Herança

É o relacionamento entre classes em que uma classe chamada de subclasse (classe filha, classe derivada) é uma extensão, um subtipo, de outra classe chamada de superclasse (classe pai, classe mãe, classe base). Devido a isto, a subclasse consegue reaproveitar os atributos e métodos dela. Além dos que venham a ser herdados, a subclasse pode definir seus próprios membros.

Códigos:

Java

class A extends B {

...

}

C#

class A : B {

...

}

Python

class A(B):

...

Características

Tipos de herança:

* Simples - A classe filha tem só uma classe mãe

Funcionário (classe mãe)

Gerente Vendedor Faxineiro (classes filhas)

* Múltipla - A classe filha tem uma ou mais classes mães

Estudante (classes mães) Funcionário

Estagiário (classe filha)

Java nem C# não possui herança múltipla

C++

Python

Class A(B,C):

Pass

Upcast

Java

A a = new B();

C#

A a = new B();

Python

X

Downcast(Operação perigosa)

Java

B a = (B) new A();

C#

B a = (B) new A();

Python

X

Polimorfismo

"A mesma ação, se comportando diferente.”

Possibilidade de um mesmo método se comportar diferente, flexibilidade de uma mesma ação se comportando diferente.

Sobrescrita

"A mesma ação, podendo se comportar diferente. "

Associação

"Possibilita um relacionamento entre classes/objetos, no qual estes possam pedir ajuda a outras classes/objetos e representar de forma completa o conceito ao qual se destinam. Neste tipo de relacionamento, as classes e os objetos interagem entre si para atingir seus objetivos”.

Tipos:

* Estrutural

-Composição: Comportamento “Com Parte Todo”

Ex:

Pessoa 🡪 Endereço

Endereço

A parte só existe com o todo.

Java

class Pessoa {

Endereco endereco

}

-Agregação: Oposto “Sem Parte Todo”

Ex:

Disciplina -> Aluno

Aluno aluno

Java

Class Disciplina {

Aluno aluno;

}

* Comportamental

- Dependência: Comportamental “Depende de”

Ex:

Compra

Cliente cliente -> Cupom

finalizar(Cupom cupom,...)

finalizar(...)

class Compra {

...

finalizar (Cupom, cupom ....) {

...

}

}

Herança x Associação

Herança é uma relação mais rígida, classe herdada a outra, isso

significa que quando a aplicação começar a rodar não vai mudar. Já associação é um pouco mais flexível pois pode mudar durante o programa.

Como usar:

Herança

É um subtipo da outra?

Sim

Associação

É um subtipo da outra?

Não

Então ela vai usar a outra, ou seja deve utilizar associação.

Interface

"Define um contrato que deve ser seguido pela classe que a implementa. Quando uma classe implementa uma interface, ela se compromete a realizar todos os comportamentos que a interface disponibiliza."

Códigos:

Java

interface A {

...

}

class B implements A {

...

}

C#

interface A {

...

}

class B : A {

...

}