

 08

Sobre herança no Kotlin

Para reutilizar o código entre as classes de dialog, criamos uma superclass responsável em manter todo o código comum entre os dialogs de adição e alteração. Mas para que fosse possível aplicar o conceito de herança, ocorreram algumas mudanças. Considerando esse contexto, marque as alternativas que correspondem aos detalhes vistos sobre herança no Kotlin:

Seleccione 3 alternativas

A Não podemos utilizar o modificador de acesso privado para membros abstratos.

B Por padrão, não podemos herdar de qualquer classe do Kotlin.

C Superclasses precisam ser abstrata para que outras classes se estendam a partir dela.

D Para estender uma classe do Kotlin é necessário abrí-la.

E Os membros de uma superclass não podem ser privados.