09

Conclusão

Transcrição

Chegamos ao final do curso de Android com Kotlin! Parabéns!

Vamos passar por todos os pontos do projeto para revisarmos o que aprendemos. Inicialmente, vimos que para criarmos um projeto no Kotlin usamos o Android Studio na versão `3.0`.

Ao criarmos este projeto, todos os arquivos de configuração foram criados automaticamente, sem que tivessemos que fazê-lo manualmente. Vimos como criar uma *activity* e que, para implementarmos uma classe no Kotlin, em vez de *extends* usamos os dois pontos (`:`), incluindo em seguida a classe de que queremos a herança.

Criamos uma função de maneira similiar à maneira como fazemos em Java, com a herança, a partir da qual pedimos as funções, e aí se usa o `override`, como no Java.

Além disso, aprendemos que o Kotlin permite uma maneira bem mais objetiva de acessarmos componentes, com o *synthetic*, em vez do `findViewById()`.

Nesta linguagem também há o conceito de *properties*. No *adapter*, por exemplo, vimos que precisamos pegá-las para utilizá-las nas funções, como é o caso de `transacoes e context`.

As *properties*, diferentemente dos atributos, por padrão nos permitem acesso pelos *getters* e *setters*, isto é, não acessamos seus valores diretamente, por mais que pareça que sim.

Também vimos as diferenças ao utilizarmos uma lista, uma vez que usamos um operador similar ao *array* para pegarmos um elemento, sendo que na verdade, por debaixo dos panos, ele está chamando o *get*.

Começamos a criar nossos modelos, "Tipo" (com a criação de `enum`), e "Transacao", de maneira bem sucinta, sem a necessidade de criação do corpo da classe para definirmos as *properties*.

Também conseguimos fazer com que os valores fossem padrões, sem a obrigação de envio de uma categoria para podermos colocá-la. Caso queiramos mudá-la, aí sim, poderemos fazer isto.

Há duas opções ao utilizarmos uma *property*, uma variável: `var e val`, escolhidas de acordo com a necessidade de mutabilidade de seus valores. Segundo as boas práticas relacionadas a isso, é recomendado manter-se o `val`, trocando-o somente se houver necessidade de alteração de valor.

Na parte de criação de modelos, aprendemos que estávamos usando *limited parameters* na *activity*. Agora, não precisamos mais nos preocupar com a ordem de envio dos parâmetros no construtor ao construirmos objetos, e podemos enviar os parâmetros desejados.

Algo bem diferente do que estamos acostumados quando trabalhamos com Java são as extensões das classes. Agora, somos capazes de atribuir novos comportamentos em alguma classe.

Implementamos uma extensão para o `BigDecimal`, para o formato do padrão da moeda brasileira, por exemplo. Fizemos algo similar para o formato de datas no `Calendar`, e também para a `String`, formatando as categorias das transações no projeto para um limite de caracteres desejado.

O aspecto visual final a que chegamos é uma lista com as transações e suas devidas indicações de cor e ícones correspondentes, bem próximo ao projeto desenvolvido em Java.

Espero que tenham aproveitado bastante o conteúdo do curso, e na próxima parte veremos como melhorar ainda mais a nossa app. Até mais!