

iOs - Praktikum

Aufgabe 12

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.

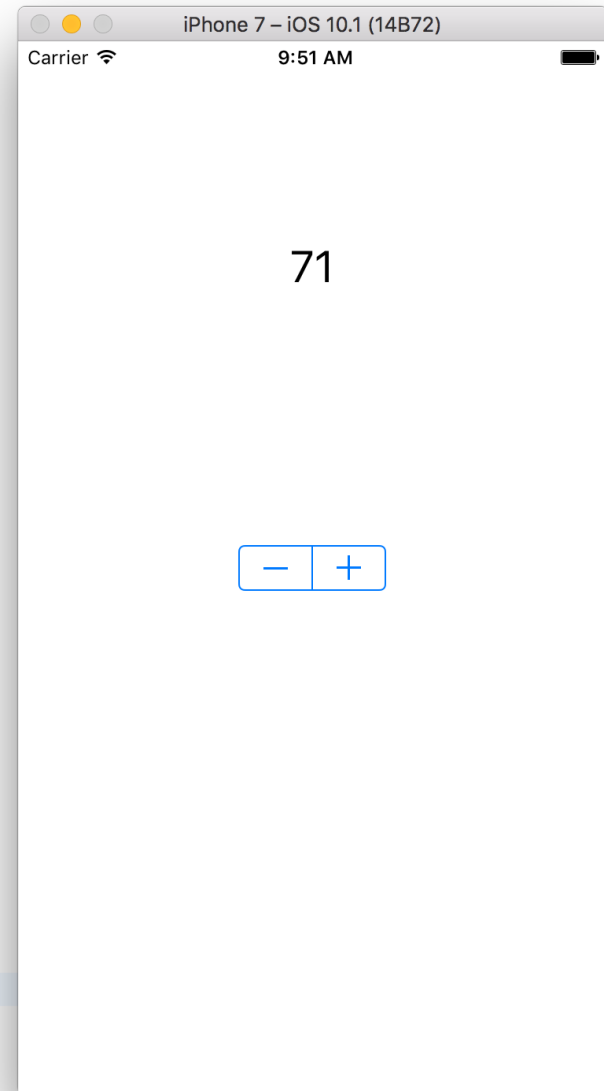
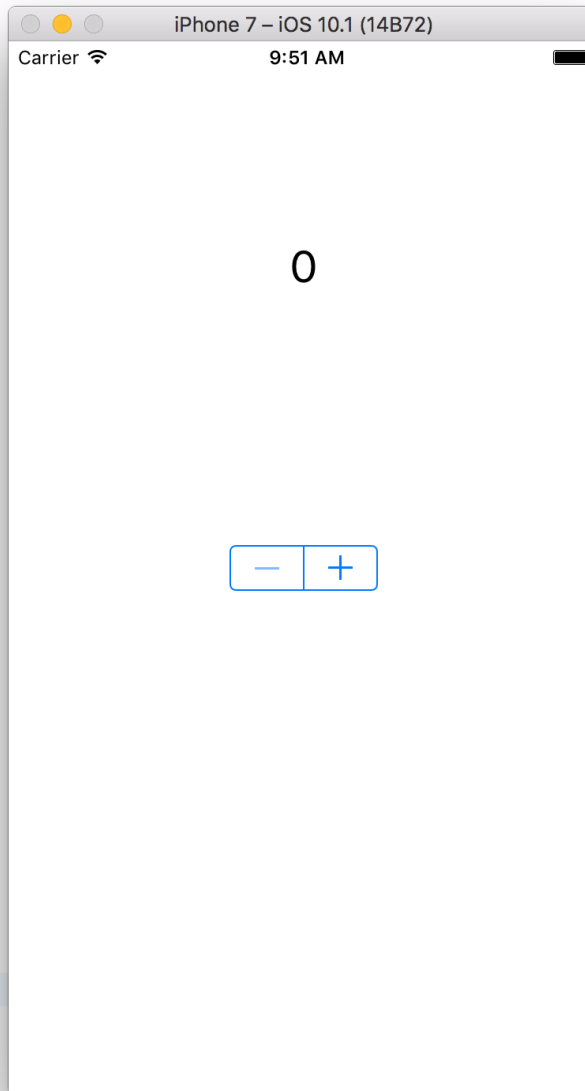
Gehen Sie dabei auf das Schlüsselwort `dynamic` ein und was nötig ist, um damit KVO realisieren zu können. Vergleichen Sie die KVO von Swift mit der von Objective-C insbesondere bezüglich der Typisierung der beobachteten Variablen. Erwähnen Sie auch mögliche Alternativen zu dem oben genannten Schlüsselwort.

