

iOS: Core Data

FORTGESCHRITTENE MOBILE ANWENDUNGEN
MARTIN HELLER & FARAH TAHIR

Agenda

- ▶ Einblick in Core Data
- ▶ Vorstellung der Core Data-App
- ▶ Vertiefte Erklärung anhand der App
- ▶ Leitfaden
- ▶ Übungsaufgabe



Einblick in Core Data

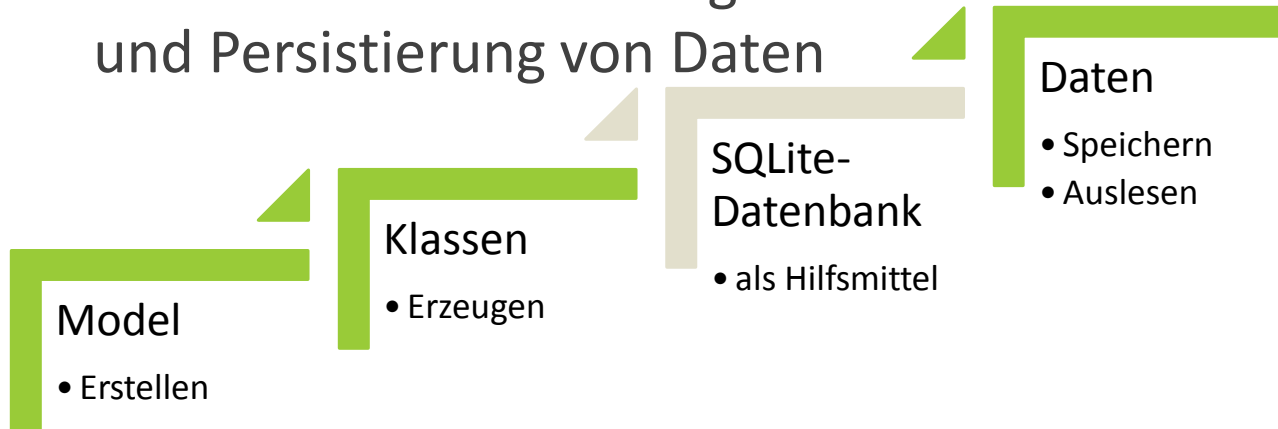
FEATURES UND ARCHITEKTUR

Was ist Core Data?

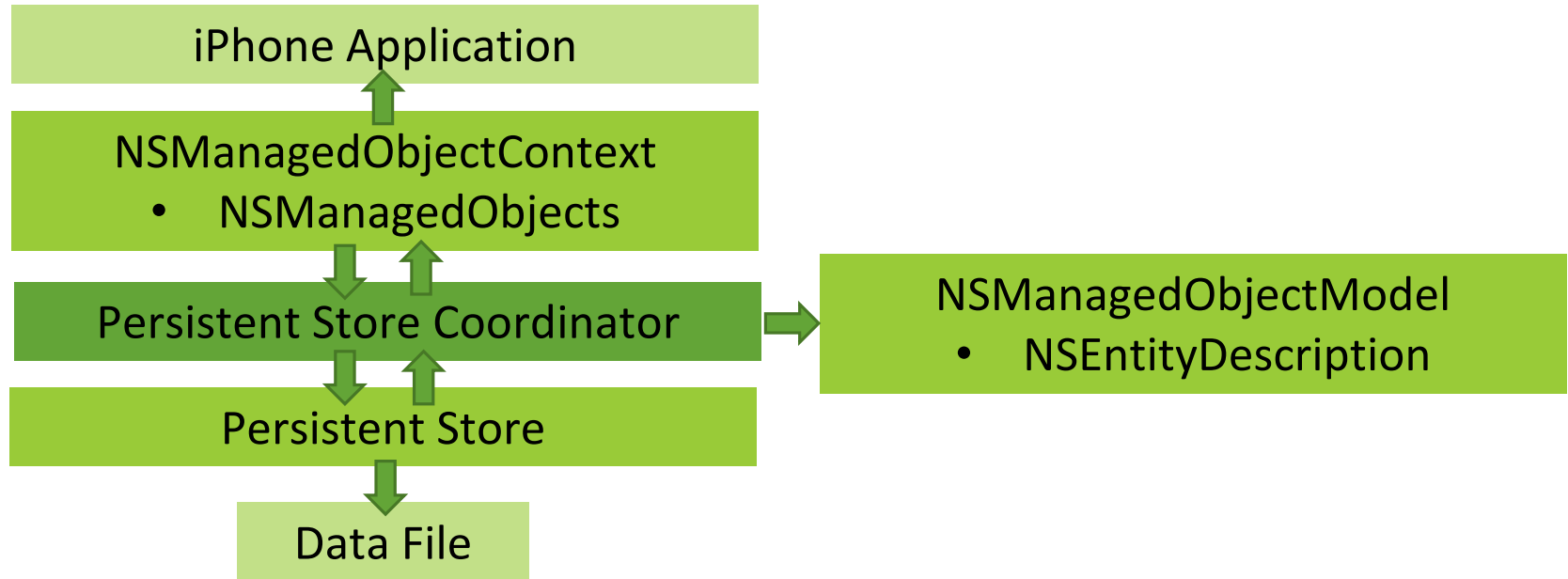
- ▶ Keine Datenbank
- ▶ Framework zur Verwaltung
und Persistierung von Daten

