



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình iOS

Bài 2. *Core Data Framework*

Ngành Mạng & Thiết bị di động

2015





Nội dung

1. Giới thiệu

- Tổng quan
- Các tính năng

2. Tích hợp Core Data Framework

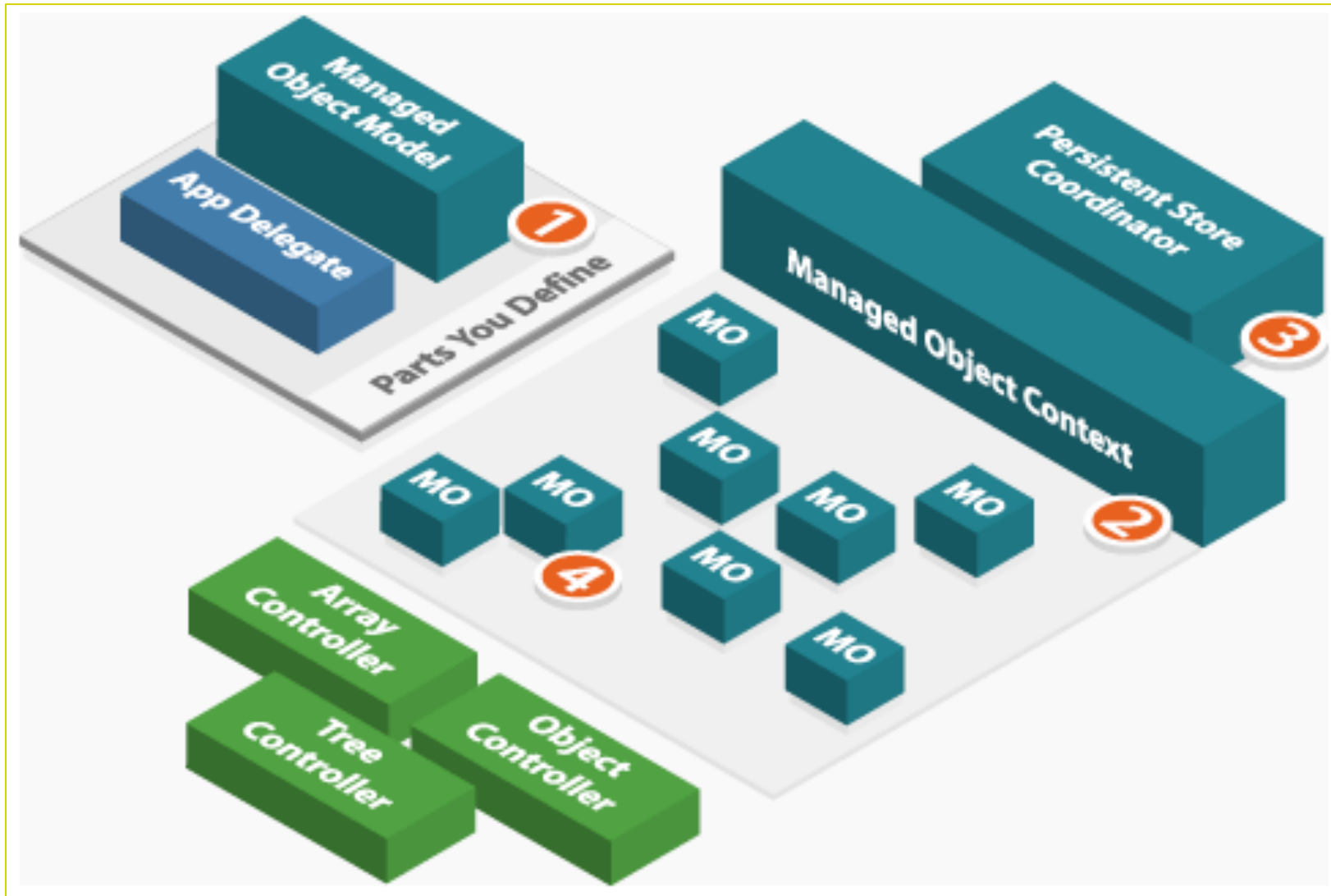
3. Model entity

4. Tạo các lớp đối tượng

5. Tương tác với dữ liệu dùng Core Data Manage Context



1. Giới thiệu Core Data Framework





1.1 Tổng quan

- ❑ Core Data Framework là một trong những lựa chọn tối ưu cho việc lưu trữ trong ứng dụng, đặc biệt là các ứng dụng cỡ trung và lớn.
- ❑ Cung cấp một công cụ trực quan để thiết kế cơ sở dữ liệu cùng các quan hệ (Data Relationship)
- ❑ Tiết kiệm đáng kể thời gian viết code.
- ❑ Giúp giảm đáng kể tài nguyên vùng nhớ (Memory) bởi khả năng thu hồi bộ nhớ.



