

Programmierung in iOS mit Swift Studiengang MI

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- <http://mwilhelm.hs-harz.de>
- Raum 2.202
- Tel. 03943 / 659 338

Gliederung

Überblick:

- **Einleitung, Geschichte, xcode**
- **Sprache**
 - elementare Datentypen
 - Variablen und Kontrollstrukturen
 - Arrays, Funktionen, Rekursionen
 - Objekte, etc.
- **Playground**
- **Grafische Oberfläche**
 - Einfache UI-Elemente
 - Layouts
 - Core-Data
 - Sensoren
- **Viele Beispiele**

Literatur

- iOS-Apps programmieren mit Swift
Christian Bleske
dpunkt.verlag
ISBN 978-3-86490-263-5



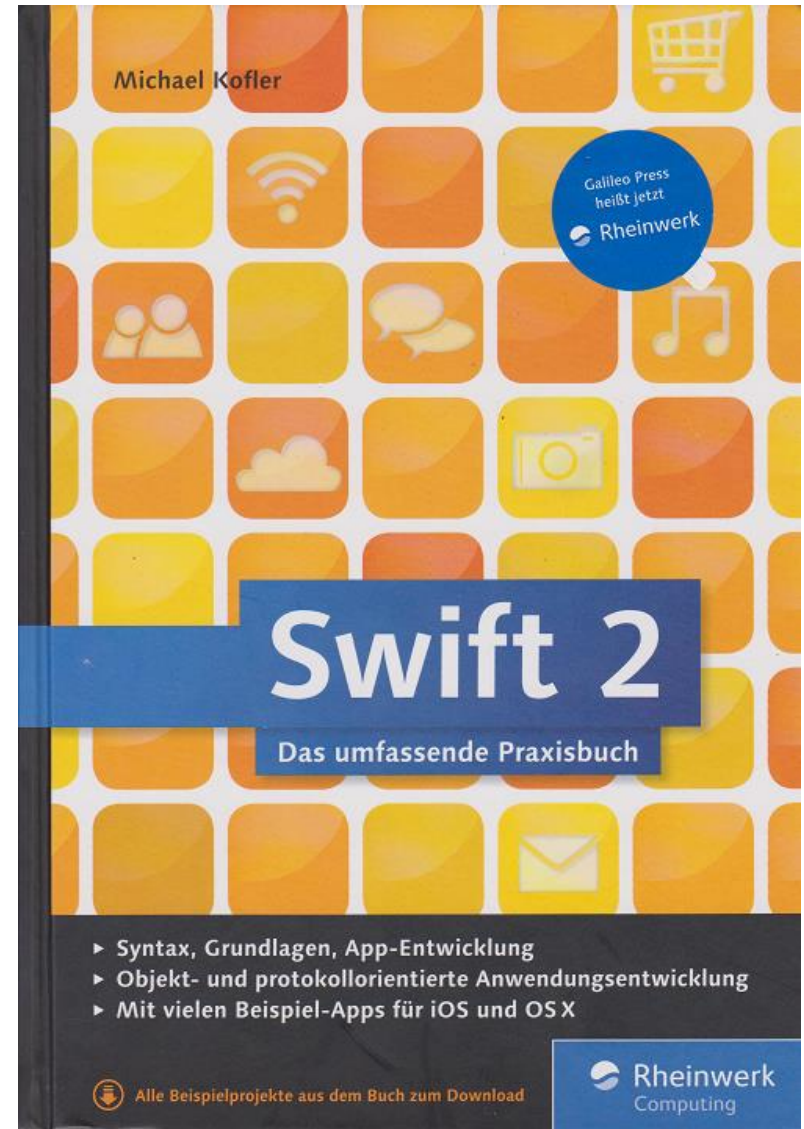
Literatur

■ Swift 2

Michael Kofler

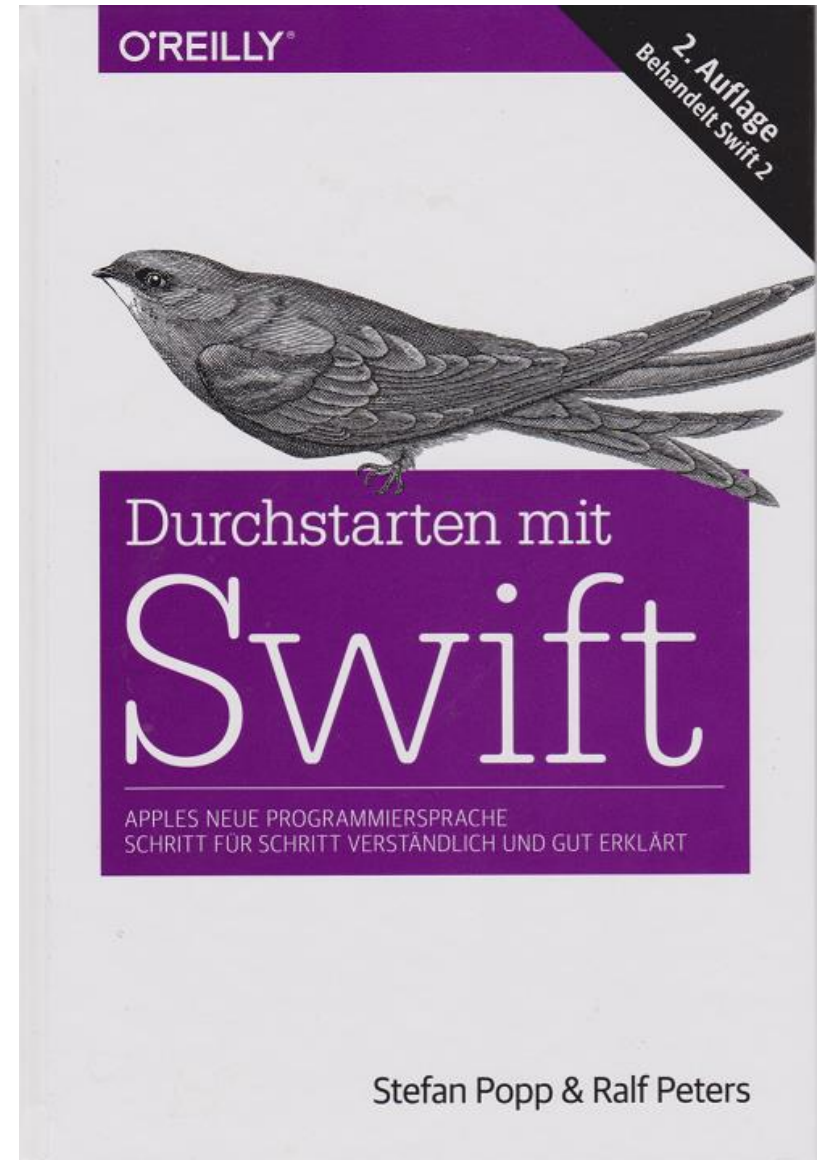
Rheinwerk

ISBN: 978-3-8362-3651-5



Literatur

- Durchstarten mit Swift
Stefan Popp & Ralf Peters
O'Reilly
ISBN: 978-3-96009-005-2



Literatur

- **Mac OS X 10.6 Snow Leopard**
- Leopard – einfach und wirkungsvoll
- ISBN: 978-3-908497-96-7
- Preis: 14,95 EUR

- **Das Mac-Buch für Windows-Umsteiger: Aktuell zu Mac OS X 10.5 Leopard (Galileo Design)**
- Jörg Rieger, Markus Menschhorn
- Preis: 19,90 EUR

- **Der Schnellkurs: Mac für Umsteiger**
- Prinz, Müller
- ISBN 978-3-8362-1016-4
- Preis: 6,95 EUR

Literatur

- **Mac OS X Leopard und Linux**
- ISBN 978-3-8362-1016-4

Links

- **Erste Schritte**
