



Para saber mais - Utilização do lateinit

A inicialização por meio do `lateinit` é bem útil em situações pelas quais precisamos inicializar properties em algum momento não determinado...

Porém, considerando alguns pontos que vimos, ela tende a ser problemática justamente por ainda manter a responsabilidade do programador de lembrar da inicialização, como também, fazer com que lidemos com properties `var`. Em outras palavras, considerando apenas a nossa situação atual, não há um grande benefício ao considerá-la, certo?

Motivo real para considerar o uso do lateinit

Entretanto, a proposta do Kotlin sobre a utilização desse tipo de inicialização é permitir também um recurso muito comum em programação conhecido como injeção de dependência que, basicamente, faz com que a aplicação se responsabilize pela inicialização de properties, ou seja, para esses casos, o `lateinit` se faz necessário.

Posts e artigos sobre o assunto

Caso tenha dúvidas sobre injeção de dependência, deixo a referência de um [post no blog da Caelum](http://blog.caelum.com.br/ioc-e-di-para-frameworks-mvc/) (<http://blog.caelum.com.br/ioc-e-di-para-frameworks-mvc/>), e também, de um [artigo que escrevi sobre CRUD API com Spring Boot](https://medium.com/collabcode/implementando-uma-crud-api-no-spring-boot-com-kotlin-parte-1-c6e281d0f8f8) (<https://medium.com/collabcode/implementando-uma-crud-api-no-spring-boot-com-kotlin-parte-1-c6e281d0f8f8>).