JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

Mobile Access to Web Services

Systemtechnik Labor
5BHITT 2015/16, GruppeA

Klaus Ableitinger

Version 0.2
Begonnen am 21. April 2016
Beendet am 21. April 2016

Note:
Betreuer: Mi. Borko

Inhaltsverzeichnis

1 Einführung

Diese Ubung gibt einen Einblick in Entwicklungen von mobilen Applikationen.

1.1 Ziele

Das Ziel dieser Ubung ist eine Anbindung einer mobilen Applikation an ein Webservices. Die Anbindung soll mit Hilfe eines RESTful Webservice (Gruppe1) umgesetzt werden.

1.2 Voraussetzungen

- Grundlagen Java und XML
- Grundlegendes Verstandnis uber Entwicklungs- und Simulationsumgebungen
- Verstandnis von RESTful Webservices

1.3 Aufgabenstellung

Es ist eine mobile Anwendung zu implementieren, die sich an das Webservice aus der Ubung DezSysLabor-09 "Web Services in Javaänbinden soll. Dabei mussen die entwickelten Schnittstellen entsprechend angesprochen werden. Es ist freigestellt, welche mobile Implementierungsumgebung dafur gewahlt wird. Empfohlen wird aber eine Implementierung auf Android

1.4 Quellen

1. Android Restful Webservice Tutorial – How to call RESTful webservice in Android – Part 3"; Posted By Android Guru on May 27, 2014;
   online: `http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-`
2. Referenzimplementierung von DezSys09; Paul Kalauner;
   online: `https://github.com/pkalauner-tgm/dezsys09-java-webservices`

Bewertung: 16 Punkte

- Anbindung einer mobilen Applikation an die Webservice-Schnittstelle (6 Punkte) - Registrierung von Benutzern (3 Punkte)
- Login und Anzeige einer Willkommensnachricht (3 Punkte)
- Simulation bzw. Deployment auf mobilem Gerat (2 Punkte)
- Protokoll (2 Punkte)

2 Ergebnisse

Die App wurde in iOS mit der Programmiersprache Swift entwickelt.

2.1 Swift

Um Nochmal Swift zu wiederholen hab ich mir das offizielle Tutorial von Apple angeschaut (1).

Basics
In Swift können im allgemeinen redundante Symbole weggelassen werden; also Strichpunkte und runde Klammern bei `if`s, sowie `break`s in `switch` statements können weggelassen werden (funktioniert dabei ähnlich wie Python).

Swift ist stark typisiert, der Dateityp wird aber vom Compiler inferred; das heißt man kann, aber muss ihn nicht explizit dazu schreiben:

```
1 var x = 0
```

Listing 1: Normale Variablendefinition

```
var str: String = "asdf"
```

Listing 2: Variablendefinition mit definiertem Datentyp

Konstanten werden mit `let` definiert. Laut Konvention sollten alle Variablen, welche während des Programmablaufs nicht geändert werden sollten mit `let` deklariert werden. Der Compiler warnt auch, wenn er eine `var` Deklaration findet, welche nicht geändert wird.

```
let const = 123
```

Listing 3: Definition einer Konstanten

Funktionen
Funktionen sehen in Swift wie folgt aus:

```
func <name>(<parameters>) -> <return type> {
    <function body>
}
```

Listing 4: Allgemeine Funktionsdefinition

Die Rückgabetypdefinition kann, wenn es keinen Rückgabetyp gibt auch weggelassen werden; `-> Void` ist also nicht notwendig.

Class und Struct
In Swift gibt es einen Unterschied zwischen Klassen und Structs. Structs können prinzipiell mit ihrem Gegenstück in C verglichen werden, sie sind quasi eine Ansammlung an Variablen/Werten und haben keine Funktionen (abgesehen von einer init Funktion).

Klassen hingegen funktionieren wie in anderen Objektorientierten Programmiersprachen auch; Sie können Objekt-Funktionen haben und voneinander erben. Ein anderer wichtiger Unterschied ist, dass Structs *pass by value* und Klassen *pass by reference* sind.


```
struct SomeStruct {
    var name: String
    init(name: String) {
        self.name = name
    }
}
```

Listing 5: Beispiel Structdefinition

```
class SomeClass {
    var name: String
    init(name: String) {
        self.name = name
    }

    func someFunc() -> String {
        return self.name
    }
}
```

Listing 6: Beispiel Klassendefinition

```
class SomeSubClass: SomeClass {
    var subName: String
}
```

Listing 7: Beispiel Subklassendefinition von SomeClass

2.2 Implementierung

Literatur

[1] *iOS Developement Tutorial* Apple, 2016, online: `https://developer.apple.com/library/ios/referencelibrary/GettingStarted/DevelopiOSAppsSwift/`; zuletzt abgerufen am
Eric Weisstein, *Number Field Sieve* (dt. Zahlenkörpersieb), 2016, online: `http://mathworld.wolfram.com/NumberFieldSieve.html`; zuletzt abgerufen 31. März 2016.