Was ist Core Data?

- ▶ Keine Datenbank
- ▶ Framework zur Verwaltung und Persistierung von Daten



Architektur



Architektur: Bestandteile

- ▶ **NSManagedObject**

- ▶ Repräsentation einer Entity aus Model

- ▶ **NSManagedObjectContext**

- ▶ Verantwortlich für das Speichern oder Verwerfen von Änderungen an NSManagedObject

NSManagedObjectContext

- NSManagedObjects

Architektur: Bestandteile

► NSManagedObject

Entity in Model

Buch

- titel[string]
- isbn[int]
- preis[decimal number]
- beschreibung[string]



Subklasse im Projekt

```
import Foundation
import CoreData

class Buch: NSManagedObject {
    @NSManaged var titel: String
    @NSManaged var isbn: Int
    @NSManaged var preis: NSDecimalNumber
    @NSManaged var beschreibung: String
}
```


Architektur: Bestandteile

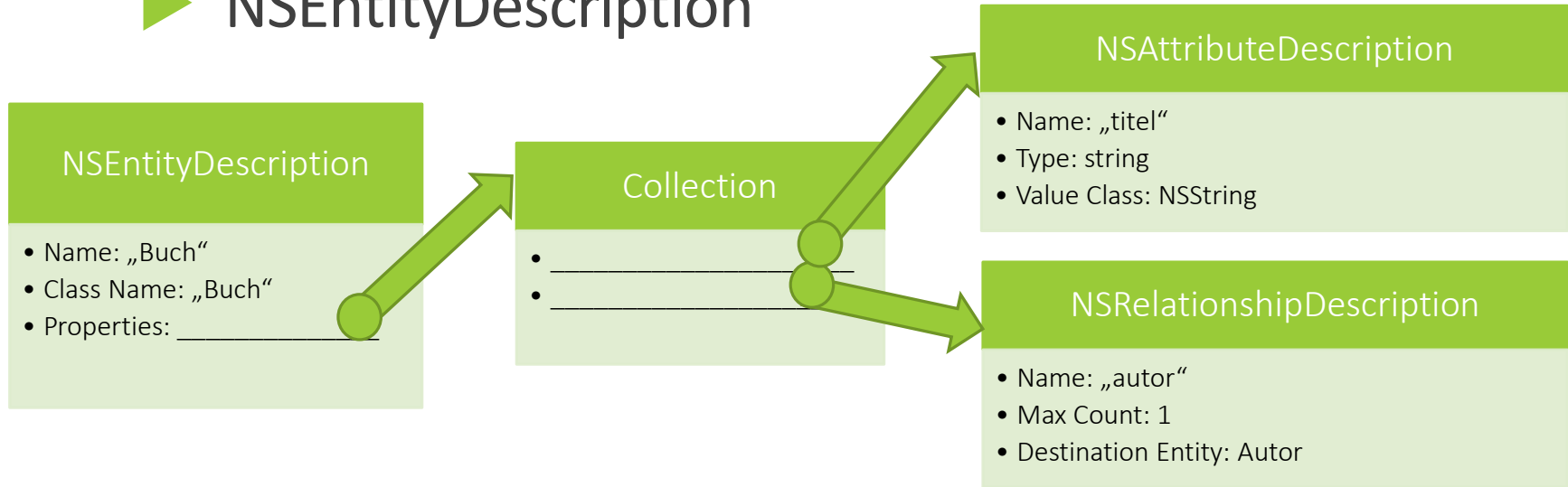
- ▶ **NSEntityDescription**
 - ▶ Metadaten von Entity
- ▶ **NSManagedObjectContext**
 - ▶ Schema, das NSManagedObjectContexts beschreibt
 - ▶ XCode: Data Model Design Tool

