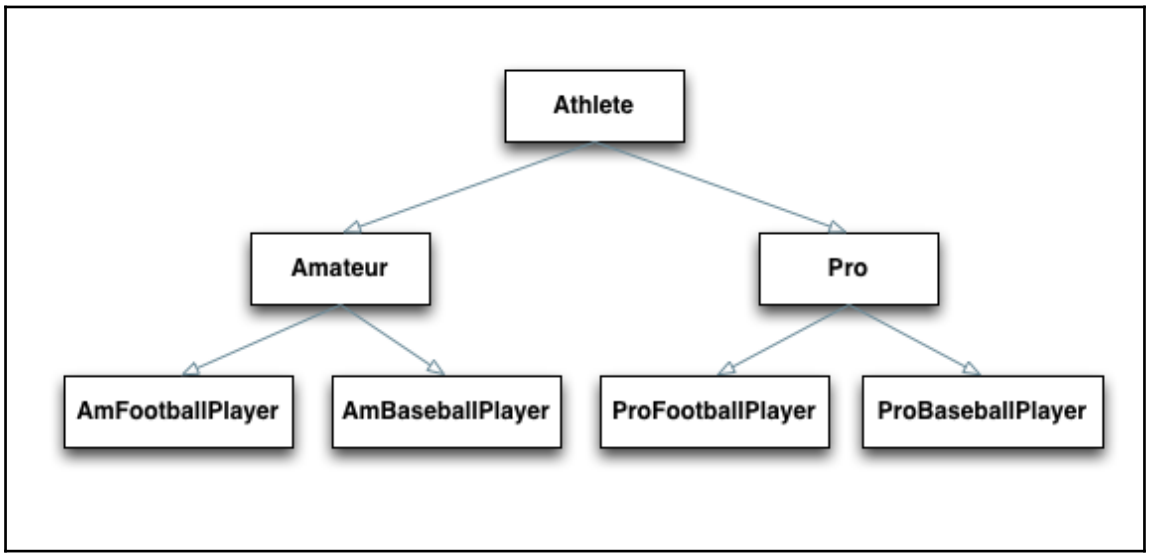
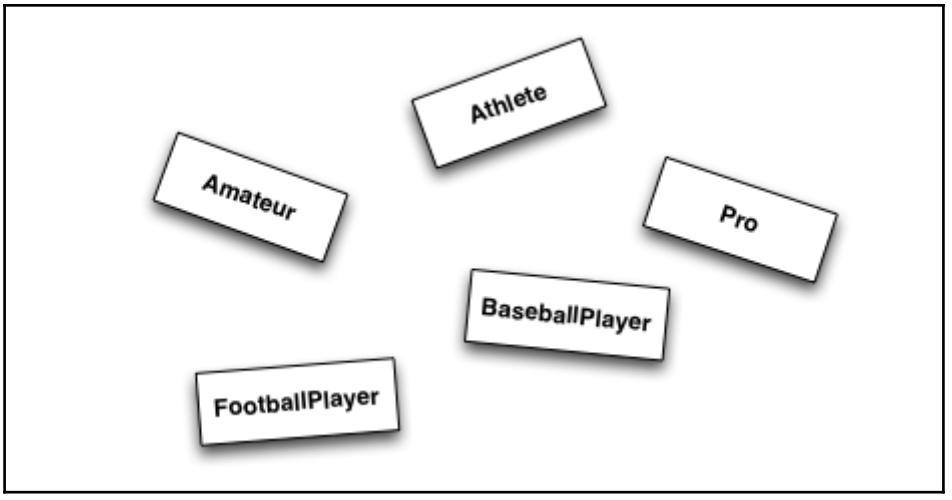
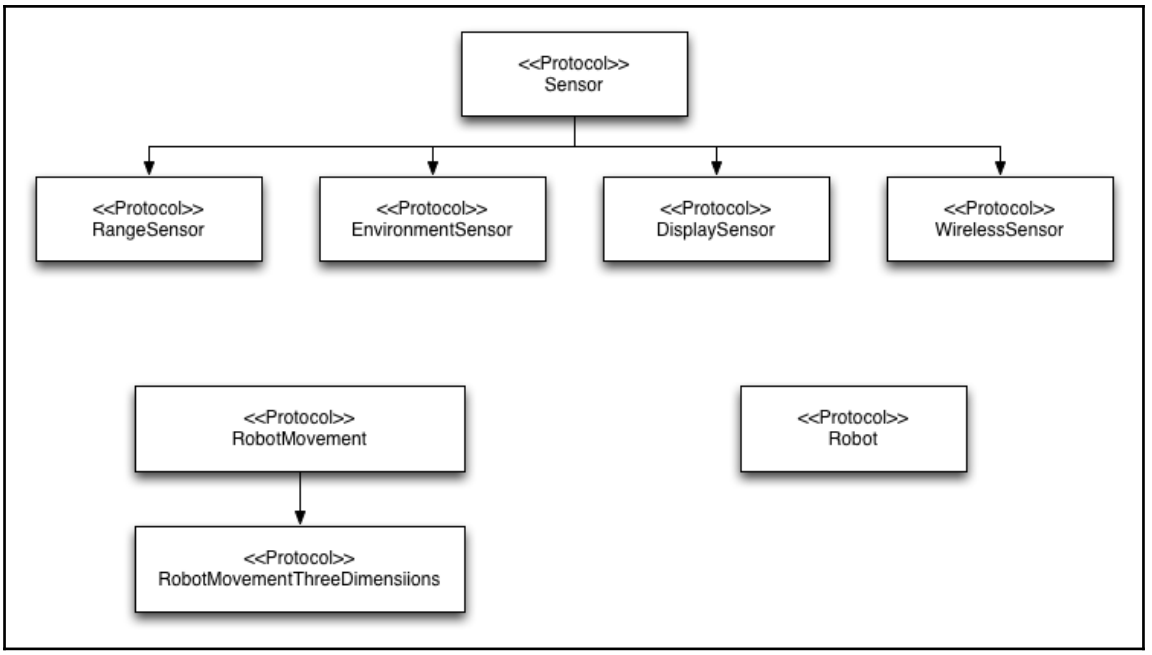
Chapter 1:Starting with the Protocol



