CoreData

Part I

План лекции

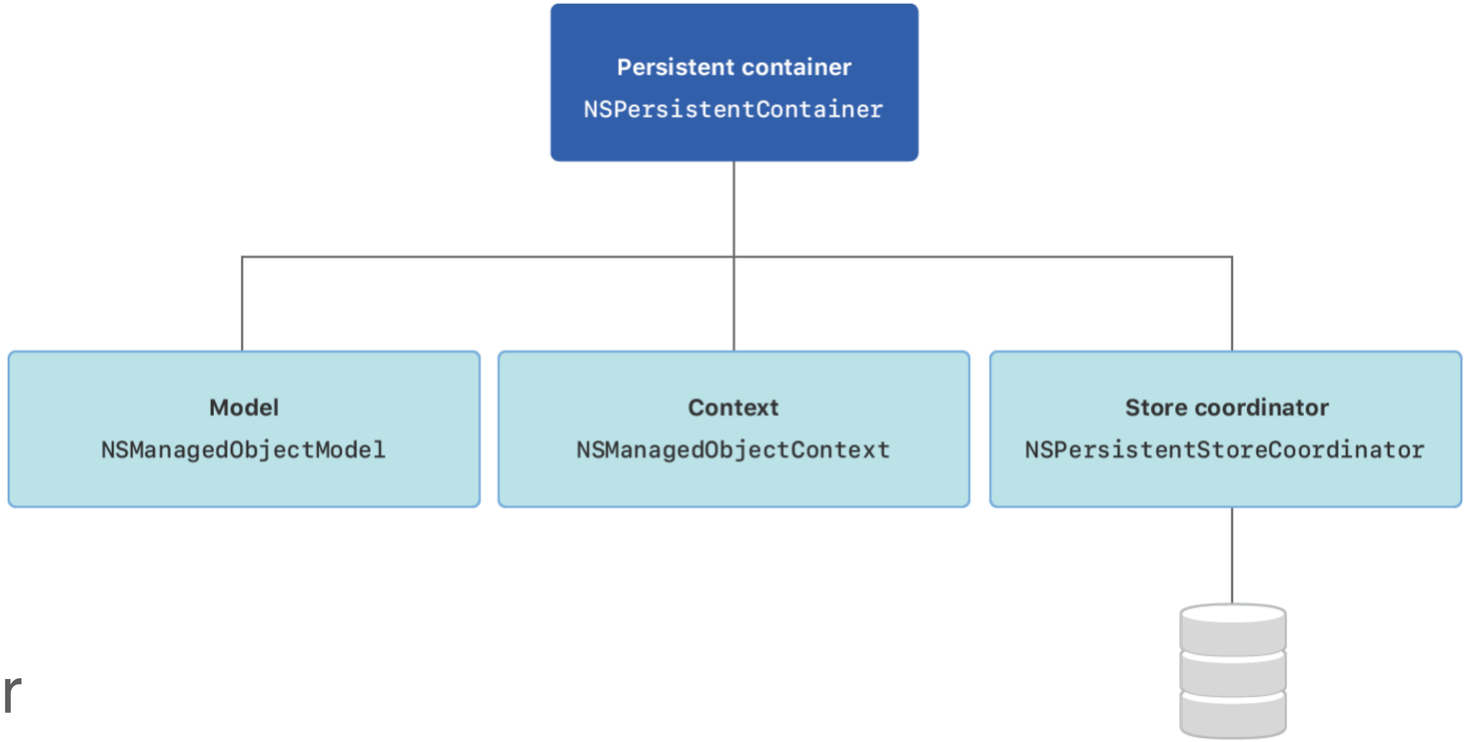
- Что такое и зачем нужно?
- Core Data Stack
- Data Model
- Managed Objects And Fetch Request
- NSManageObjects

Что такое CoreData?

