07

Cuidados com delegação por lazy

Transcrição

[00:00] Continuando com essa abordagem, da property delegada, como a gente viu aqui no nosso "lazy", tem alguns cuidados que a gente precisa tomar, também, quando está utilizando essa abordagem.

[00:07] Porque, por mais que a gente tenha conseguido fazer aquele esquema de inicializar a nossa variável apenas quando ela fosse necessária, como é o caso, quando a gente está criando aqui a nossa tela, a gente também tem alguns riscos para a gente fazer isso. A gente vai entender agora como são esses ricos e os cuidados que a gente precisa tomar.

[00:24] Qual seria esse risco que a gente toma? Vamos fazer o seguinte, aqui a gente está inicializando nossa "viewDaActivity" e, por enquanto, o nosso primeiro uso, está sendo feito, apenas quando a gente cria a nossa "view", que a gente chama o "setContentView". O que acontece se a gente, simplesmente, chegar aqui fazer um "private val_outraViewDaActivity" vamos criar uma outra view da Activity e, nessa outra view da Activity, a gente vai inicializar por meio de quem? Por meio da nossa property que foi delegada, o que acontece?

[00:57] Vamos dar uma olhada? Vamos executar aqui, Alt+Shift+F10, vamos ver o que aconteceu na nossa App, vamos subir aqui. A gente teve o crash, porque que a gente teve o crash? vamos voltar aqui, deixa eu só tirar esse filtro que eu tinha colocado, e vamos ver o que acontece aqui no crash. O nosso crash que a gente teve, novamente, o Null Pointer Exception, porque a gente esta tentando pegar uma "decorView", e olha o momento que ele fala que a gente teve esse problema, vamos ver aqui, foi dentro do nosso "lazy".

[01:30] Por mais que o "lazy" tenha a capacidade de tentar nos proteger dessa inicialização que a gente não quer fazer naquele momento, ainda assim a gente corre alguns riscos se a gente não tomar alguns cuidados, que é, justamente, tentar utilizar o "lazy" no momento que ele não pode ser utilizado, porque a inicialização dele não pode ser feita naquele momento. Então, considerando esse aspecto, a gente tem que tomar bastante cuidado quando a gente utilizar o "lazy".

[01:51] Ele não vai dar a solução perfeita para gente, 100%, a gente tem que estar atento a isso. Portanto, o que eu quero passar para vocês, o que eu quero passar para vocês é o seguinte, se, por exemplo, agora a gente tá querendo utilizar essa variável aqui, a "outraViewDaActivity", por meio de uma inicialização que está sendo feita via "lazy", a gente também vai precisar, por exemplo, utilizar aqui o "by lazy". A gente vai, também, ter que delegar essa responsabilidade.

[02:17] Porque a gente vai ter que fazer isso, se caso, a gente tem essa necessidade, por exemplo? Porque, a gente só vai utilizar essa "outraViewDaActivity " no momento em que, realmente, a gente tiver acesso à nossa "view". Por exemplo, se eu chegar aqui para vocês, deixa eu até apagar esses logs aqui de baixo, se eu chegar aqui para vocês, agora, e colocar um log que eu estou utilizando essa outra view.

[02:40] "teste lazy 5", que é um novo, a gente vai colocar agora essa "outra view", e aqui eu vou fazer até o "toString" dela, usando o String template, "outraViewDaActivity.toString". Vamos ver o que acontece, até completou para a gente, deixa eu só pegar aqui, ele já está falando que não precisa, aqui já imprime para a gente valor dela, a gente não precisa, eu coloquei desnecessariamente. Eu só vou fazer esse teste aqui para vocês verem e vamos ver o que acontece nessa parte do teste do "lazy", vamos lá.

[03:11] Olha o que acontece. A gente faz o "by lazy", vamos ver se ainda quebra a nossa App. Veja que o Android Studio executou, e ele volta a funcionar, deixa eu subir aqui de novo, ele tá funcionando normalmente. Se a gente tentar colocar aqui as transações, de despesa, por exemplo, ele ainda tá funcionando, porque ele funcionou?