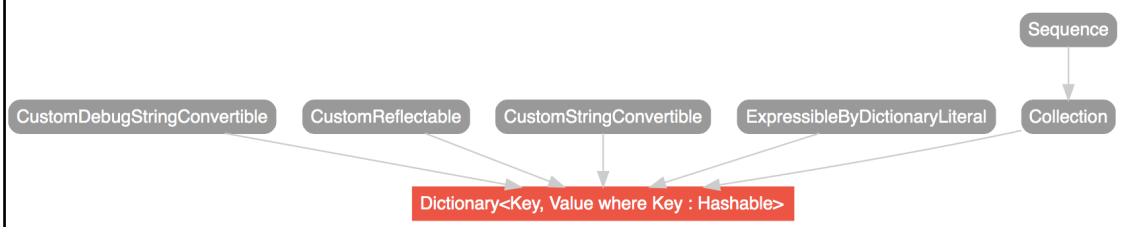




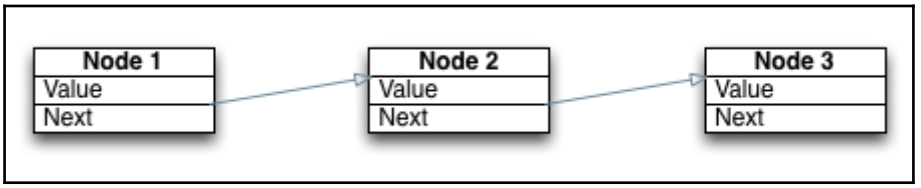
Inheritance `Collection`, `CustomDebugStringConvertible`, `CustomReflectable`, `CustomStringConvertible`, `ExpressibleByDictionaryLiteral`, `Sequence`