1.2 Các tính năng

	Core Data	NSKeyedArchiver
Entity Modeling	Yes	No
Querying	Yes	No
Speed	Fast	Slow
Serialization Format	SQLite, XML, or NSData	NSData
Migrations	Automatic	Manual
Undo Manager	Automatic	Manual

Bảng so sánh Core Data và NSKeyedArchiver



1.2 Các tính năng

❑ Tính năng:

- Cho phép tương tác, xử lý lưu trữ đối tượng tối ưu.
- Cung cấp phương thức cho việc quản lý đối tượng cũng như việc quản lý Undo-Redo.
- Tối ưu hoá bộ nhớ ứng dụng.
- Hỗ trợ việc lưu trữ theo version và tích hợp dữ liệu khi cần thiết.



Nội dung

1. Giới thiệu

2. Tích hợp Core Data Framework

- Tích hợp trong lúc tạo project
- Tích hợp vào project có sẵn

3. Model entity

4. Tạo các lớp đối tượng

5. Tương tác với dữ liệu dùng Core Data Manage Context



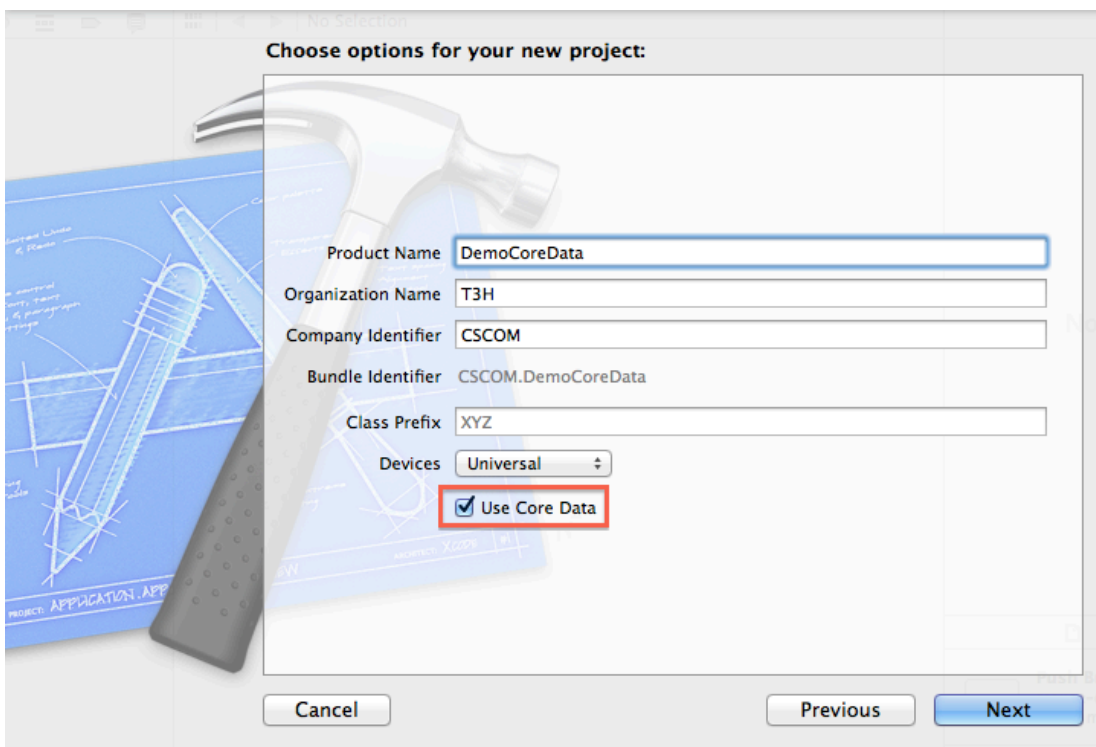
2. Tích hợp Core Data Framework

- ❑ Ta có thể tích hợp Core Data Framework vào dự án theo 2 cách sau:
 - Tích hợp trong lúc tạo project.
 - Tích hợp vào project có sẵn.