[03:25] Vamos voltar aqui no Logcat, vamos colocar aqui o "teste" e, novamente, ele fez a inicialização do nosso "lazy" nesse momento que a gente chamou essa outra variável, se a gente tentar reutilizar essa nossa property, aqui, a "viewDaActivity", por exemplo, para poder colocar em alguma outra variável, porque vai usar ela como referência, a gente vai precisar para usar o "lazy", também, para poder garantir esse comportamento que a gente espera, que é inicializar, no momento que realmente faz sentido inicializar.

[03:52] No momento em que a gente já tem uma view criada, que, no caso, chamando o "setContentView". Essa é uma abordagem que a gente precisa estar tomando cuidado, quando a gente vai utilizar a delegação de property, por meio do "lazy", porque a gente não pode inicializar aquela variável, naquele momento, esse é o cuidado que a gente precisa tomar. E aproveitando essa abordagem, a gente pode até mesmo fazer uma refaturação no nosso código.

[04:13] Por exemplo, reparem que a gente, inicialmente, começamos com essa proposta de pegar a "viewDaActivity", para poder reutilizá-la dentro do nosso código. Portanto, existe alguns códigos, que se a gente estiver reutilizando ela, a gente está fazendo um cast, o que significa isso? Significa que a gente pode tentar fazer, também, um cast dentro de uma property para gente. Para a gente, também, reutilizar esse cast aqui, que a gente está fazendo, toda vez que a gente está querendo mandar uma ViewGroup, por exemplo.

[04:39] Em dois momentos, a gente pode, também, fazer isso, ao invés de deixar, por exemplo, "outraViewDaActivity ", a gente pode colocar como "viewGroupDaActivity". E aqui dentro, a gente pode fazer, por exemplo, o nosso cast, como uma "as ViewGroup". Agora, a gente pode reutilizar essa property, agora que ela está sendo feita via "lazy" também, que ela só vai carregar, apenas quando ela for chamada e, no momento que ela for chamada a gente já tem, meio que, a certeza, de que, a viu já está pronta, assim, a gente pode utilizar ela.

[05:16] Eu vou lá, e vou colocar ela aqui para gente, eu vou lá novamente, e coloco ela aqui também. Então, agora, a gente já tá conseguindo utilizar as nossas properties, dentro de cada um dos membros, que estavam reutilizando a parte do "window.decorView". A gente estava utilizando dessa maneira e, agora, a gente conseguiu colocar elas como properties, porque, antes, a gente tinha que ficar chamando todas essas properties da própria Activity, estava ficando um pouquinho verboso.

[05:42] Agora a gente já criou a "viewDaActivity", que é o que a gente precisa, como, também, a "viewGroupDaActivity", que são outras classes que precisam dela também. A gente conseguiu finalizar essa parte da refaturação, inclusive, posso até apagar essa parte da inicialização, do teste de inicialização, e a gente percebe que, agora, nosso código ficou bem mais fácil de utilizar. Vamos até executar novamente para ver se tudo está funcionando, Alt+Shift+F10, veja que o Android Studio conseguiu.

[06:08] Vamos ver aqui se está tudo funcionando como esperado, primeiro vou adicionar uma receita aqui de R$ 100, aqui eu vou alterar só para ver se tá funcionando também, vou colocar uma data qualquer, não vou nem selecionar muitas informações. Ele tá funcionando, da mesma maneira, vamos testar a despesa, uma despesa de R$ 60, mudar uma data, só para ter certeza, mudar a categoria, "Comunicações", ele já está funcionando. A gente conseguiu fazer essa refaturação, agora utilizando essa delegação preguiçosa, delegação por "lazy".

[06:38] A gente viu as peculiaridades que a gente tem que ter, quando a gente está utilizando essa abordagem, então até mais!