- <http://www.apple.com/de/findouthow/mac/#anatomy>
- **Vom PC zum Mac**
- <http://www.apple.com/de/findouthow/mac/#switcher>
- **Widget**
- <http://www.apple.com/downloads/dashboard/top50/>

Links

- **xcode:**

- <https://developer.apple.com/xcode/>

- **Erste Schritte**

- <http://www.apple.com/de/findouthow/mac/#anatomy>

- **MacEinsteiger**

- <http://www.maceinsteiger.de/>

- **Vom PC zum Mac**

- <http://www.apple.com/de/findouthow/mac/#switcher>

- **Widget**

- <http://www.apple.com/downloads/dashboard/top50/>

Vorlesung

- Vorlesung per Powerpoint
- Beispiele an der Tafel
- Beispiele auf der Homepage
- Beispiele während der Vorlesung, am Rechner
- Note:
 - mündliche Prüfung
 - Projektarbeit
- „Musterklausur“ am Ende der Vorlesung, Ende Juni
- Klausurvorbereitung

Programmierung unter Mac

Objective-C

- Vor 2014 die einzige native Sprache, Java geht fast immer
- Objective-C ist die Sprache für Cocoa (OS X)
- Objective-C++ erweitert die Sprache um Objektorientierung
- orientiert sich an Smalltalk
- Etwas krude Syntax, Mac: leichte Bedienung, schwere Programmierung

- `NSString *urlString=@"http://":`
- `urlString = [urlString stringByAppendingString:Id];`
- `NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"%@",urlString];`
- `NSURLRequest *requestObj = [NSURLRequest requestWithURL:url];`
- `[uiWebView loadRequest:requestObj];`