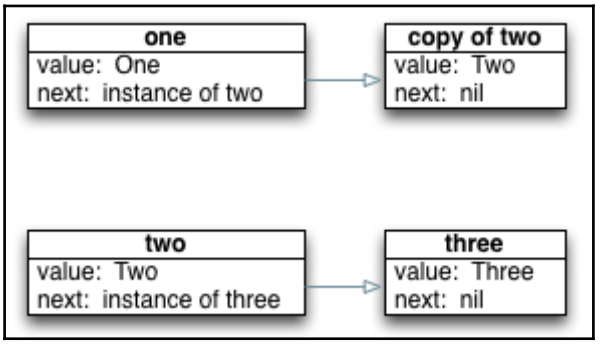
VIEW PROTOCOL HIERARCHY →

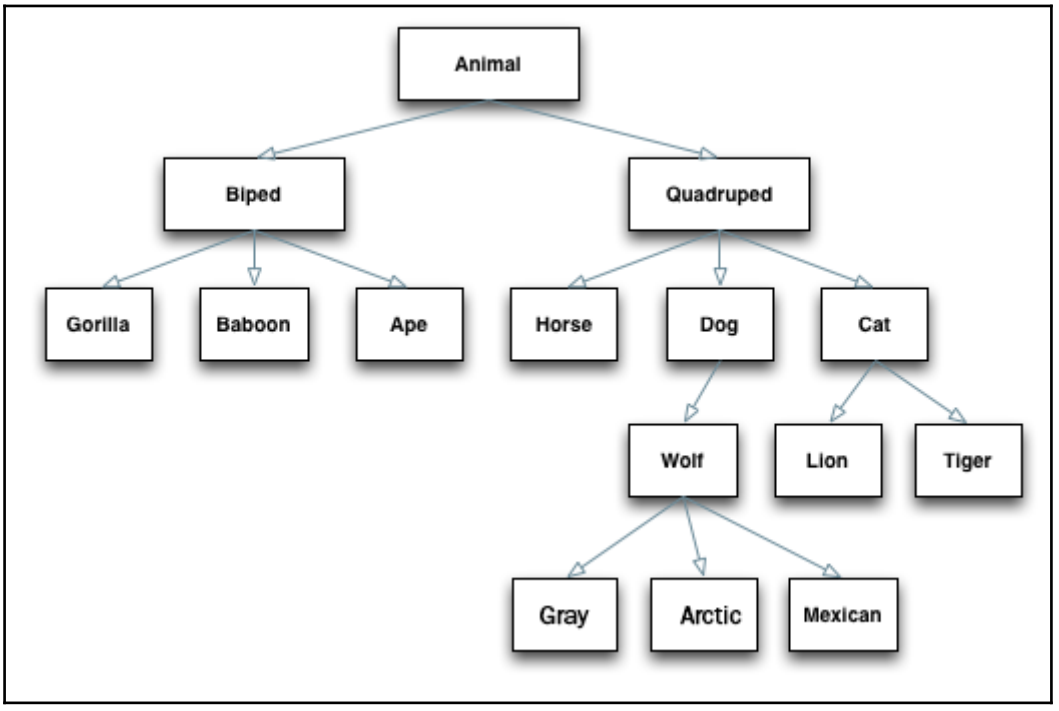
SwiftDoc.org



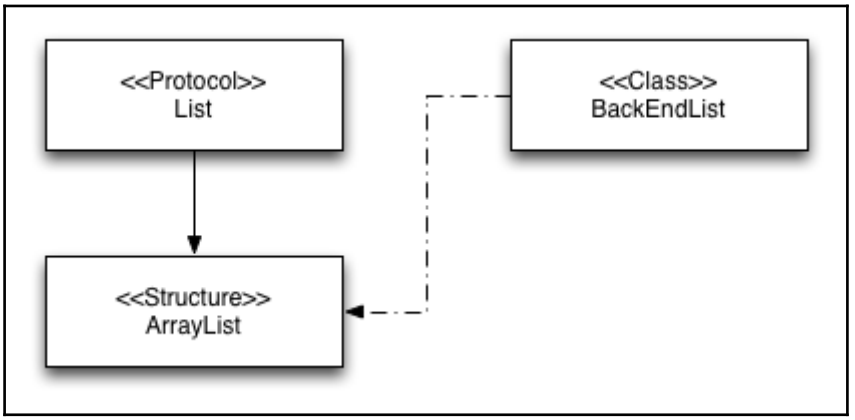
Chapter 2:Our Type Choices

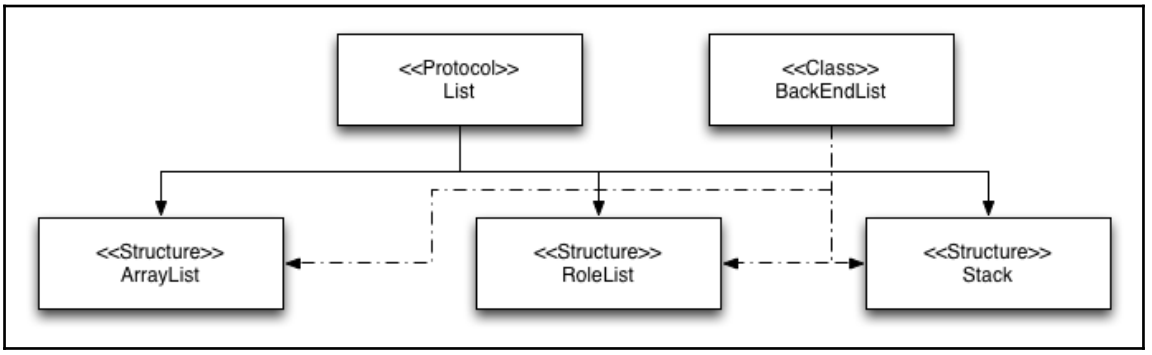






Chapter 4:Generics





SwiftDoc.org

Search

ON THIS PAGE
Declaration
Initializers
Instance Variables
Subscripts
Instance Methods
Conditionally Inherited Items

DISPLAY
Expand all [a]
Hide inherited (91) [i]

SWIFT VERSION
Swift 3.1 (Xcode 8.3b2)

Array

```
struct Array<Element>
```

An ordered, random-access collection.

Arrays are one of the most commonly used data types in an app. You use arrays to organize your app's data. Specifically, you use the `Array` type to hold elements of a single type, the array's `Element` type. An array can store any kind of elements---from integers to strings to classes.

