

Properties

Pühringer Lukas

Unsere Themen

- Stored Properties
- Computed Properties
- Lazy Properties
- Property Observer
- Read only Properties

Stored Properties

- Haben Instanzvariable
- Deklaration als Type Property
- In Objective C:
 - Im Setter andere Properties aktualisieren oder das Property
 - Methodenaufruf, falls Property sich ändert

Stored Properties

- In Swift:
 - Computed Properties
 - Lazy Stored Properties
 - Property Observers

```
class niceClass {  
    var someText = "text"  
    var otherText = "more text"  
}
```

Computed Properties

- Objective C hatte das auch schon
- Keine zugrundeliegende Variable
- Dient um andere Properties zu ändern
- Auch Properties in einem anderen Property zu aktualisieren
- Accessor Methoden in Objective C = get/set in Swift

Computed Properties

- Deklaration nicht als Type Properties

```
class greatClass {  
    var someText = "text"  
    var otherText : String {  
        get {  
            return someText  
        }  
        set (newText) {  
            someText = newText  
        }  
    }  
}
```

Lazy Properties

- lazy keyword: wird just in time kompiliert
- Verhalten aufgrund von fehlender get/set nachbilden
- Wird erst initialisiert wenn es das erste mal gelesen wird

```
class someClass {  
    var someText = "text"  
    lazy var someView = UIView() // or some really expensive class  
}  
  
var mc = someClass()  
mc.someText  
mc.someView // someView initialized
```

Property Observer

- Swift kein Key/Value Observer
- Property hat ähnlichen Effekt
- willSet und didSet für Zugriff auf neuen und alten Wert
- Swift Weg um Methoden bei Property Änderung aufzurufen

```
class aClass {  
    var text: NSString? {  
        willSet {  
            // do something  
            newValue  
        }  
        didSet {  
            // do something  
            oldValue  
        }  
    }  
  
    var text2: NSString? {  
        willSet (shinyNew) {  
            // do something  
            shinyNew  
        }  
        didSet (oldVersion) {  
            // do something  
            oldVersion  
        }  
    }  
}
```


Read only Properties

- Objective C Modifier readonly nicht in Swift
- In Swift: Property private deklarieren und mit Computed Property zugreifen
- Effekt: von außen keine Änderung

Beispiel

- Schreibe eine Klasse Kreis, die vielleicht lazy sein darf. Die Klasse soll eine Farben Property und eine Init-Methode haben. Die Init Methode soll die Farbe setzen.
- Setze danach die Farben auf drei Werte und gib die aus.

Lösung

```
class Square {  
    var color: String  
    init(color: String) {  
        // Initialize the property.  
        self.color = color  
    }  
}  
  
// Create instance of class.  
var test = Square(color: "blue")  
  
// Print property value.  
print(test.color)  
  
// Reassign property and print it again.  
test.color = "red"  
print(test.color)
```