Swift

- Vorgestellt im Jahr 2010 von Chris Lattner als Open-Source
- Grundlage war das Forschungsprojekt LLVM (Low Level Virtual Maschine)
 - modularer quelloffener Compiler
- 2007 baute Chris Lattner ein Frontend für C/C++ und Objective-C
- 2010 wurde xcode entwickelt
- Ende 2012 stellte Lattner sein Projekt Apple vor
- Schwächen von Objective-C
 - Basis ist C (Pointer, Speicher erzeugen, freigeben, Headerdateien)
 - Keine Typüberprüfung
 - Aufsatz C++
 - Mischung von C, C++ und Objective-C
 - `id obj = @[@"string1", @"string2"];`
 - `int i = [obj length];`
 - `NSArray *array = obj;`

Objective-C

Objective-C:

```
[string stringByReplacingOccurenceOfString:@"Hello" withString:@"Hi"];
```

Java:

```
string.replaceAll("Hello","Hi")
```

// einer String-Variable wird eine URL zugewiesen, Browser-Aufruf

```
NSString *urlString="http://";
```

```
urlString [urlString stringByAppendingString:Id];
```

```
NSURL *url = [NSURL URLWithString:[NSString
```

```
stringWithFormat:@"%@",urlString]];
```

```
NSURLRequest *requestObj = [NSURLRequest requestWithURL:url];
```

```
[uiWebView loadRequest:requestObj];
```

Objective-C

// + **Klassenmethoden**

+ (id) calculator;

+ (id) calculatorWithValue:(int)value;

// - **Instanzmethoden**

- (id) initWithValue:(int)no;

- (void) increment;

- (void) add:(int)no;

- (void) addNumber1:(int)param1 number2:(int)param2;

- (int) value;

- (void) setValue:(int)newValue;

Programmierung unter Mac mit Swift

Swift

- Vorgestellt im Jahr 2014 durch Herrn Chris Lattner
- Nachfolger von Objective-C
- Schwächen von Objective-C werden vermieden
- Orientiert sich an Rust, Haskell, Ruby, Python, C#, CLU und D
- Gegenüber Java ist einiges anders
- Open Source
- Xcode, aktuell Version 7.0
 - xcode 1.0 wurde im Herbst 2003 vorgestellt.
 - Neue IDE
 - Unterstützt alle Geräte (Mac, iPhone, iPad)
 - Mit Simulator
 - Viele Vorlagen
 - Unterstützt iOS, OS X und Linux

Eigenschaften von Swift

- Keine Pointer,
- Keine Header-Dateien
- Laufzeittypisierung (var dummy=12.45)
- Generics, Templates, Arrays, Dictionaries, Sets
- Mehrere Parametertypen (value,inout,benannte Parameter, bel. viele)
- Tupels (mehrfache Rückgabewerte, auch benannt)
- String Methoden (etc.)
- Enumeration
- Klassen und Protokolle (Interface)
- Funktionale Programmierung (nicht deklarativ)
- try catch
- Neue IDE
- Simulator, Playground
- UI-Vorlagen (MVC)

Programmierung unter Mac mit Swift

■ Welcher Mac genügt

- iMac ab Mitte 2007
- MacBook (13" Aluminium ab Ende 2009)
- MacBook (13", ab Anfang 2009)
- MacBook Pro (13" Aluminium ab Mitte 2009)
- MacBook Pro (15" oder 17" ab Mitte 2007)
- MacBook Air (ab Ende 2008)
- Mac mini ab Anfang 2009
- Mac Pro (ab Anfang 2008)

■ Welche OS-Version

- Mavericks
- Yosemite
- El Capitan

Beispiel mit Swift

```
var implicitInteger = 70
```

```
var implicitDouble = 70.0
```

```
var explicitDouble: Double = 70
```

```
let apples = 3
```

```
let oranges = 5
```

```
let appleSummary = "Ich habe \ \(apples) Äpfel."
```

```
let fruitSummary = "Ich habe \ \(apples + oranges) Früchte."
```

```
print("Hallo Welt!")
```

```
let people = ["Anna": 67, "Julia": 8, "Hans": 33, "Peter": 25]
```

```
for (name, age) in people {
```

```
    print("\ \(name) ist \ \(age) Jahre alt.")
```

```
}
```

Beispiel mit Swift

```
var implicitInteger = 70
```

varianter Datentyp

```
var implicitDouble = 70.0
```

```
var explicitDouble: Double = 70
```

Double Datentyp, Pascal

```
let apples = 3
```

Basic

```
let oranges = 5
```

```
let appleSummary = "Ich habe \ \(apples) Äpfel."
```

Variable: apples (%d)

```
let fruitSummary = "Ich habe \ \(apples + oranges) Früchte."
```

```
print("Hallo Welt!")
```

```
let people = ["Anna": 67, "Julia": 8, "Hans": 33, "Peter": 25]
```

Array

```
for (name, age) in people {
```

foreach

```
    print("\ \(name) ist \ \(age) Jahre alt.")
```

Variable: name, age

```
}
```

Aufbau des OS X

AQUA

Carbon

Cocoa

Java

Quartz

OpenGL

QuickTime

DARWIN