2.1 Tích hợp trong lúc tạo project

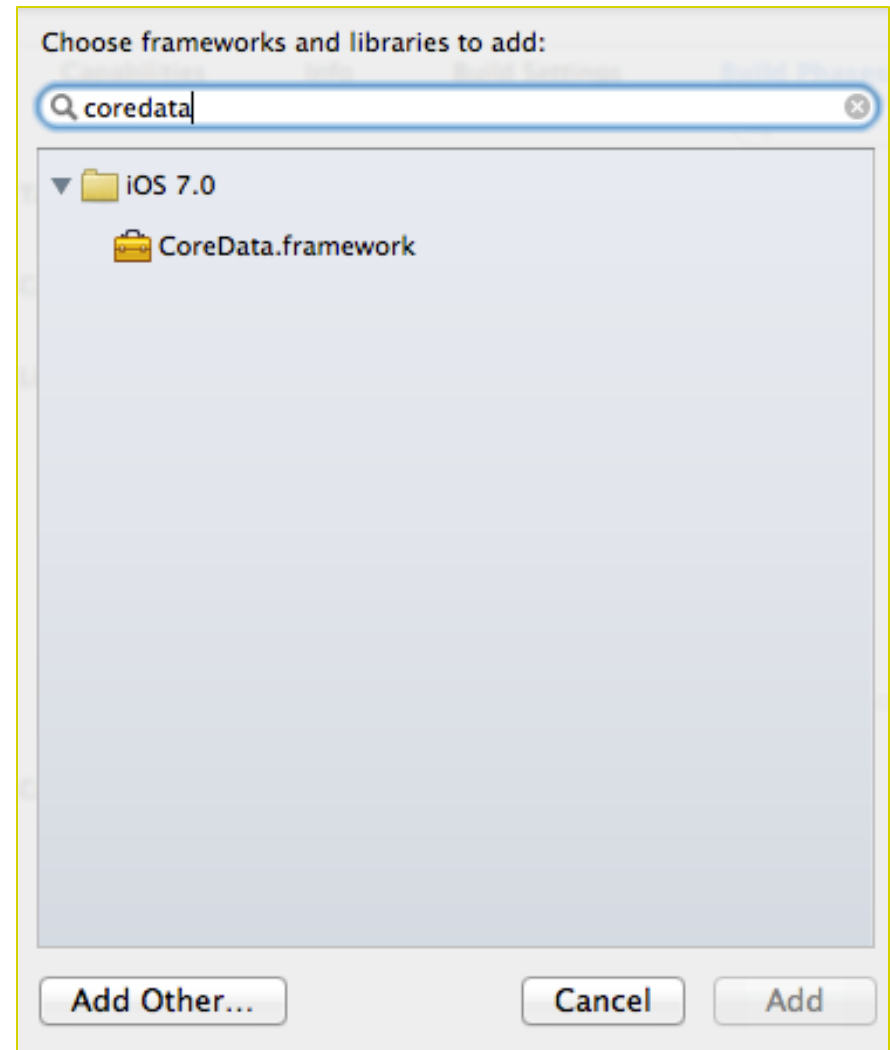
- ❑ Tích hợp trong lúc tạo project tương đối đơn giản. Khi tạo project ta chỉ việc check vào dòng “Use Core Data” hệ thống sẽ tự tạo những thông tin cần thiết.





2.1 Tích hợp vào project có sẵn

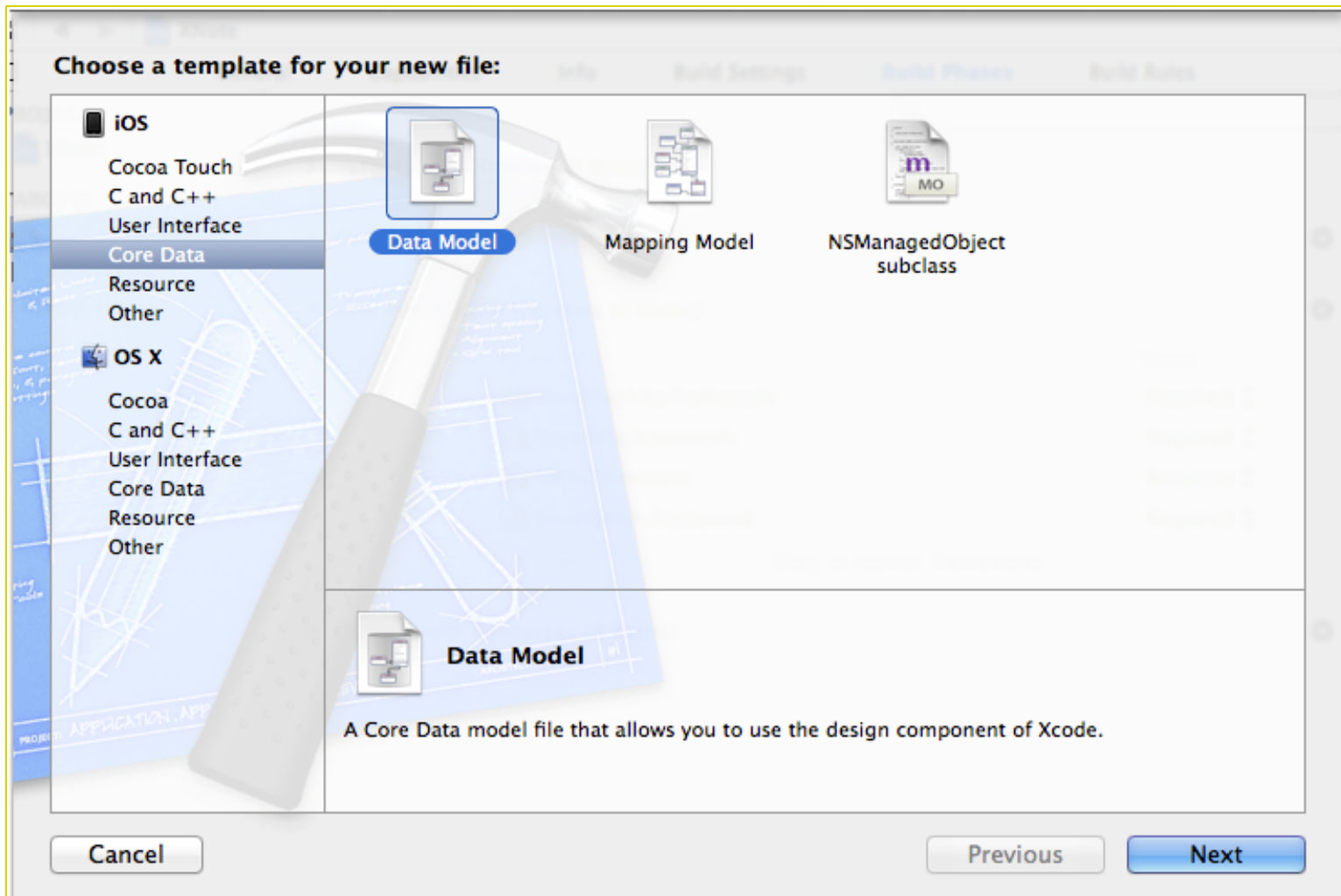
- ❑ Để tích hợp vào project có sẵn ta thực hiện những bước sau.
 - Bước 1: Thêm Framework Core Data vào project.