dynamic

- mit dem Schlüsselwort `dynamic` wird eine Variable für den KVO-Mechanismus von Objective-C beobachtbar gemacht
- dazu muss die Klasse von `NSObject` erben => Enum, Generics können nicht mit `dynamic` versehen werden
- `addObserver()` fügt Beobachter hinzu, der bei Änderungen die `observeValue()`-Methode aufruft

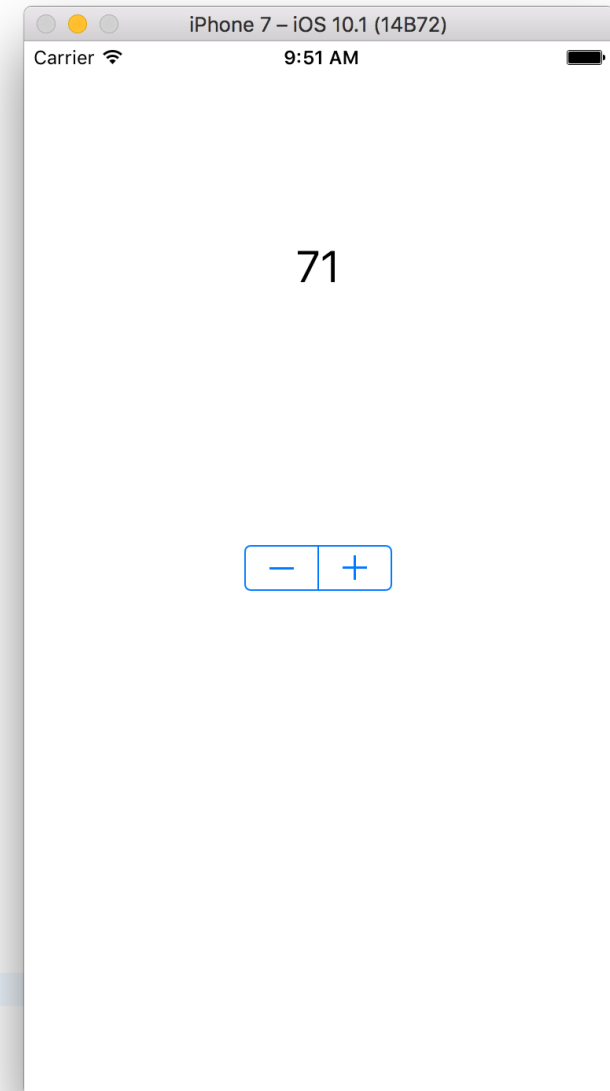
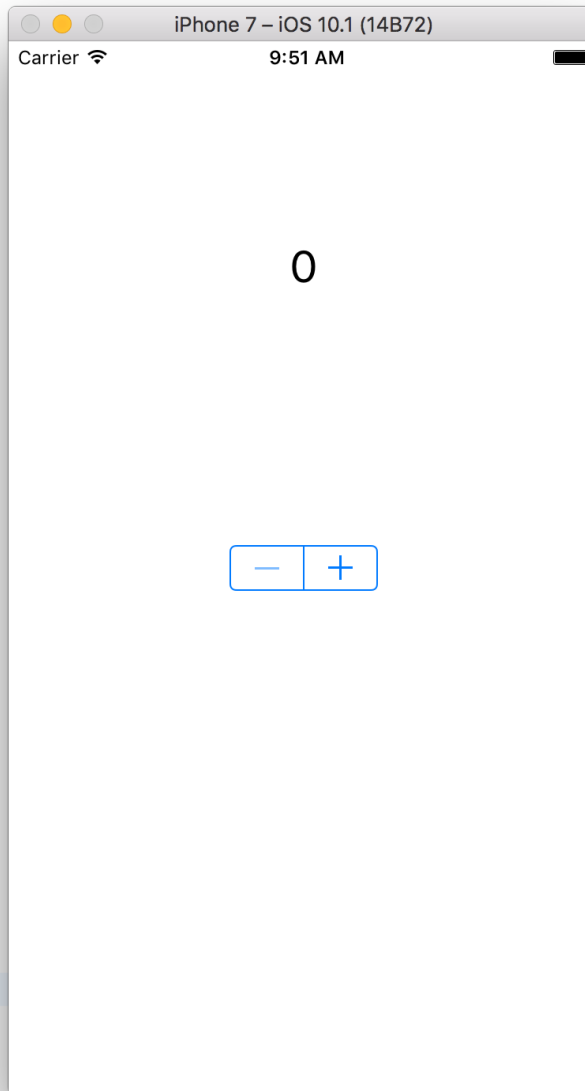
Beispiel in Swift

