

Adicionando a transação do usuário

Agora que somos capazes de criar transações a partir do dialog, vamos dar início ao processo para adicionar a transação criada na lista de transações. Processo que também vai atualizar o resumo.

Modificando a referência da lista de transações

O primeiro passo, será modificar a referência da lista que atualmente é feita a partir da função `transacoesDeExemplo()`, que nos devolve uma lista de transações qualquer. Em outras palavras, remova essa função.

Em seguida, crie a property `transacoes` do tipo `List<Transacao>` para que compute as nossas transações. Então, para a inicialização da mesma, utilize a função `listOf()`.

Com essa modificação, todas as funções membros da `ListaTransacaoActivity` tem acesso à mesma lista, portanto, não precisam mais receber uma lista via parâmetro. Sendo assim, remova todos os parâmetros que recebem uma lista de transação das funções da Activity.

Utilizando collection mutável

Por mais que o código esteja compilando, se tentarmos adicionar uma transação na property `transacoes` com a função `add()`, o compilador reclama indicando que não existe essa função. Isso acontece, porque o Kotlin segue a ideia de que as collections são imutáveis.

Ou seja, modifique a interface da lista imutável `List<Transacao>` para a mutável `MutableList<Transacao>`, então, initialize a mesma com a função `mutableListOf()`.

Adicionando transações

Agora que temos uma lista mutável, logo após criar a transação a partir do dialog, utilize a property `transacoes` e chame a função `add()` enviando a transação criada.

Atualizando os componentes

Após adicionar a transação na property `transacoes`, chame a função `configuraLista()` e `configuraResumo()` para que atualize tanto a lista como o resumo.

Boa prática

Veja que todas essas chamadas para adicionar a transação na property `transacoes`, como também, atualizar os componentes necessários, no final das contas tem o único objetivo, que é **atualizar as transações**.

Sendo assim, extraia a função `atualizaTransacoes()` enviando todo o código que responsável em atualizar as transações.

Para essa extração, recomendo a utilização do AS por meio do atalho **Ctrl + Alt + M** ou no Mac **CMD + Alt + M**.

Por fim, execute a App e veja se ao adicionar uma transação, a lista e o resumo são atualizados de acordo com as informações que foram computadas na transação.