NSManagedObjectContext

- NSEntityDescriptions

Architektur: Bestandteile

► NSEntityDescription



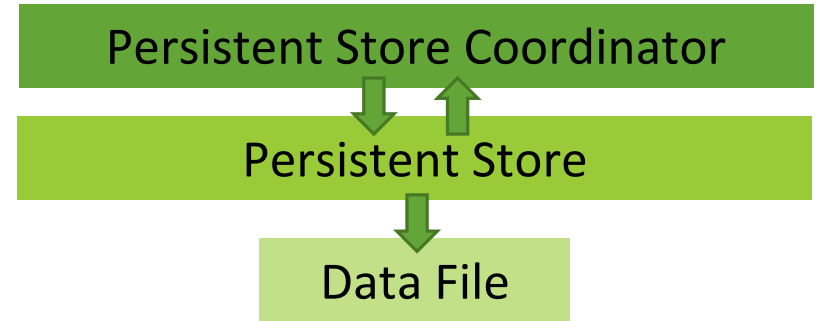
Architektur: Bestandteile

- ▶ Persistent Store

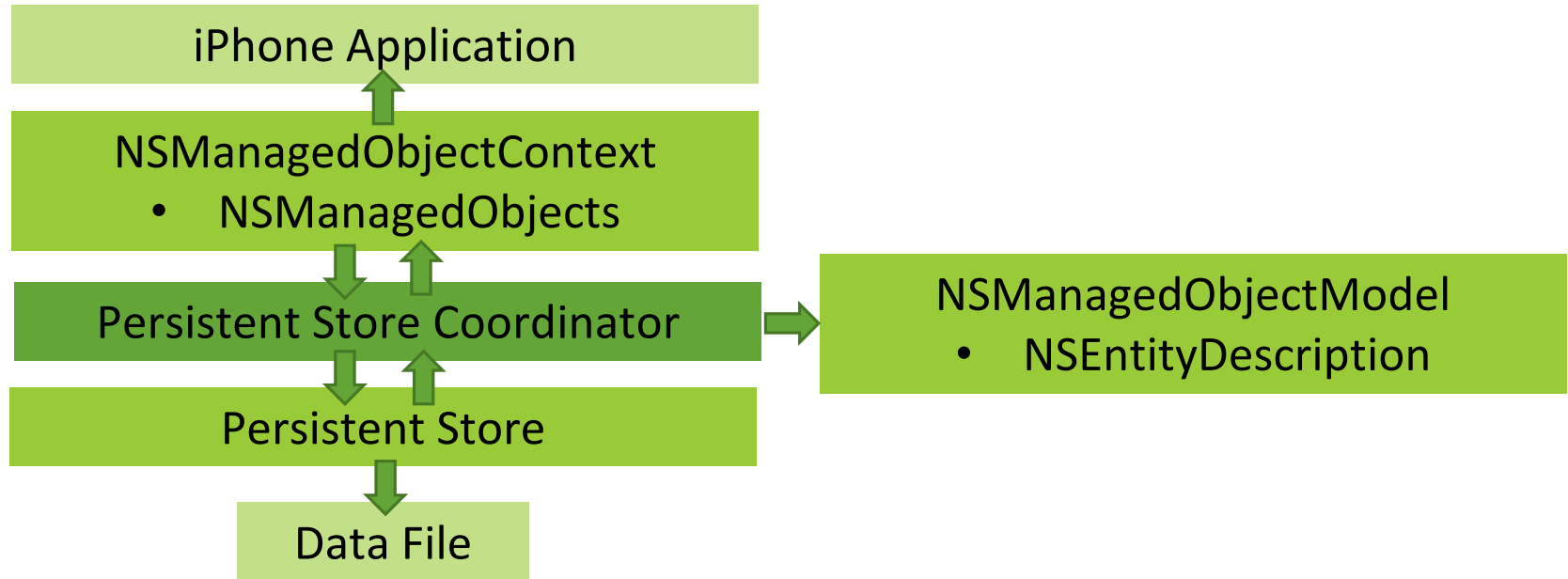
- ▶ Abstrakte Basisklasse zur Abbildung des Datenbanktyps

