02

Criando a transação pelo dialog

Caso tenha a necessidade de pegar o projeto com todas as alterações que foram feita na aula passada, fique à vontade em baixá-lo por meio deste link (https://github.com/alura-cursos/financas-kotlin-parte2/archive/aula-4.zip).

Agora que implementamos toda parte visual do nosso dialog, precisamos começar com o processo que vai pegar todas as informações que o usuário enviar e criar uma transação.

Implementando o listener do botão positivo

Para isso, o nosso primeiro passo é implementar o listener no botão positivo, portanto, implemente a interface `DialogInterface.OnClickListener` no segundo parâmetro da função `setPositiveButton()` utilizando a expressão lambda.

Pegando os valores dos componentes

Dentro da expressão lambda, pegue todos os campos dos componentes e atribua para variáveis `val` .

No caso do componente de valor e data que são `EditText` , basta apenas pedir o `toString()` da property `text` .

Entretanto, no componente da categoria, utilize a property `selectedItem` e chame o `toString()` dela.

Repare que todos os campos se tratam de Strings, sendo que para criar uma transação precisamos, por exemplo, que a variável que representa o valor seja do tipo `BigDecimal` e a data `Calendar` .

Convertendo o campo valor

Para converter a String em `BigDecimal` , crie uma instância de `BigDecimal` enviando via construtor a variável que está representando o valor em String, então, atribua para um objeto `val` .

Convertendo o campo data

A conversão do campo data é um pouco mais trabalhosa. Primeiro é preciso criar um formatador por meio da classe `SimpleDateFormat` , durante a instância da mesma, envie como parâmetro do construtor o seguinte padrão: `"dd/MM/yyyy"` .

Em seguida, atribua para um objeto que vai representar o formato brasileiro, então, com esse objeto, chame a função `parse()` e envie a variável que está com o valor da data em String e atribua para um objeto.

Repare que essa atribuição devolve um objeto do tipo `Date` que não é compatível com a nossa API de data que é a `Calendar` .

Inserindo um Date na API Calendar

Entretanto, podemos inserir a data computada dentro de um objeto `Date` num `Calendar` . Portanto, crie um objeto do tipo `Calendar` a partir da função `getInstance()` , então, atribua o objeto do tipo `Date` na property `time` .

Criando a transação

Após obter todos os valores necessários para criar uma transação, faça uma instância da classe `Transacao` enviando todos os parâmetros, a única diferença é que a property `tipo` vai ser fixa, ou seja, do tipo receita `Tipo.RECEITA`.

Por fim, imprima os valores da transação em um Toast para testar se funcionou a criação da mesma.