- framework
- позволяет работать со слоем модели (MVC)
- снижение количества кода на 50-70%
- создан для управления графом объектов

CoreData Stack

- набор классов и объектов
- NSPersistentContainer
- NSManagedObjectModel
- NSManagedObjectContext
- NSPersistentStoreCoordinator



NSPersistentStoreCoordinator

Исходя из названия, NSPersistentStoreCordinator осуществляет сохранение данных и обеспечивает соответствие сохраненных данных и модели данных. Служит связующим звеном между хранилищем данных и MOC (managed object context). Также отвечает за загрузку и кэширование данных.

Persistent Store Coordinator дирежирует оркестром CoreData. Несмотря на важную роль в стэке, мы редко взаимодействуем с ним напрямую

NSManagedObjectContext

NSManagedObjectContext управляет набором объектов модели, экземплярами класса NSManagedObject. Разумно иметь в проекте несколько NSManagedObjectContext’ов - например, при работе с многопоточностью. В то время, как OM или PSC могут использоваться в разных потоках, MOC никогда не должен вызываться из потока, отличного от потока его создания. Данная тема будет освещена на второй лекции

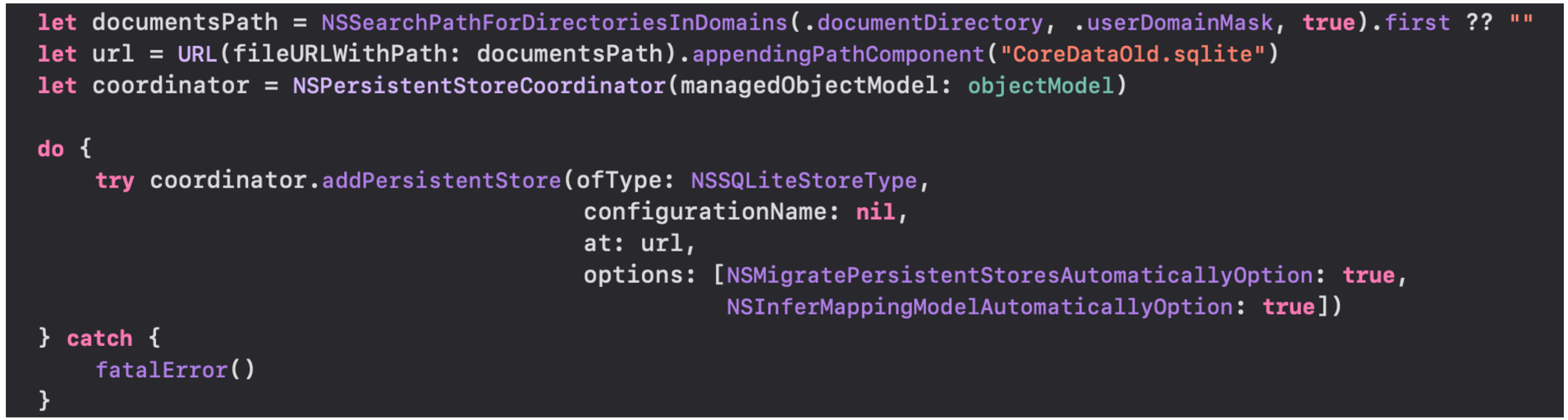
NSManagedObjectModel

Managed Object Model - модель данных приложения.
Хотя Core Data и не является БД, корректным будет сравнить MOM и схему БД. Она содержит информацию о сущностях, их атрибутах и связях друг с другом.

