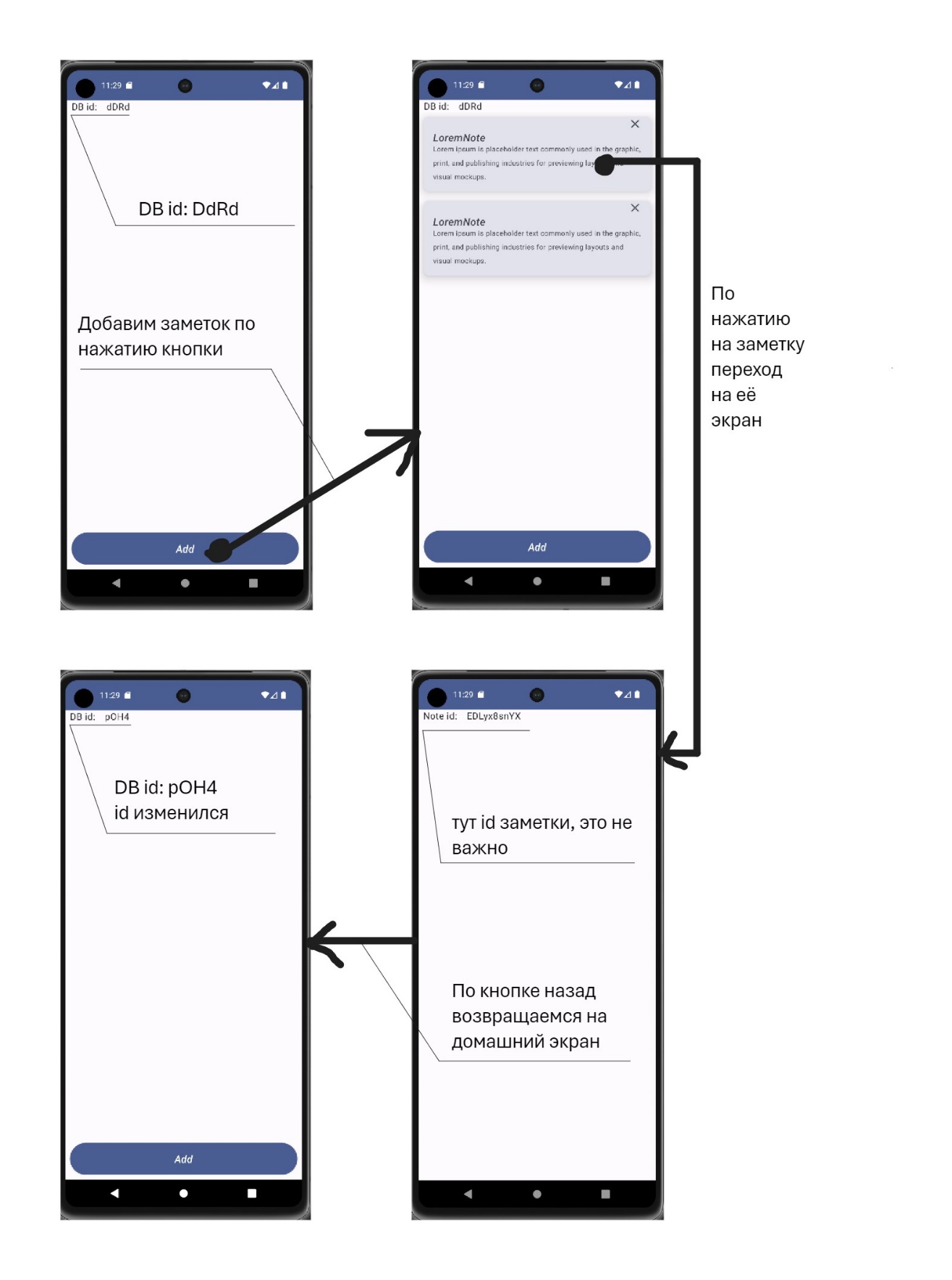
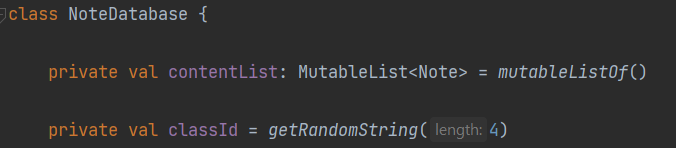
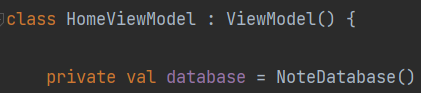
Non-singleton реализация:



Поведение приложения без использования Singleton-паттерна для класса базы данных NoteDatabase.