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     dynamic var counter: Int = 0
14
15
16     @IBOutlet weak var label: UILabel!
17     @IBAction func stepper(_ sender: UIStepper) {
18         counter = Int(sender.value)
19     }
20
21     override func viewDidLoad() {
22         super.viewDidLoad()
23
24         addObserver(self, forKeyPath: #keyPath(counter), options: [.initial], context: nil)
25     }
26
27
28     override func observeValue(forKeyPath keyPath: String?, of object: Any?, change: [NSKeyValueChangeKey : Any]?, context: UnsafeMutableRawPointer?) {
29         if keyPath == #keyPath(counter) {
30             label.text = "\(counter)"
31         }
32     }
33     override func didReceiveMemoryWarning() {
34         super.didReceiveMemoryWarning()
35         // Dispose of any resources that can be recreated.
36     }
37
38
39 }
40
41
```



Beispiel in Swift

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     dynamic var counter: Int = 0
14
15
16     @IBOutlet weak var label: UILabel!
17     @IBAction func stepper(_ sender: UIStepper) {
18         counter = Int(sender.value)
19     }
20
21     override func viewDidLoad() {
22         super.viewDidLoad()
23
24         addObserver(self, forKeyPath: #keyPath(counter), options: [.initial], context: nil)
25     }
26
27
28     override func observeValue(forKeyPath keyPath: String?, of object: Any?, change: [NSKeyValueChangeKey : Any]?, context: UnsafeMutableRawPointer?) {
29         if keyPath == #keyPath(counter) {
30             label.text = "\(counter)"
31         }
32     }
33     override func didReceiveMemoryWarning() {
34         super.didReceiveMemoryWarning()
35         // Dispose of any resources that can be recreated.
36     }
37
38
39 }
40
41
```



```
[bankInstance addObserver:personInstance
forKeyPath:@"accountBalance"
options:NSKeyValueObservingOptionNew
context:NULL];
```

```
-(void) observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object
change:(NSDictionary *)change context:(void *)context
{
    // Custom Implementation
    if ([keyPath isEqualToString:@"accountBalance"] ) {
        // do something... e.g. update accountBalance Information
    }
}
```

Alternativen zu *dynamic*

```
class Observable<T> {  
  
    let didChange = Event<(T, T)>()  
    private var value: T  
  
    init(_ initialValue: T) {  
        value = initialValue  
    }  
  
    func set(newValue: T) {  
        let oldValue = value  
        value = newValue  
        didChange.raise(oldValue, newValue)  
    }  
  
    func get() -> T {  
        return value  
    }  
}
```

```
class Car {  
    let miles = Observable<Int>(0)  
    let name = Observable<String>("Turbo")  
}
```

```
func viewDidLoad() {  
    var car = Car();  
    car.name.didChange.addHandler(self, handler: ViewController.nameDidChange)  
    car.name.set("Speedy")  
    println("The car is now called \(car.name.get())")  
}  
  
func nameDidChange(oldValue: String, newValue: String) {  
    println("Name changed from \(oldValue) to \(newValue)")  
}
```

Quellen

- <https://www.ralfebert.de/snippets/ios/swift-key-value-observer/>
- <http://blog.scottlogic.com/2015/02/11/swift-kvo-alternatives.html>