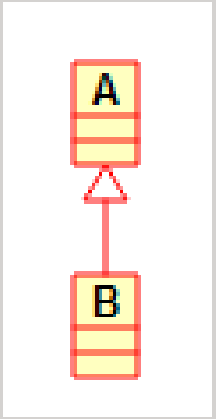




- Faltam campos

Herança

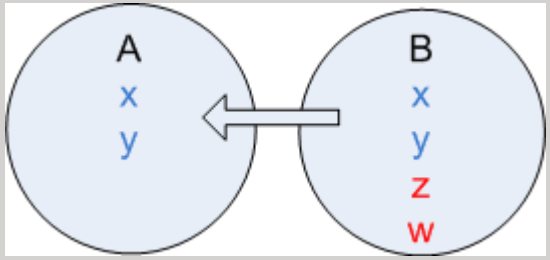

```
public class A {
   public A( ){  }
}
public class B extends A{
   public B( ) {  }
   public void metodoB(){
      return "B";
   }
}
```

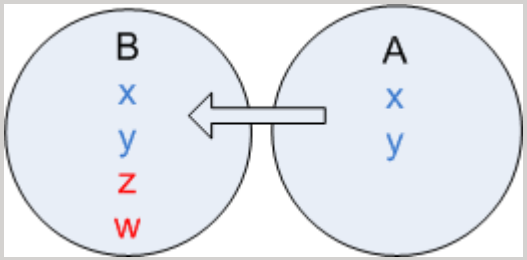


- B é um A

```
public class C {
    A a = new A ( );
    B b = new  B( );
    a = b;        //'a' continua sendo 'A'
    ((B) a).metodoB();

    b = (B) a; //downcast
                  //(necessário a=b)
}
```





- Lado direito deve, pelo menos, possuir todos os campos do lado esquerdo.

- Faltam campos

- Lado direito deve, pelo menos, possuir todos os campos do lado esquerdo.

Herança

- Construtor:
  - Construtores nunca são herdados.
  - É a primeira instrução invocada pela classe.
  - Há chamadas sucessivas na hierarquia para inicializar as variáveis de instância de cada classe

Herança

- Garbage Collector:
  - Método finalize da subclasse deve invocar o método finalize da superclasse com sobrecarga.
  - Caso contrário, apenas a subclasse será indicada para Garbage Collector.
  - Portanto, é uma boa prática de programação sempre incluir um método finalize na superclasse.

Revisão

Revisão

- Herança:
  - É um mecanismo nativo de linguagens de Programação Orientadas a Objetos que permite o reaproveitamento de código.
  - Herança melhora a legibilidade do código e auxilia na organização da estrutura do projeto.

Exercícios

<Ver conteúdo na Plataforma de Ensino>

Referências

- Referências bibliográficas da disciplina.