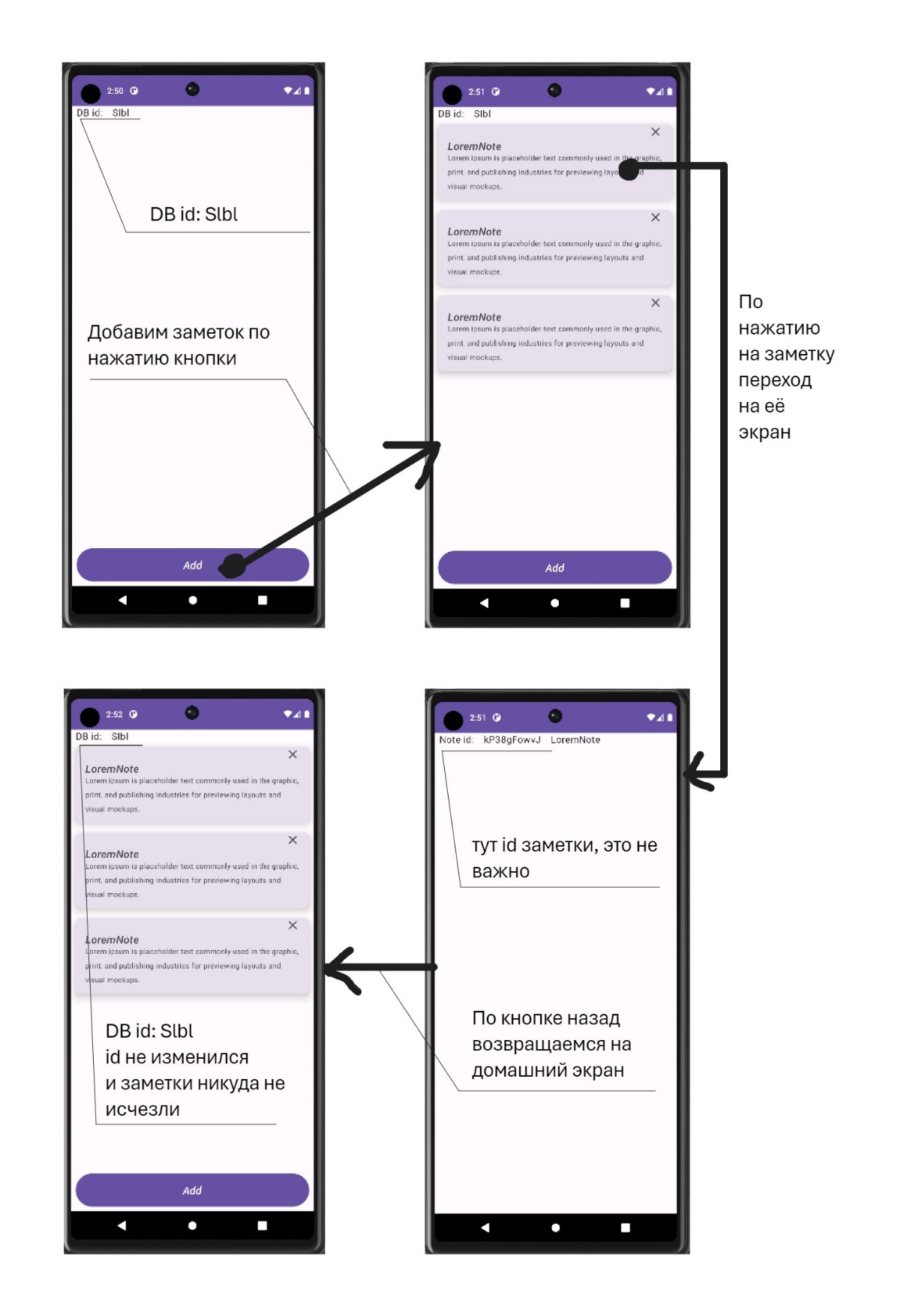


При каждом возвращении на домашний экран, контроллер экрана пересобирается и заново обращается к конструктору NoteDatabase.

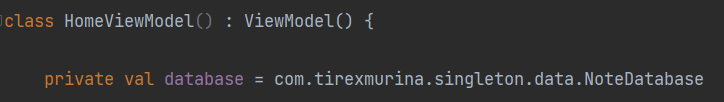


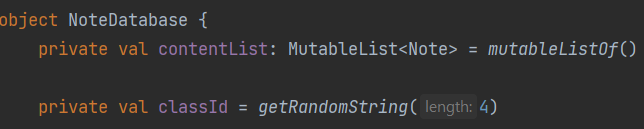
Далее получает новый объект этого класса и старые данные, записанные в старый объект NoteDatabase благополучно потерялись. Проблема явная: база данных буквально не является таковой, теряет данные при каждом пересоздании любого класса, обращающегося к её конструктору. Любой класс, обращаясь к ней из любой области пространства имен, имеет право создать свой собственный «нулевый» объект NoteDatabase, игнорируя изменения, вносимые другими классами. Это две типичные проблемы, которые решают Singleton’ы: контроль над конструктором и единая точка доступа к классу.

Singleton реализация:

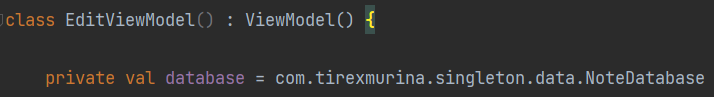


Теперь, как видим, все работает как надо, и при пересборке, HomeViewModel получает все тот же объект с тем-же id

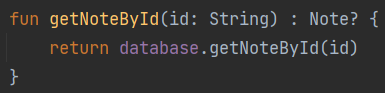




Более того, на втором экране теперь так же идет обращение к NoteDatabase



И данные которые на нем отображаются так же берутся напрямую из класса базы данных



То есть теперь, не зависимо от точки обращения, всегда будет получен один и тот же экземпляр NoteDatabase