

## Para saber mais: Herança

Até aqui, já tínhamos visto dois dos quatro pilares da orientação a objetos:

1. **Abstração**
2. **Encapsulamento**

Agora estamos começando a falar do 3º pilar: **Herança**.

Uma das formas de reutilizar código é fazendo uma classe estender outra, ou seja, herdar tudo que uma classe base tem, e adicionar comportamentos específicos.

Mas vale ressaltar que herança não é a solução para todos os problemas de reutilização de código. Muitas vezes pode até ser prejudicial.

Talvez ainda não fique 100% claro, mas vale a pena a leitura deste artigo: [Como não aprender orientação a objetos: Herança](https://blog.caelum.com.br/como-nao-aprender-orientacao-a-objetos-heranca/) (<https://blog.caelum.com.br/como-nao-aprender-orientacao-a-objetos-heranca/>).

Caso não faça sentido ainda, salve este link em seus favoritos e releia no final do curso. Depois, com alguns meses de experiência com orientação a objetos, releia mais uma vez. Garanto que fará cada vez mais sentido a cada leitura.