

Archiving in Swift

Aufgabe 13

Marcel Zickler

13. Dezember 2016

Aufgabe

- SwiftCoding Interface anlegen mit `encode` und `init`

Aufgabe

- SwiftCoding Interface anlegen mit `encode` und `init`
- SwiftKeyedArchiver Klasse anlegen mit Serialisierungsmethode, die Daten persistent speichert

- `SwiftCoding` Interface anlegen mit `encode` und `init`
- `SwiftKeyedArchiver` Klasse anlegen mit Serialisierungsmethode, die Daten persistent speichert
- Alle Objekte die `SwiftCoding` implementieren, können ihre Variablen in Datenstruktur abspeichern, die serialisiert wird.

- `SwiftCoding` Interface anlegen mit `encode` und `init`
- `SwiftKeyedArchiver` Klasse anlegen mit Serialisierungsmethode, die Daten persistent speichert
- Alle Objekte die `SwiftCoding` implementieren, können ihre Variablen in Datenstruktur abspeichern, die serialisiert wird.
- Funktionalität durch Speichern und Wiederherstellen einer Klassenstruktur mit zwei Vererbungshierarchien zeigen

SwiftCoding-Interface

```
import Foundation

protocol SwiftCoding{

    init(withDict dict: ArchiveDict)
    func encode(withDict dict: inout ArchiveDict)

}
```

SwiftKeyedArchiver

```
11 public class SwiftKeyedArchiver{
12
13     class func archive(rootObject: SwiftCoding, toFile file: String) throws -> Bool{
14         //create a dict and pass it as a reference to the object to be serialized
15         var dict = ArchiveDict()
16         rootObject.encode(withDict: &dict)
17         //after all instance variables were added to the dict, they can be written to a plist
18         return dict.write(toFile: file) //returns true, if successfull
19     }
20
21     class func restore(fromFile file: String) -> SwiftCoding?{
22         if let dict = ArchiveDict(contentsOfFile: file){ //converts plist to dict
23             if(dict.getInt(forKey: Car.KEY_HP) != nil){
24                 //is a car
25                 return Car(withDict: dict)
26             }else if(dict.getInt(forKey: Bicycle.KEY_FRAME_SIZE) != nil){
27                 //is a bicycle
28                 return Bicycle(withDict: dict)
29             }else if(dict.getInt(forKey: Vehicle.KEY_NUMBER_OF_WHEELS) != nil){
30                 //is a vehicle
31                 return Vehicle(withDict: dict)
32             }
33         }
34         return nil
35     }
36 }
37 }
```

Car: Vehicle: SwiftCoding

```
26 init(manufacturer: String, model: String, hp: Int, maxKph: Int, licensePlate: String) {
27     self.hp = hp;
28     self.maxKph = maxKph;
29     self.licensePlate = licensePlate
30     super.init(manufacturer: manufacturer, model: model, numberOfWheels: 4)
31 }
32
33 //have to get all items from the dict and initialize this class with them
34 required convenience init(withDict dict: ArchiveDict) {
35     if let lp = dict.getString(forKey: Car.KEY_LICENSE_PLATE),
36         let kph = dict.getInt(forKey: Car.KEY_MAX_KPH),
37         let hp = dict.getInt(forKey: Car.KEY_HP),
38         let man = dict.getString(forKey: Car.KEY_MANUFACTURER),
39         let mod = dict.getString(forKey: Car.KEY_MODEL){
40         self.init(manufacturer: man, model: mod, hp: hp, maxKph: kph, licensePlate: lp)
41     }else{
42         fatalError("cannot init with dict, because some key is missing")
43     }
44 }
45
46 //encode all variables from parent and then its own
47 override func encode(withDict dict: inout ArchiveDict){
48     super.encode(withDict: &dict)
49     dict.putString(licensePlate, forKey: Car.KEY_LICENSE_PLATE)
50     dict.putInt(hp, forKey: Car.KEY_HP)
51     dict.putInt(maxKph, forKey: Car.KEY_MAX_KPH)
52 }
53
54 }
```