- ▶ Persistent Store Coordinator

- ▶ Verknüpfung herstellen zwischen Store und Model



Architektur

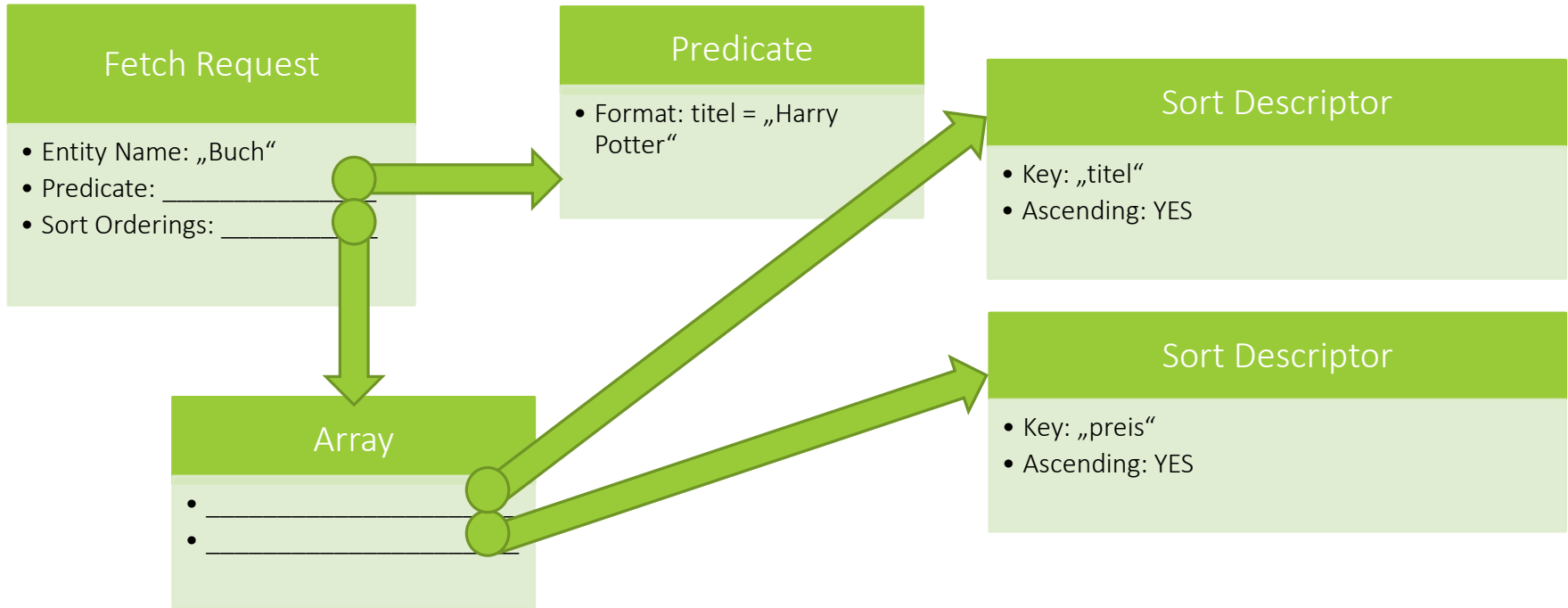


Objekte anlegen und löschen

```
let newItem = NSEntityDescription.insertNewObjectForEntityForName("LogItem",  
inManagedObjectContext: self.managedObjectContext!)
```

```
let logItemToDelete = logItems[indexPath.row]  
managedObjectContext?.deleteObject(logItemToDelete)
```