2.1 Tích hợp vào project có sẵn

- Bước 2: Tạo Database Model.





2.1 Tích hợp vào project có sẵn

- Bước 3: Thêm các đoạn mã cần thiết sau:
 - Trong tập tin AppDelegate.h:

```
#import <UIKit/UIKit.h>

@interface AppAppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;

@property (readonly, strong, nonatomic) NSManagedObjectContext *managedObjectContext;
@property (readonly, strong, nonatomic) NSManagedObjectModel *managedObjectModel;
@property (readonly, strong, nonatomic) NSPersistentStoreCoordinator *persistentStoreCoordinator;

- (void)saveContext;
- (NSURL *)applicationDocumentsDirectory;

@end
```



2.1 Tích hợp vào project có sẵn

- Triển khai các phương thức đã khai báo trong tập tin AppDelegate.m

```
- (NSManagedObjectContext *)managedObjectContext
{
    if (_managedObjectContext != nil) {
        return _managedObjectContext;
    }

    NSPersistentStoreCoordinator *coordinator = [self persistentStoreCoordinator];
    if (coordinator != nil) {
        _managedObjectContext = [[NSManagedObjectContext alloc] init];
        [_managedObjectContext setPersistentStoreCoordinator:coordinator];
    }
    return _managedObjectContext;
}

// Returns the managed object model for the application.
// If the model doesn't already exist, it is created from the application's model.
- (NSManagedObjectModel *)managedObjectModel
{
    if (_managedObjectModel != nil) {
        return _managedObjectModel;
    }
    NSURL *modelURL = [[NSBundle mainBundle] URLForResource:@"DemoCoreData" withExtension:@"momd"];
    _managedObjectModel = [[NSManagedObjectModel alloc] initWithContentsOfURL:modelURL];
    return _managedObjectModel;
}
```



2.1 Tích hợp vào project có sẵn

```
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator
{
    if (_persistentStoreCoordinator != nil) {
        return _persistentStoreCoordinator;
    }

    NSURL *storeURL = [[self applicationDocumentsDirectory]
        URLByAppendingPathComponent:@"DemoCoreData.sqlite"];

    NSError *error = nil;
    _persistentStoreCoordinator = [[NSPersistentStoreCoordinator alloc] initWithManagedObjectModel:[self
        managedObjectModel]];
    if (![ _persistentStoreCoordinator addPersistentStoreWithType:NSSQLiteStoreType configuration:nil URL:
        storeURL options:nil error:&error]) {
        NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
        abort();
    }

    return _persistentStoreCoordinator;
}
```



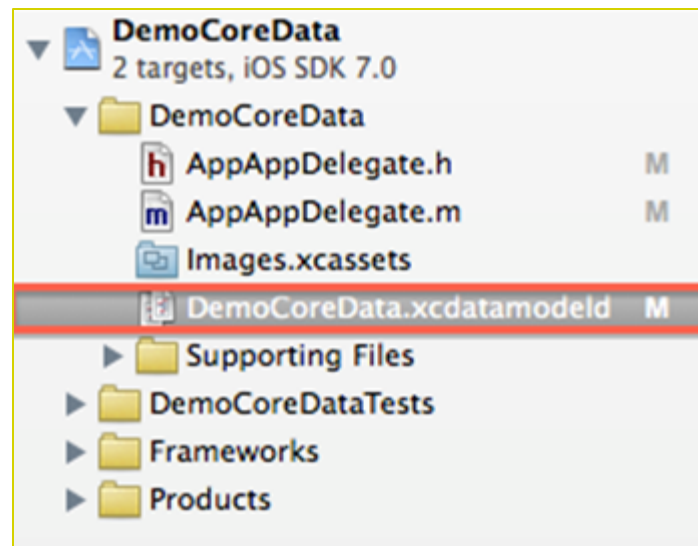
Nội dung

1. Giới thiệu
2. Tích hợp Core Data Framework
- 3. Model entity**
4. Tạo các lớp đối tượng
5. Tương tác với dữ liệu dùng Core Data Manage Context