ArchiveDict

- Datenstruktur, in die Instanzvariablen für Serialisierung gespeichert werden
- Verwaltet eine NSDictionary
- Erlaubt nur Ints und Strings

```
40 //casts to NSNumber, because plists don't allow Ints
41 public func putInt(_ value: Int, forKey key: String){
42     dict.setValue(NSNumber(value: value), forKey: key)
43 }
44
45 public func putString(_ value: String, forKey key: String){
46     dict.setValue(value, forKey: key)
47 }
48
49 public func write(toFile file: String) -> Bool{
50     return dict.write(toFile: file, atomically: true)
51 }
52
53 public func getInt(forKey key: String) -> Int?{
54     return dict.value(forKey: key) as? Int
55 }
56
57 public func getString(forKey key: String) -> String?{
58     return dict.value(forKey: key) as? String
59 }
60
```

Beispiel

[illegible]

Carrier

5:41 PM

< Master

Vehicle

Car

Bicycle

Specialized

Venge

Frame size

Bicycle type

Save

Cancel

„Venge“	Vengeance	Vengeful
q w e r t z u i o p ü		
a s d f g h j k l ö ä		

Klick auf Save erstellt das jeweilige Objekt..

```
48 @IBAction func saveOnClick(_ sender: AnyObject) {
49     let obj: SwiftCoding
50     switch selectSC.selectedSegmentIndex {
51         //create a obj object with data from boxes. When required boxes are empty an error will
           occur. Also, it is not checked whether the user only typed number literals when casting
           to int
52         case 1:
53             obj = Car(manufacturer: manufacturerTv.text!, model: modelTv.text!, hp: Int(hpTv.text!)!,
                        maxKph: Int(maxKphTv.text!)!, licensePlate: licensePlateTv.text!)
54         case 2:
55             obj = Bicycle(manufacturer: manufacturerTv.text!, model: modelTv.text!, type:
                            bicycleTypeTv.text!, frameSize: Int(frameSizeTv.text!))
56         default:
57             obj = Vehicle(manufacturer: manufacturerTv.text!, model: modelTv.text!, numberOfWheels:
                            Int(numWheelsTv.text!))
58     }
59     archive(obj)
60     onSave(obj) //adds object to list in MasterViewController
61     self.navigationController?.popViewController(animated: true)
62 }
63 }
```

Beispiel

...und archiviert es

```
64 //archives the object created in this view controller
65 private func archive(_ obj: SwiftCoding){
66     //generate path for plist with "unique" id for this object
67     let documentDirectory = NSSearchPathForDirectoriesInDomains(.documentDirectory, .userDomainMask,
68         true)[0] as String
69     let path = documentDirectory.appending("/serialized_\n
70         (CreateObjectViewController.archiveCounter).plist")
71     //the actual archiving
72     do{
73         let success = try SwiftKeyedArchiver.archive(rootObject: obj, toFile: path)
74         if(success){
75             print("saved at \((path)")
76             CreateObjectViewController.archiveCounter += 1
77         }
78     } catch let error{
79         print("could not archive object: \(error.localizedDescription)")
80     }
81 }
```

Beispiel

Beendet man die App und startet sie erneut, werden die Objekte wieder hergestellt...

```
33 private func restoreAll(){
34     //generate paths for plists, as long as they exist
35     var archiveCounter = 0
36     let documentDirectory = NSSearchPathForDirectoriesInDomains(.documentDirectory, .userDomainMask,
37         true)[0] as String
38     var path = documentDirectory.appending("/serialized_\(archiveCounter).plist")
39
40     let fileManager = FileManager.default
41     while fileManager.fileExists(atPath: path){
42         //add restored objects to list
43         if let obj = SwiftKeyedArchiver.restore(fromFile: path){
44             addObject(obj)
45             archiveCounter += 1
46             path = documentDirectory.appending("/serialized_\(archiveCounter).plist")
47         }
48     }
49 }
```

Beispiel

...Und alle Objekte erscheinen in der Liste