Objekte abfragen



Vorstellung der Core Data-App

HAUSHALTSBUCH

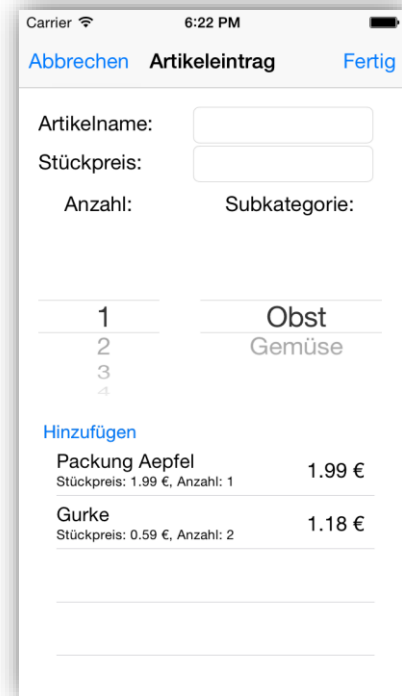
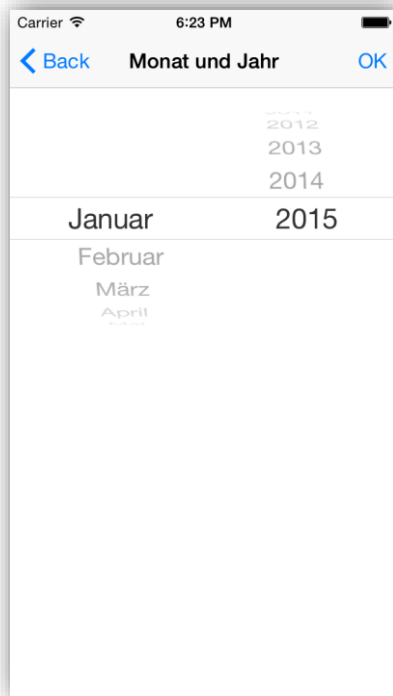
A solid green horizontal bar at the bottom of the slide.

Haushaltsbuch – Idee und Hauptfeatures

- ▶ App zur Verwaltung der eigenen Einnahmen und Ausgaben
- ▶ Buchungen können Artikel enthalten (Beispiel: Penny-Einkauf)
- ▶ Gefilterte Anzeige nach Monat und Jahr
- ▶ Gesamte Einnahmen und Ausgaben als Übersicht
- ▶ Buchungen und Artikel können verwaltet werden

[illegible]

Haushaltsbuch – Idee und Hauptfeatures



Vertiefte Erklärung anhand der App

HAUSHALTSBUCH

A solid green horizontal bar spanning the width of the slide at the bottom.