3. Model Entity

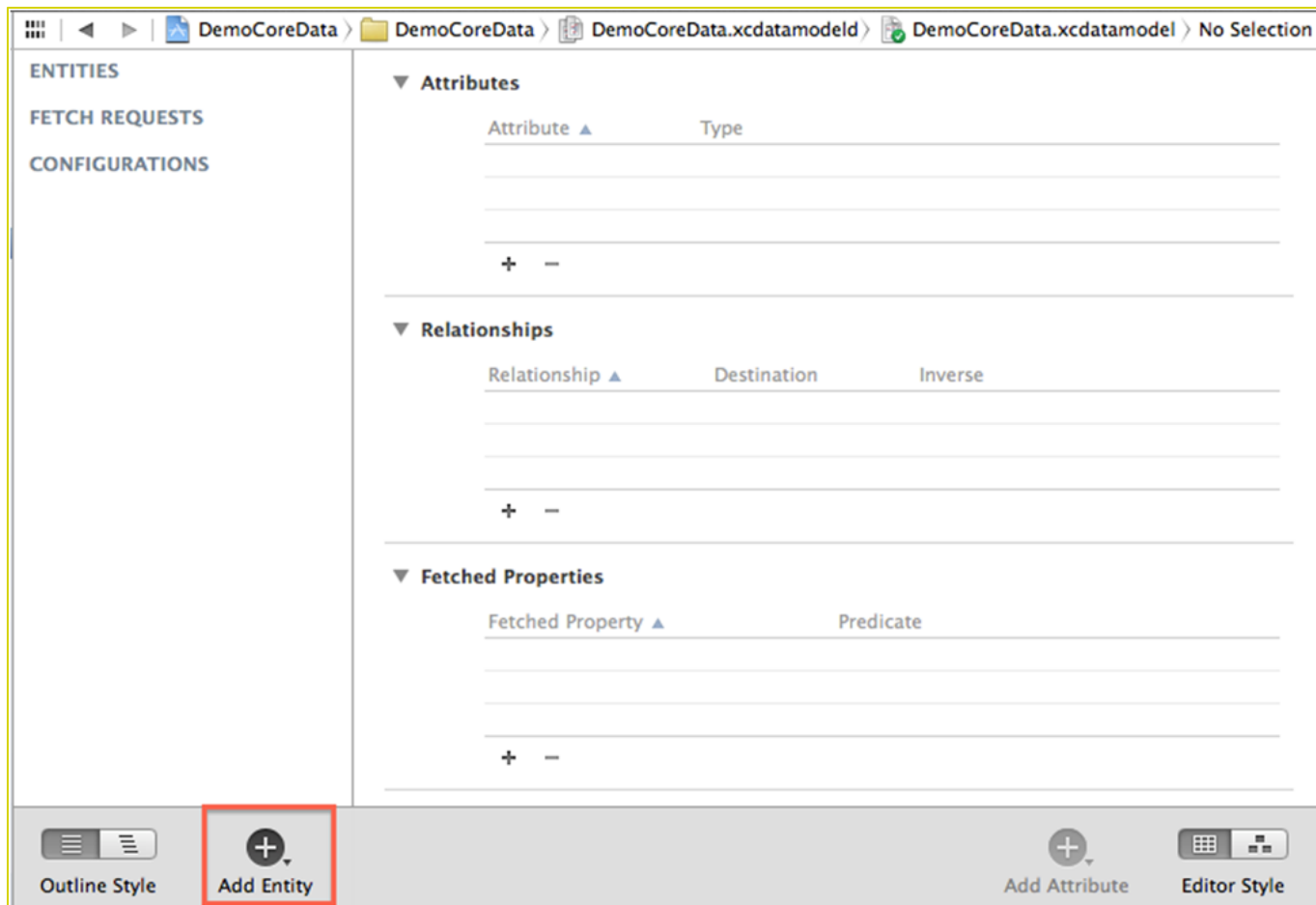
- ❑ Model Entity là đối tượng mô tả những thực thể mà ta cần lưu trữ. Có thể hiểu đối tượng này giống như các bảng dữ liệu trong database.
- ❑ Để tạo một entity thực hiện các bước sau:
 - Bước 1: chọn tập tin *.xcdatamodeld





3. Model Entity

- Bước 2: Thêm các Entity, ta thực hiện tương ứng với việc tạo bảng trong database:

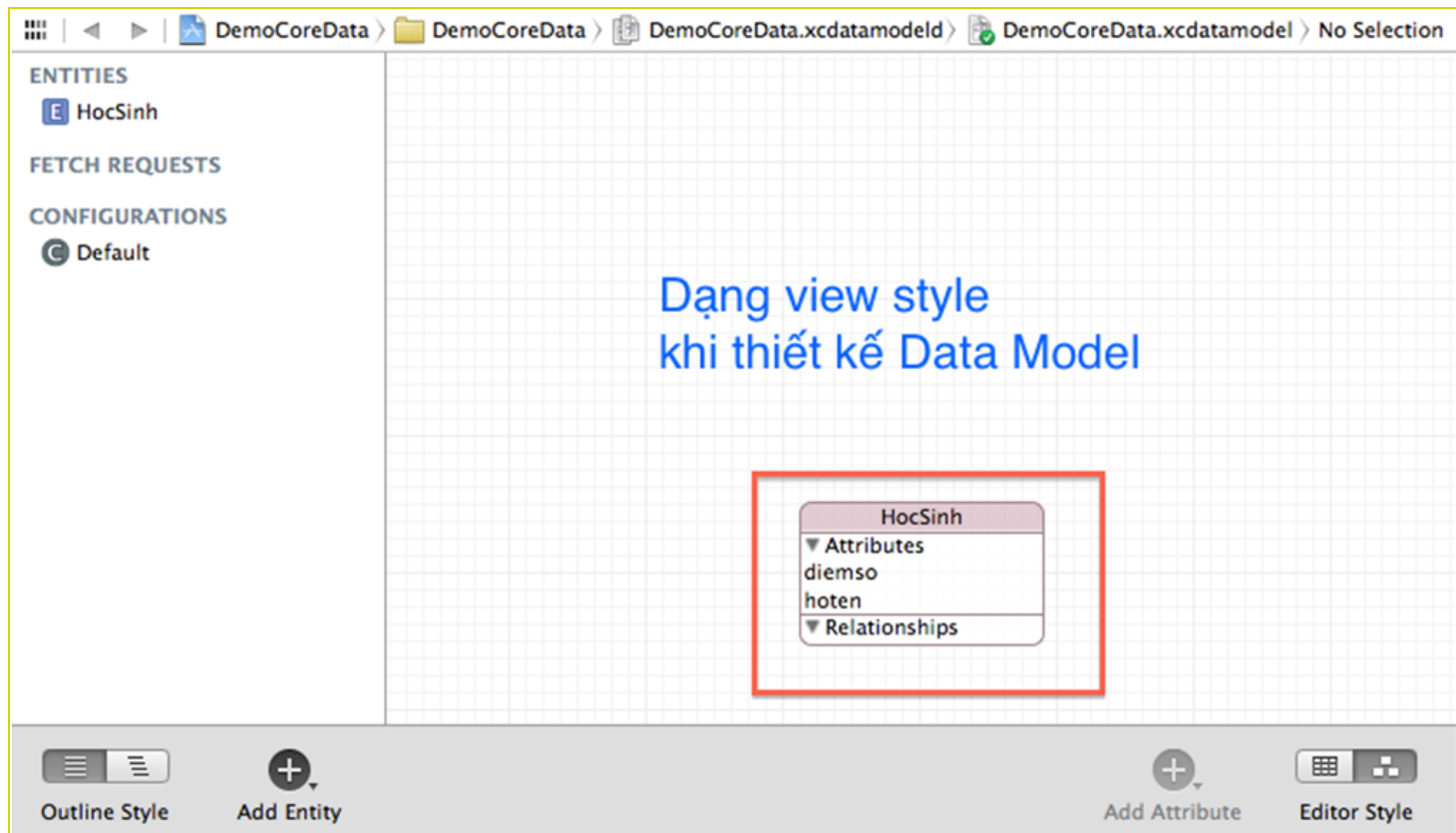




3. Model Entity

❑ Có hai dạng để thiết kế database:

