08

Adicionando a soma das despesas

Agora que refatoramos o código que vai ficar responsável em adicionar os valores do resumo, crie a função `adicionaDespesa()` dentro da classe `ResumoView` e implemente o código para somar as transações de despesa e adicionar o valor da soma.

Para facilitar o processo, copie e cole o código da função `adicionaReceitaNoResumo()` e modifique todo o código que referencia as transações de receitas. Exemplo: deixe o nome da variável `totalDespesa`, a comparação para o `Tipo.DESPESA` e o componente para `resumo_card_despesa`.

Aproveite também para renomear a função `adicionaReceitaNoResumo()` para `adicionaReceita()`, pois, já que estamos dentro da classe `ResumoView`, é subentendido que iremos adicionar o valor para o resumo.

Adicionando as transações como uma property

Em seguida, repara que ambas as funções recebem uma lista de transações sendo que a ideia é utilizar a mesma referência. Envie a lista de transações via construtor primário na classe `ResumoView` e remova o parâmetro `transacoes: List<Transacao>` tanto da função `adicionaReceita()` como da `adicionaDespesa()`.

Modificando a chamada dentro da activity

Agora, na Activity, selecione todo o código que faz uso da `ResumoView` e extraia a função `configuraResumo()` (tente utilizar a extração de função do AS).

Em seguida, modifique a instância da classe `ResumoView` para que receba a `view` e, em seguida, a lista de transações. Depois, atribua a instância para um objeto.

Com o objeto em mãos, chame a função `adicionaReceita()` e depois a função `adicionaDespesa()`.

Por fim, execute a App e veja se a despesa aparece da mesma forma como apareceu a receita.