

## Praktikum iOS-Entwicklung im Wintersemester 2016/2017 Übungsblatt 6

- Abgabetermin:** 13.12.2016, spätestens 23:59 Uhr  
(Gilt nur für die Teilnehmer, die ihre Lösungen am Besprechungstermin vorstellen. Dies geschieht in Form einer ca. 5-minütigen Powerpoint-Präsentation. Die Abgabe muss daher die Präsentationsdatei enthalten und erfolgt über Uniworx.)
- Besprechung:** 14.12.2016

### Aufgabe 12: (H) KVO in Swift

In der Vorlesung wurde KVO für Objective-C Properties vorgestellt. In dieser Aufgabe sollen Sie kurz vorstellen, wie KVO in Swift eingesetzt werden kann.

Gehen Sie dabei auf das Schlüsselwort `dynamic` ein und was nötig ist, um damit KVO realisieren zu können. Vergleichen Sie die KVO von Swift mit der von Objective-C insbesondere bezüglich der Typisierung der beobachteten Variablen. Erwähnen Sie auch mögliche Alternativen zu dem oben genannten Schlüsselwort.

### Aufgabe 13: (H) Archiving in Swift

In der Vorlesung wurde `NSKeyedArchiver` vorgestellt, der Objekte mit allen Instanzvariablen serialisiert und persistent speichert. Um diesen Mechanismus auch in Swift zu verwenden, dürfen nur `NSCoding`-konforme Objective-C-Datentypen verwendet werden. In dieser Übung soll ein `SwiftKeyedArchiver` entwickelt werden, der keine Objective-C-Typen voraussetzt:

- Legen Sie zunächst ein Interface `SwiftCoding` an, dass die Methode `encode` und einen speziellen Initialisierer enthält.
- Legen Sie die Klasse `SwiftKeyedArchiver` an, die die folgenden Funktionalitäten unterstützt:
  - Die Klasse stellt eine Methode zur Serialisierung bereit, die ein `root`-Objekt und einen Dateinamen als Argumente erwartet. Das `root`-Objekt muss das `SwiftCoding`-Interface implementieren.
  - Am Ende des Serialisierungsvorgangs sollen die Daten unter dem übergebenen Dateinamen gespeichert werden können.
- Die Serialisierung soll nun so ablaufen, dass alle Objekte, die das `SwiftCoding`-Interface implementieren, eine Datenstruktur übergeben bekommen, in der sie ihre Instanzvariablen ablegen. Die Serialisierung soll als atomare Datentypen zunächst nur Integer und Strings unterstützen.

Demonstrieren Sie die Funktionalität ihrer Serialisierung, indem Sie eine Klassenstruktur mit zwei Vererbungshierarchien speichern und wiederherstellen. Die einzelnen Klasseninstanzen sollen jeweils mindestens zwei Instanzvariablen vom Typ Integer oder String enthalten.

## Aufgabe 14: (H) SQLite in iOS

Die in der Vorlesung vorgestellten Methoden zur persistenten Speicherung von Anwendungsdaten sind vor allem für kleinere Datenmengen sinnvoll. In dieser Aufgabe sollen Sie sich deshalb mit der Verwendung von SQLite-Datenbanken vertraut machen. SQLite ist ein relationales Datenbanksystem, das eine SQL-Datenbank in einer einzelnen Datei verwaltet. Wegen seiner etwas eingeschränkten Funktionalität gegenüber MySQL oder PostgreSQL eignet sich dieses Datenbanksystem sehr für den Einsatz in mobilen Systemen.

Unter <http://chinookdatabase.codeplex.com/downloads/get/557773> steht eine Beispieldatenbank von Einkäufen aus dem iTunes-Store bereit. Ziel dieser Aufgabe ist es, die Daten, die in der Datenbank enthalten sind, strukturiert auf der Konsole auszugeben. Die Ausgabe soll die folgende Form haben:

```
Name der Spalte 1 | Name der Spalte 2 | Name der Spalte 3 | ...
===== | ===== | ===== | ...
Zeile 1, Wert 1 | Zeile 1, Wert 2 | Zeile 1, Wert 3 | ...
...
```

Aggregieren Sie die Daten der Datenbank so, dass Sie zu den Einkäufen von

- Michelle Brooks aus New York,
- Fynn Zimmermann aus Frankfurt und
- Lucas Mancini aus Rom

die folgenden Metadaten der einzelnen Produkte ausgeben können:

- Artist
- Track
- Album
- Artist
- Genre
- Name des Medientyps

Um die enthaltenen Daten anzuschauen, können Sie auf das Tool `sqlitebrowser` zurückgreifen, das unter <http://sqlitebrowser.org/> zum Download angeboten wird.

- a. Verwenden Sie für den Zugriff auf die Datenbank Objective-C.
- b. Verwenden Sie für den Zugriff auf die Datenbank Swift. Falls Sie einen Wrapper verwenden, der den Datenbankzugriff verwaltet, stellen Sie kurz vor, warum Sie sich für diese Lösung entschieden haben und wie der Aufruf der `sqlite3`-Methoden sonst funktionieren würde.