- Dạng Editor
- Dạng Style: tương tự dạng thiết kế diagram trong SQL server





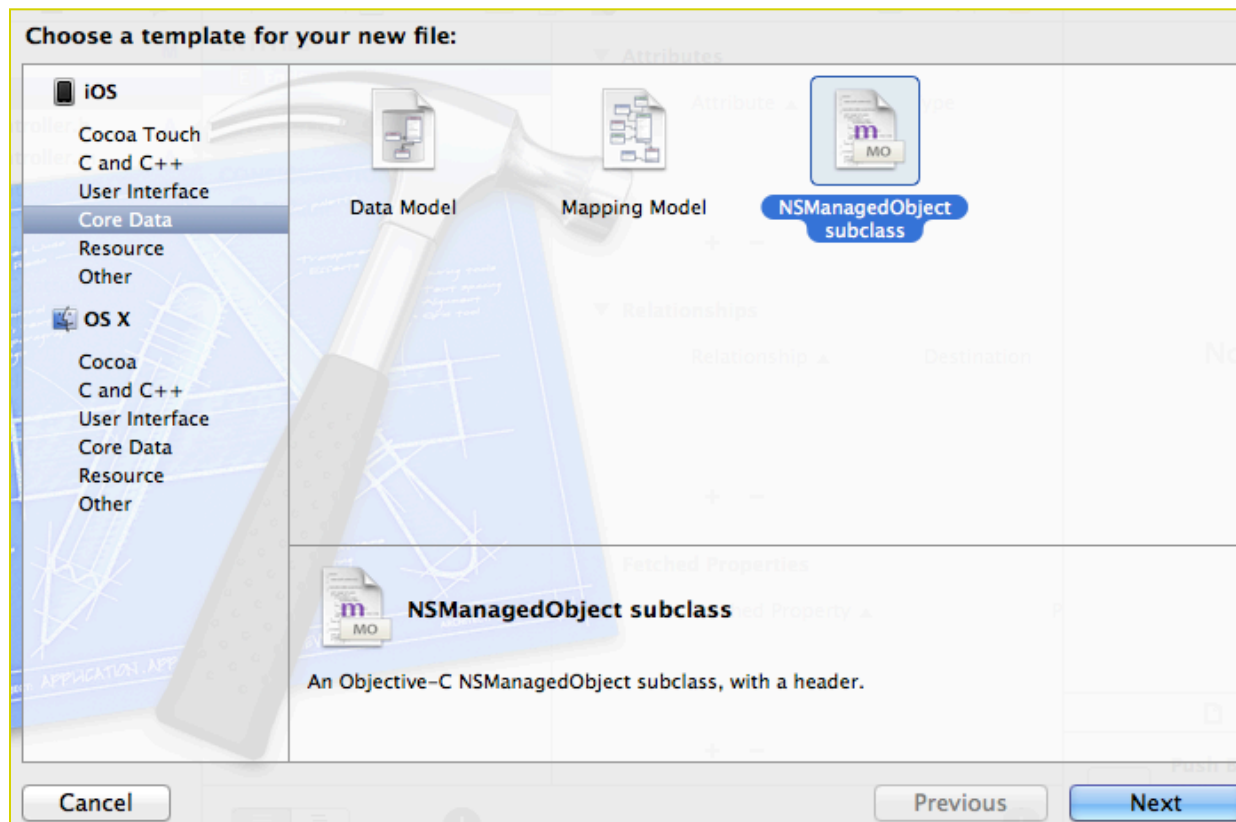
Nội dung

1. Giới thiệu
2. Tích hợp Core Data Framework
3. Model entity
4. Tạo các lớp đối tượng
5. Tương tác với dữ liệu dùng Core Data Manage Context



4. Tạo các lớp đối tượng

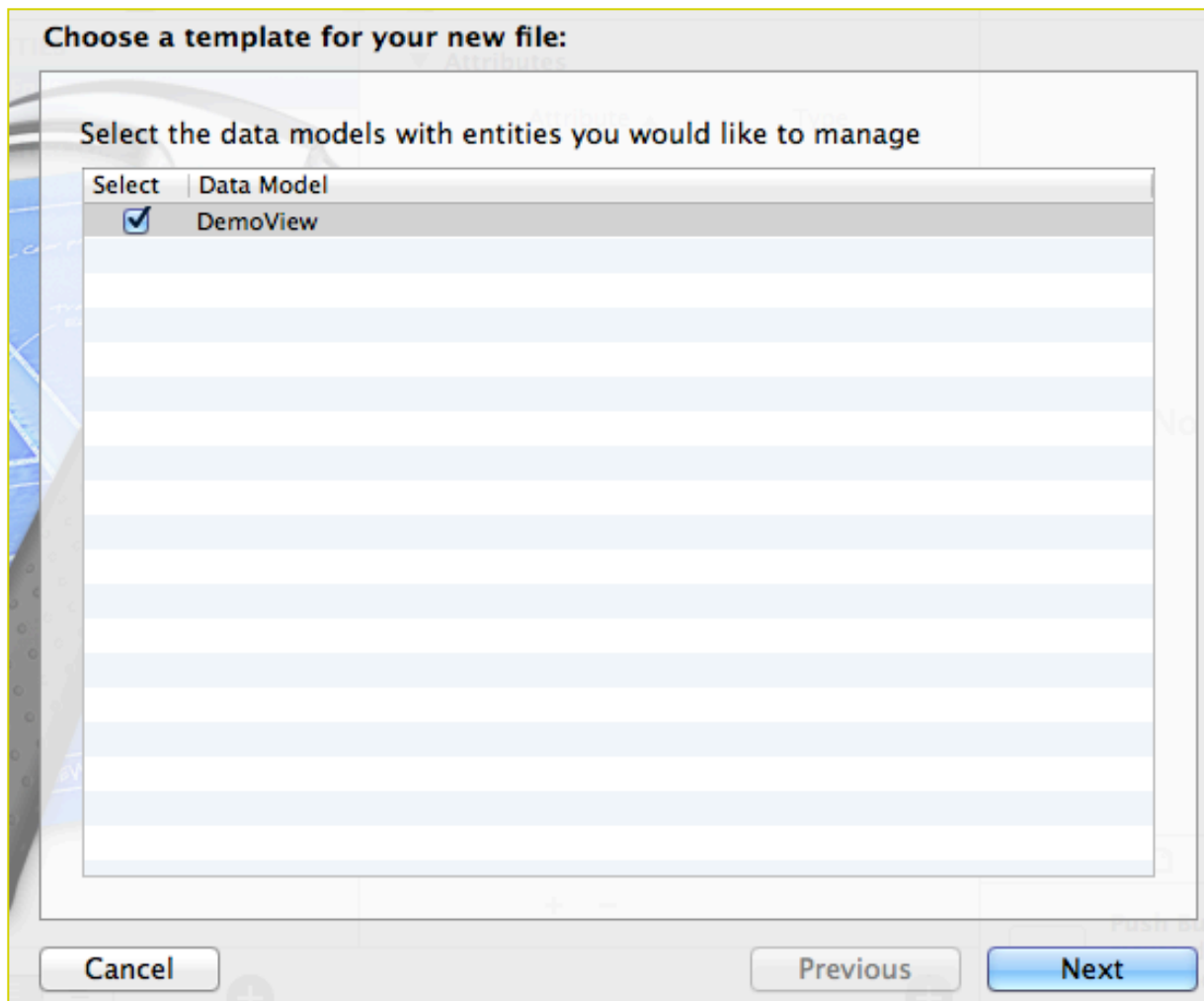
- ❑ Manage Object Class là lớp đối tượng được tạo ra cho các Entity. Việc phát sinh mã là hoàn toàn tự động. Ta thực hiện như sau.
 - New -> New File... -> Core Data -> NSManagedObject subclass.