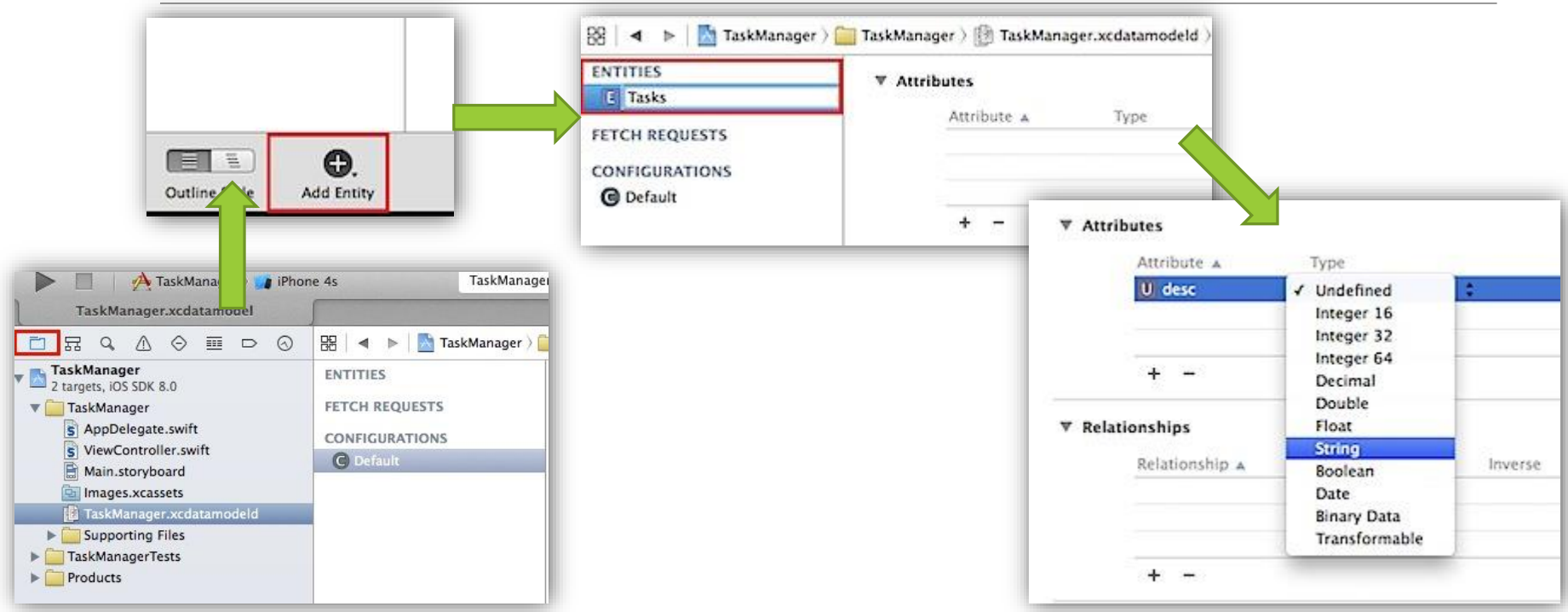
Leitfaden

DIE WICHTIGSTEN SCHRITTE

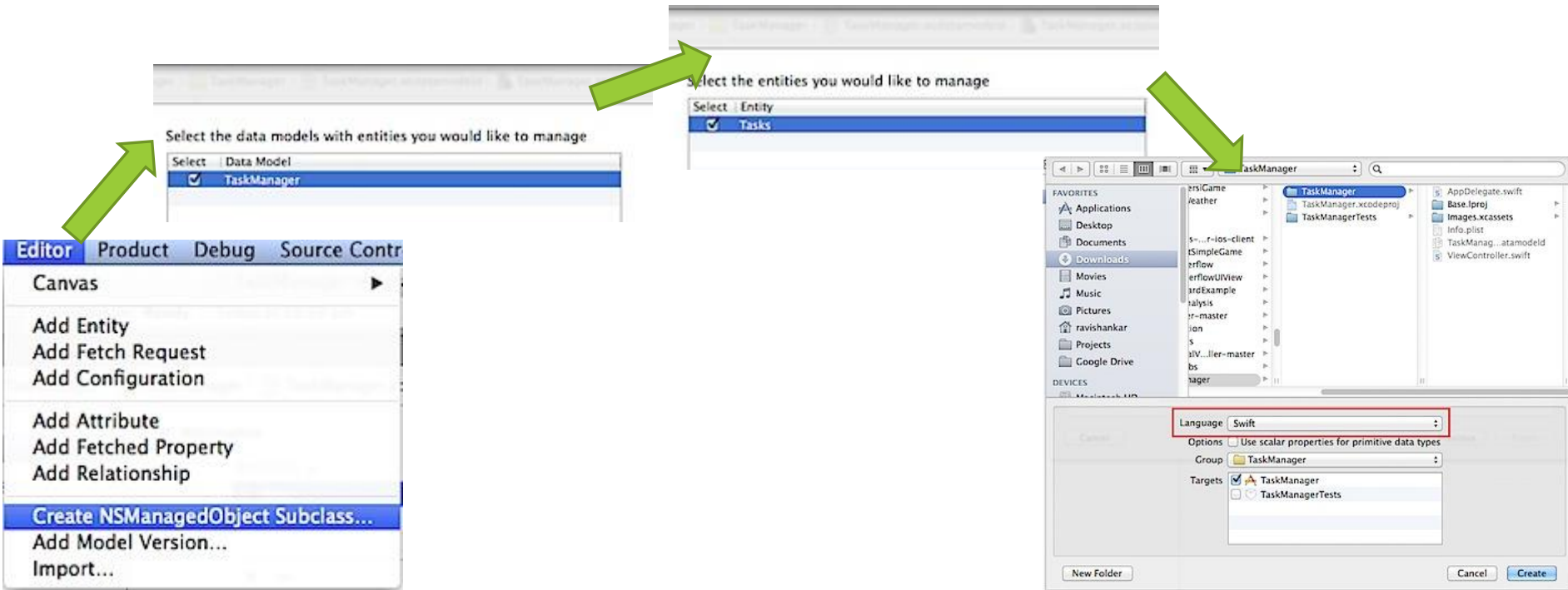
Leitfaden

- ▶ Beim Erstellen eines neuen Projekts ☒ Use Core Data
- ▶ Model (Entities, Attribute, Relationships) erstellen
- ▶ Subklassen von `NSManagedObject` erzeugen
- ▶ `ManagedObjectContext` holen
- ▶ Objekte erstellen/ abfragen/ löschen
- ▶ `FetchRequest` filtern/ sortieren
- ▶ Objektänderungen speichern/ verwerfen

Leitfaden: Model erstellen



Leitfaden: Subklasse erzeugen



Leitfaden: Objekte verwalten

ManagedObjectContext holen:

```
let managedObjectContext = (UIApplication.sharedApplication().delegate as AppDelegate).managedObjectContext
```

Zugriff auf Objekt:

```
let task = fetchedResultsController.objectAtIndex(indexPath) as Tasks
```

Objekt erstellen:

```
let task = NSEntityDescription.insertNewObjectForEntityForName("Tasks",  
inManagedObjectContext: managedObjectContext!) as? Tasks
```

Objekt löschen:

```
let managedObject:NSManagedObject = fetchedResultsController.objectAtIndex(indexPath)  
as NSManagedObject managedObjectContext?.deleteObject(managedObject)
```