Swift makes it easy to create arrays in your code using an array literal: simply surround a comma separated list of values with square brackets. Without any other information, Swift creates an array that includes the specified values, automatically inferring the array's `Element` type. For example:

```
// An array of 'Int' elements
let oddNumbers = [1, 3, 5, 7, 9, 11, 13, 15]

// An array of 'String' elements
let streets = ["Albemarle", "Brandywine", "Chesapeake"]
```

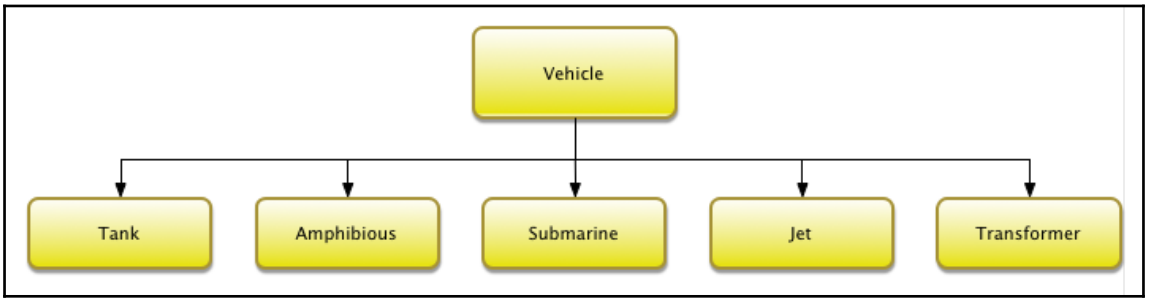
You can create an empty array by specifying the `Element` type of your array in the declaration. For example:

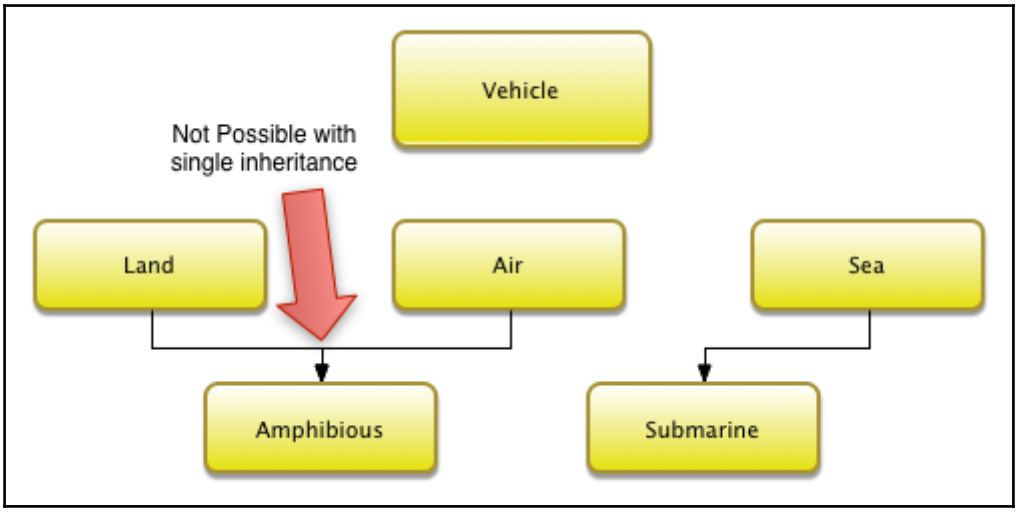
```
// Shortened forms are preferred
var emptyDoubles: [Double] = []

// The full type name is also allowed
var emptyFloats: Array<Float> = Array()
```

If you need an array that is preinitialized with a fixed number of default values, use the `Array(repeating:count:)` initializer.