4. Tạo các lớp đối tượng

- Chọn Data Model có chứa đối tượng cần phát sinh tập tin quản lý.





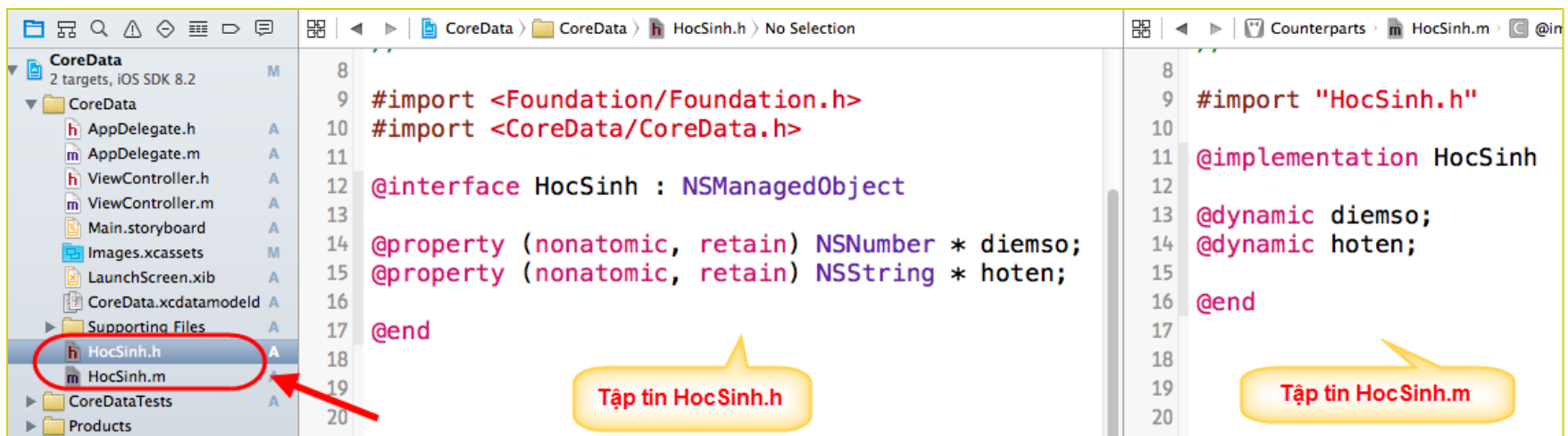
- [illegible]





4. Tạo các lớp đối tượng

- Sau khi hoàn thành ta sẽ có lớp với các trường dữ liệu của đối tượng entity.





Nội dung

1. Giới thiệu
2. Tích hợp Core Data Framework
3. Model entity
4. Tạo các lớp đối tượng
5. Tương tác với dữ liệu dùng Core Data Manage Context

5. Tương tác với dữ liệu dùng Core Data Manage Context



- ❑ Core Data Framework cung cấp cho ta đối tượng Manage Object Context tạo môi trường để thực hiện các hành động thêm, xoá, sửa cơ sở dữ liệu.
- ❑ Sử dụng lớp `NSEntityDescription` với phương thức `insertNewObjectForEntityForName:inManagedObjectContext:` để thêm dữ liệu.
- ❑ Sau khi thêm ta cần lưu lại `managedObjectContext` thông qua phương thức “save”

5. Tương tác với dữ liệu dùng Core Data Manage Context



❑ Ví dụ:

```
HocSinh* newHocSinh = [NSEntityDescription insertNewObjectForEntityForName:@"HocSinh"
                                                                    inManagedObjectContext:self.managedObjectContext];
newHocSinh.hoten = @"Tui la T3H";
newHocSinh.diemso = @10;

NSError *error = nil;
[self.managedObjectContext save:&error];
if (!error) {
    NSLog(@"Database has been saved");
}
```

5 Tương tác với dữ liệu dùng Core Data Manage Context



- ❑ Các bước cần thiết để thực hiện việc truy vấn với Manage Context:

```
// Đọc dữ liệu
// 1. Tạo ra request
NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init];
// 2. Tạo Entity cần request
NSEntityDescription *entity =
    [NSEntityDescription entityForName:@"HocSinh" inManagedObjectContext:self.managedObjectContext];
// 3. Gán Entity cho request
[fetchRequest setEntity:entity];

NSError *fetchError = nil;

// 4. Gọi hàm đọc data với request
NSArray *result = [self.managedObjectContext executeFetchRequest:fetchRequest error:&fetchError];

// 5. Kiểm tra lỗi (optional)
if (!fetchError) {
    for (HocSinh *hocsinh in result) {
        NSLog(@"Họ ten : %@ đạt được %d điểm", hocsinh.hoten, [hocsinh.diemso intValue]);
    }
}
```

Thảo luận