Как было до iOS 10

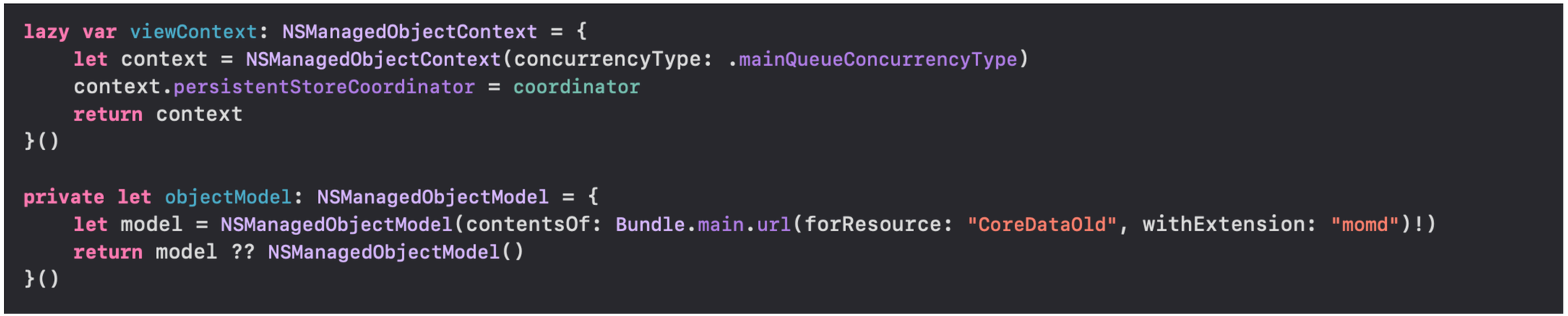
- Инициализируем NSPersistentStoreCoordinator
- Инициируем NSManagedObjectModel
- Инициируем NSManagedObjectContext
- Связываем context и coordinator
- Если контекстов несколько, то нужно синхронизировать изменения между друг другом

Как было до iOS 10



```
let documentsPath = NSSearchPathForDirectoriesInDomains(.documentDirectory, .userDomainMask, true).first ?? ""
let url = URL(fileURLWithPath: documentsPath).appendingPathComponent("CoreDataOld.sqlite")
let coordinator = NSPersistentStoreCoordinator(managedObjectModel: objectModel)

do {
    try coordinator.addPersistentStore(ofType: NSSQLiteStoreType,
                                       configurationName: nil,
                                       at: url,
                                       options: [NSMigratePersistentStoresAutomaticallyOption: true,
                                                 NSInferMappingModelAutomaticallyOption: true])
} catch {
    fatalError()
}
```



```
lazy var viewContext: NSManagedObjectContext = {
    let context = NSManagedObjectContext(concurrencyType: .mainQueueConcurrencyType)
    context.persistentStoreCoordinator = coordinator
    return context
}()

private let objectModel: NSManagedObjectModel = {
    let model = NSManagedObjectModel(contentsOf: Bundle.main.url(forResource: "CoreDataOld", withExtension: "momd")!)
    return model ?? NSManagedObjectModel()
}()
```

Как стало после?

- Достаточно проинициализировать NSPersistentStoreContainer
- Если нужно изменить, то нужно использовать NSPersistentStoreDescriptor

NSPersistentContainer (iOS 10 +)

Persistent Container использует имя, которое мы передаем при инициализации сразу и для Managed Object Model (name.momd) и как имя для Persistent Store (name.sqlite), который он достает или создает в дефолтной директории (Application Support).

Как стало после?

```
let conainer: NSPersistentContainer = {
    let container = NSPersistentContainer(name: "CoreDataOld")
    container.loadPersistentStores { desc, error in
        if let error = error {
            fatalError(error.localizedDescription)
        }
    }
    return container
}()
```

NSPersistentStoreDescriptor

- Объект для загрузки PersistentStore
- Нужен для более тонкой настройки CoreData

Data Model

- xml
- Удобный интерфейс
- Концепция как у Story Boards
- генерация Managed Object

NSManagedObject

- Является объектом сущности из модели
- Может автоматически генерироваться
- Никогда не стоит передавать данные объекты между разными потоками
- Имеют свойство objectID

NSFechRequest

- Используется в качестве запроса данных
- NSsortDescriptor
- NSPredicate
- fetchBatchSize
- fetchLimit