

## Extraíndo a superclass para o dialog

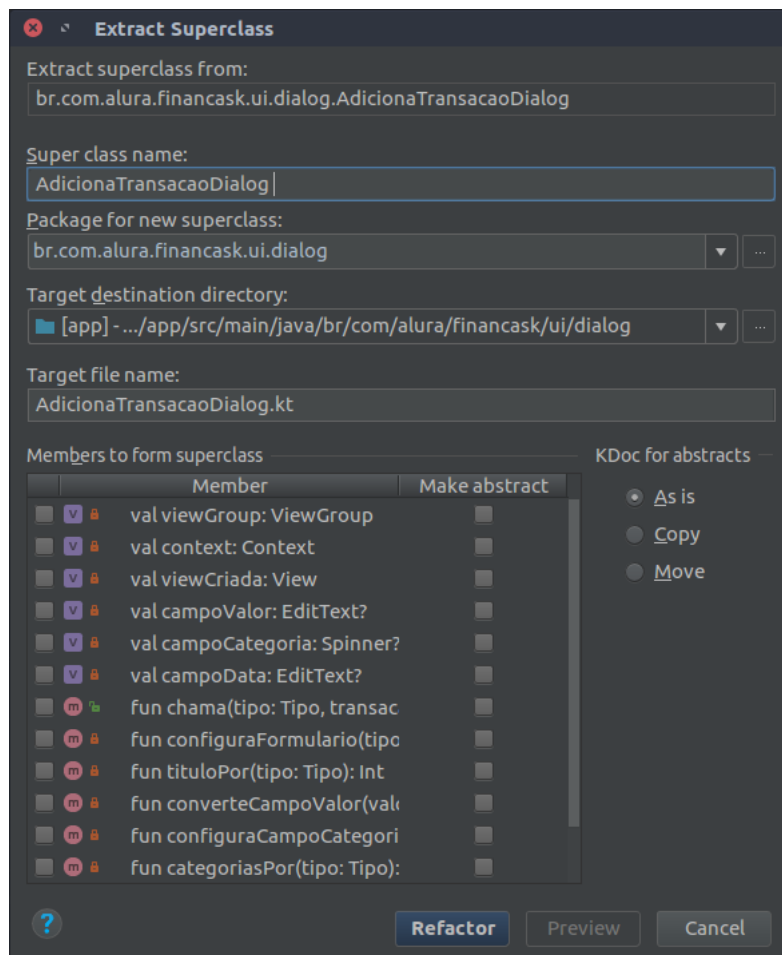
Caso você precise do projeto com todas as modificações que foram realizadas na aula anterior, fique à vontade em baixá-lo por meio [deste link \(https://github.com/alura-cursos/financas-kotlin-parte3/archive/aula-3.zip\)](https://github.com/alura-cursos/financas-kotlin-parte3/archive/aula-3.zip).

Agora que finalizamos a refatoração da Activity, vimos que as nossas classes para os dialogs apresentam comportamentos bem similares. Sendo assim, podemos aplicar a técnica de herança que aprendemos em OO.

### Criando a superclass

Como ponto de partida, vamos até a classe `AdicionaTransacaoDialog`. Utilize o atalho `Ctrl + Alt + Shift + T` para apresentar a funcionalidade **Refactor This** ou vá ao menu **Refactor > Refactor This...**

Quando surgir a janela de opções, vá na opção **Superclass...** tecle Enter ou clique nela. Escolha a opção **Extract to separate file** veja que vai abrir a seguinte janela:



Nesta janela, indique que o nome da superclass é `FormularioTransacaoDialog` e mantenha o mesmo pacote, enviando todos os membros da `AdicionaTransacaoDialog` para a superclass.

### Resolvendo as referências e imports

Repare que nesta modificação a superclass apresenta alguns problemas de compilação, por não conseguir resolver algumas referências em imports.

Sendo assim, primeiro resolva os imports necessários, em seguida, resolva as seguintes referências recebendo-as via construtor primário:

- `context` : property do tipo `Context` .
- `viewGroup` : property do tipo `ViewGroup` .

Após resolver esses problemas de compilação, precisamos ajustar a maneira como a classe filha ficou.

## Resolvendo os detalhes da classe filha

Dentro da `AdicionaTransacaoDialog` otimize os imports utilizando o atalho **Ctrl + Alt + O** ou no menu **Code > Optimize imports**.

Veja que a superclass exige o envio dos parâmetro para o `Context` e `ViewGroup` , então envie no construtor da própria superclass. Veja que a `AdicionaTransacaoDialog` não precisa mais das properties `context` e `viewGroup` , então apenas receba-as como parâmetros do construtor primário.

Outro detalhe é que a classe `AdicionaTransacaoDialog` não precisa mais do corpo, pois não possui mais nenhum membro dentro dela. Sendo assim, apague também as chaves.

## Testando a App

Com esses ajustes, tente executar a App e veja se é apresentado o mesmo resultado de antes. Lembrando que não precisa modificar a chamada da `AdicionaTransacaoDialog` na Activity.