

06

Para saber mais: Delegation

Ao implementarmos a autenticação na classe `Conta`, reutilizamos o código do titular do tipo `Cliente` delegando a responsabilidade de implementação.

Trata-se de um padrão chamado [delegation](https://en.wikipedia.org/wiki/Delegation_pattern) (https://en.wikipedia.org/wiki/Delegation_pattern), e existe uma maneira mais objetiva de implementá-lo no Kotlin utilizando a [delegação de propriedade \(Property delegation\)](https://en.wikipedia.org/wiki/Property_delegation) (https://en.wikipedia.org/wiki/Property_delegation).

Considerando o exemplo visto em vídeo:

```
abstract class Conta(
    var titular: Cliente,
    val numero: Int
) : Autenticavel {

    // restante do código

    override fun autentica(senha: Int): Boolean {
        return titular.autentica(senha)
    }
}
```

Temos o seguinte resultado com a delegação de propriedade:

```
abstract class Conta(
    val titular: Cliente,
    val numero: Int
) : Autenticavel by titular
```

Com apenas a syntax `Autenticavel by titular`, é feita a delegação de implementação, como fizemos. A restrição nesse caso é que precisamos operar com properties `val`, pois não é possível modificar a implementação depois de criarmos uma conta.