



Composição x Herança



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

Contextualização

2

Herança e Composição

3

**Vantagens e
Desvantagens**

4

Quando usar?

1| Contextualização



1. Contextualização

Vamos começar pelo básico de entender o que é uma herança e o que é uma composição.





1. Contextualização (cont)

Falando de forma geral, uma definição para as duas seriam dois mecanismos para reutilizar funcionalidades, ou seja, não ser repetitivo ou escrever um código desnecessário.



2l Herança e Composição



Herança

O que é?

Herança sempre foi considerada uma ferramenta básica de extensão e reuso da funcionalidade, baseada nos atributos e métodos de uma classe





Herança

{

```
public class Animal{  
}  
public class Cachorro extends Animal{  
}  
public class Gato extends Animal{  
}
```





Composição

O que é?

Na composição ao invés de codificarmos um comportamento estaticamente, como é feito com o uso de herança, definimos pequenos comportamentos padrão e usamos composição para definir comportamentos mais complexos.





Composição

Exemplo:

```
{  
    public class Sistema {  
  
        Pessoa pessoa = new Pessoa();  
  
    }  
    public class Pessoa {  
    }  
}
```

3l Vantagens e Desvantagens



Vantagens - Herança

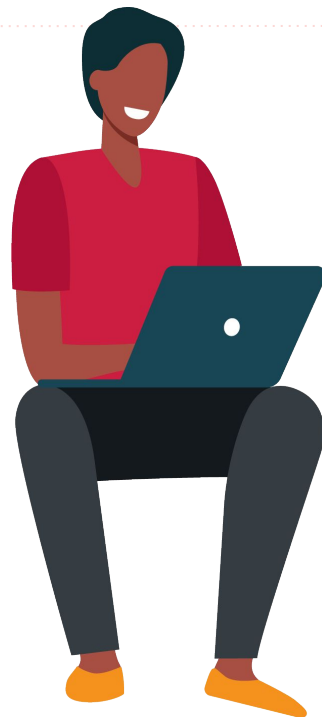
- Um dos benefícios da Herança é que ela captura o que é comum e o isola daquilo que é diferente. Além disso, a herança é vista diretamente no código até mesmo devido a sua natureza estática.
- A herança, então, permite criar uma estrutura hierárquica de classes cada vez mais especializadas. A grande vantagem disso é não ter que partir do zero para especializar uma classe existente.





Desvantagens - Herança

- Entre os problemas da Herança está o fraco encapsulamento e o forte acoplamento onde ao mudar uma superclasse pode afetar todas as subclasses
- Além disso, algumas vezes um objeto precisa ser de uma classe diferente em momentos diferentes o que não é possível com a herança, por ter uma relação estática.





Vantagens - Composição

- A grande vantagem da Composição é que o comportamento pode ser escolhido em tempo de execução em vez de estar amarrado em tempo de compilação.
- Além disso, os objetos que foram instanciados e estão contidos na classe que os instanciou são acessados somente através de sua interface seguindo assim o princípio de programar para uma interface e não para uma implementação.





Desvantagens - Composição

- A grande desvantagem é que um software muito dinâmico e parametrizado é mais difícil de entender do que software mais estático.



4l Quando usar?



Quando usar?

- De forma geral prefira utilizar sempre a Composição em relação à Herança. No entanto, pode-se definir algumas regras para identificar quando podemos usar a Herança de forma que não tenhamos os problemas que ela acarreta.





Quando usar?

1. **Utiliza-se a herança se uma instância de uma classe Filha nunca precisar tornar-se um objeto de outra classe.**





Quando usar?

1. **Utiliza-se a herança se uma instância de uma classe Filha nunca precisar tornar-se um objeto de outra classe.**
2. **Se a hierarquia de herança representar um relacionamento "É um" e não um relacionamento "Tem um".**





Quando usar?

1. **Utiliza-se a herança se uma instância de uma classe Filha nunca precisar tornar-se um objeto de outra classe.**
2. **Se a hierarquia de herança representar um relacionamento "É um" e não um relacionamento "Tem um".**
3. **Se deseja-se ou precisa-se realizar alterações globais para as suas classes filhas alterando uma classe Pai.**





Quando usar?

1. **Utiliza-se a herança se uma instância de uma classe Filha nunca precisar tornar-se um objeto de outra classe.**
2. **Se a hierarquia de herança representar um relacionamento "É um" e não um relacionamento "Tem um".**
3. **Se deseja-se ou precisa-se realizar alterações globais para as suas classes filhas alterando uma classe Pai.**
4. **Ou então quando a classe filha estender ao invés de substituir total ou parcialmente as responsabilidades da classe Pai.**



DigitalHouse>
Coding School