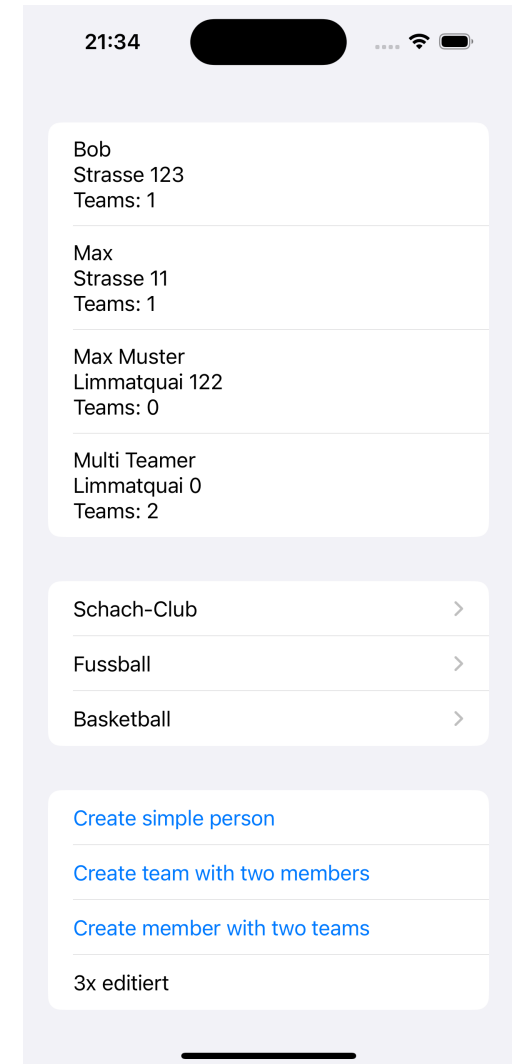


Übung 6

- Der Schwerpunkt dieser Übung ist das Persistieren von Daten mit Files und Core Data, sowie das Erstellen eines einfachen Charts mit dem Swift Charts Framework.
- Zusätzlich verwenden Sie Unit Tests, um Funktionen unabhängig von der App zu testen.



Xcode-Projekt

- Erstellen Sie ein neues Projekt **Persistence**: Wählen Sie **iOS App** als Template und **SwiftUI** als Interface. Wählen Sie ausserdem **Core Data** und **Unit Tests** als Optionen aus.
- Diese vom Projekt-Wizard generierte App bietet schon einiges an Funktionalität, vor allem in der `ContentView`.
- *Tipp: Stellen Sie sicher, dass sie die neuste Xcode Version verwenden. Wichtig: Apple hat das Core Data Default-Setup oft angepasst. Diskussionen auf StackOverflow und Co. enthalten deshalb häufig veraltete Erklärungen.*

Core Data: App umbauen für Personen-Daten

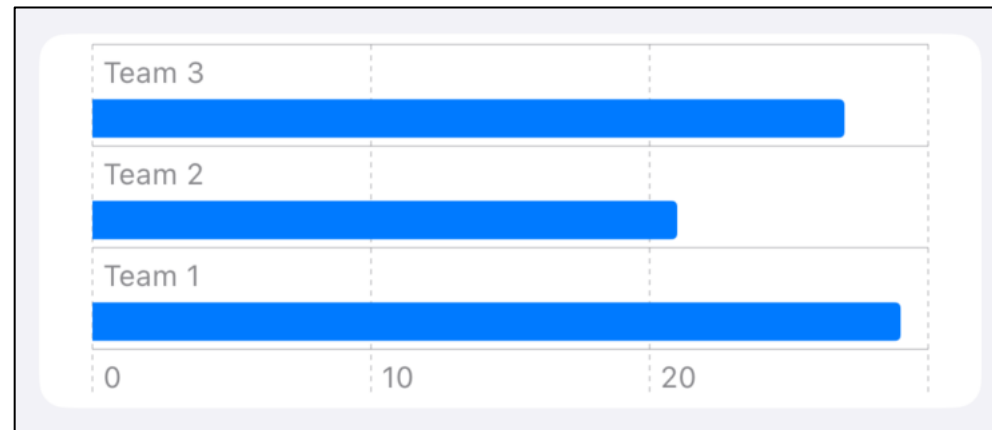
- Passen Sie im Core Data Model Editor die Entities an. Ersetzen Sie das Beispiel-Model durch zwei Entities **Person** und **Team**.
- Fügen Sie bei der **Team**-Entity folgende Properties hinzu:
 - **id** (UUID)
 - **name** (String)
 - **points** (Int)
 - optional weitere Parameter nach Belieben
- Fügen Sie bei der **Person**-Entity folgende Properties hinzu:
 - **name** (String)
 - optional weitere Parameter nach Belieben
- Erstellen Sie eine To-Many-Relationship zwischen den beiden Entities.

Core Data – User Interface

- Ersetzen Sie die Item-Liste durch eine Liste von Personen und eine Liste von Teams. Erstellen Sie für ein Team eine Detail-Seite mit allen Team-Mitgliedern.
- Für diese Übung können Sie 3 einfache Buttons erstellen, welche in Core Data «hardcoded» neue Objekte erstellen:
 - Eine einzelne Person
 - Ein Team mit zwei neuen Personen
 - Eine neue Person in zwei neuen Teams.
- Sie können die Properties der Objekte hardcoden, dynamischer Input ist nicht nötig. Als **id** bei den **Team**-Objekten setzen Sie **UUID()** (wichtig für den Chart).

Swift Charts

- Erstellen Sie einen einfachen Bar-Chart, der die **points** aller Teams darstellt. Die Punkte sollten als x-Values gesetzt werden und die id's als y-Values.
 - Wenn sie die Namen als y-Values setzen, funktioniert das auch, es werden aber Teams mit demselben Namen zusammengefasst.



Data Storage & Unit Tests

- Erstellen Sie eine neue Klasse `CodableFileWriter` und implementieren zwei statische Methoden zum Speichern und Laden von einer generischen Codable-Instanz als JSON-Datei.

```
class CodableFileWriter<T: Codable> {  
    static func write(_ content: T, to fileName: String) throws  
    { ... }  
  
    static func read(from fileName: String) throws → T  
    { ... }  
}
```

- Verwenden Sie `JSONEncoder` und `JSONDecoder`, um aus dem Codable ein Data-Objekt zu erstellen und speichern Sie das Resultat im Documents Directory.
- Erstellen Sie mindestens einen Unit Test, um beide Funktionen der Klasse zu prüfen.