(Transaktion muss noch gespeichert werden, damit es tatsächlich gelöscht wird)

Leitfaden: Objekte abfragen

FetchRequest vorbereiten:

```
let fetchRequest = NSFetchRequest(entityName: "Tasks")

let sortDescriptor = NSSortDescriptor(key: "desc", ascending: true)
fetchRequest.sortDescriptors = [sortDescriptor]

//Prädikat zum filtern erstellen/hinzufügen

let predicate1 = NSPredicate(format: "name contains %@", "b")

fetchRequest.predicate = predicate1
```

FetchRequest ausführen:

```
fetchedResultsController = NSFetchedResultsController(fetchRequest: fetchRequest,
managedObjectContext: managedObjectContext!, sectionNameKeyPath: nil, cacheName: nil)
fetchedResultsController.performFetch(nil)
```


Leitfaden: Objektänderungen speichern/ verwerfen

Änderungen speichern

```
managedObjectContext?.save(nil)
```

Änderungen verwerfen

```
managedObjectContext!.rollback()
```

Übungsaufgabe

TASK-MANAGER

goo.gl/HVNFFk

A solid green horizontal bar at the bottom of the slide.

Lösung

TASK-MANAGER

goo.gl/8NNYOJ

A solid green horizontal bar at the bottom of the slide.

Noch Fragen?

CORE DATA

A solid green horizontal bar at the bottom of the slide.