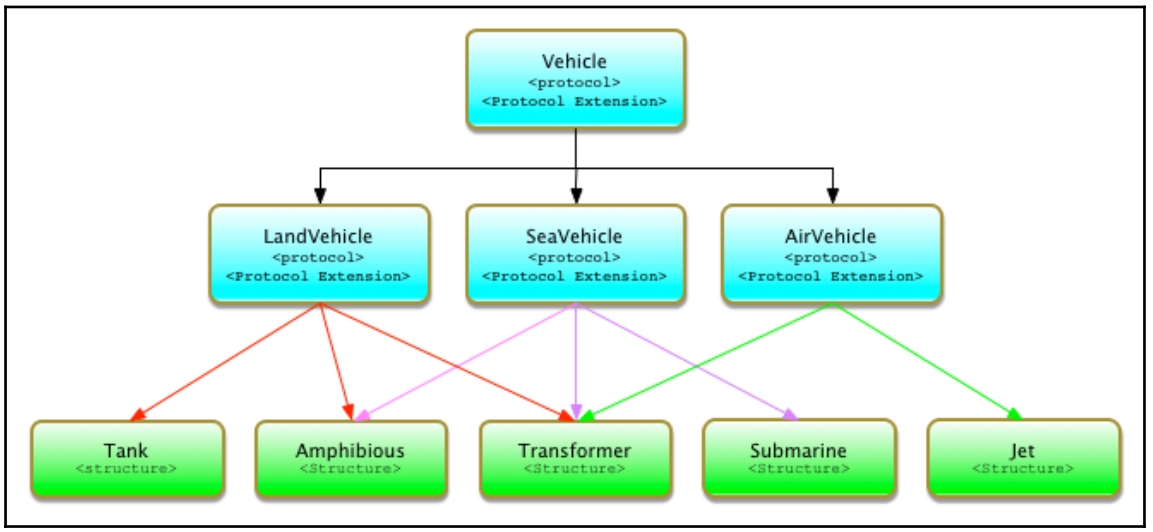
```
var digitCounts = Array(repeating: 0, count: 10)
print(digitCounts)
// Prints "[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]"
```

Chapter 5:Object-Oriented Programming

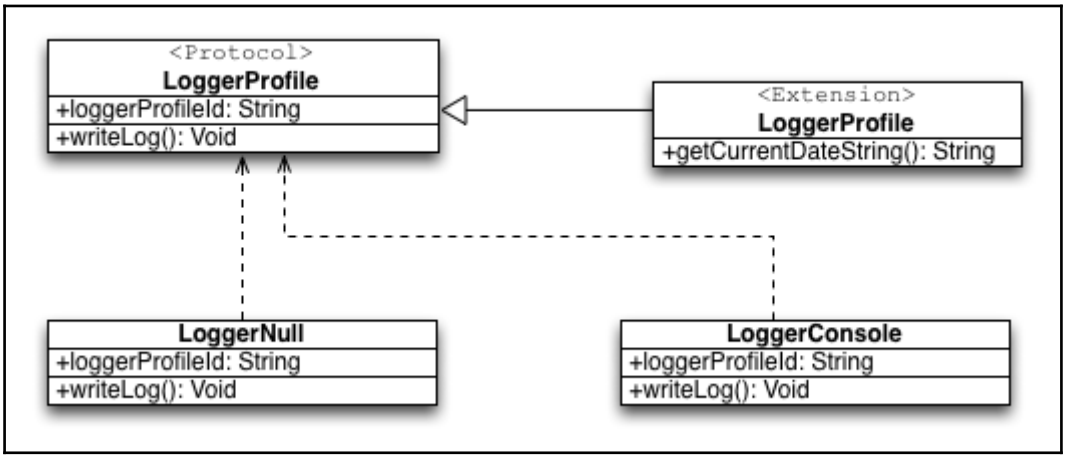


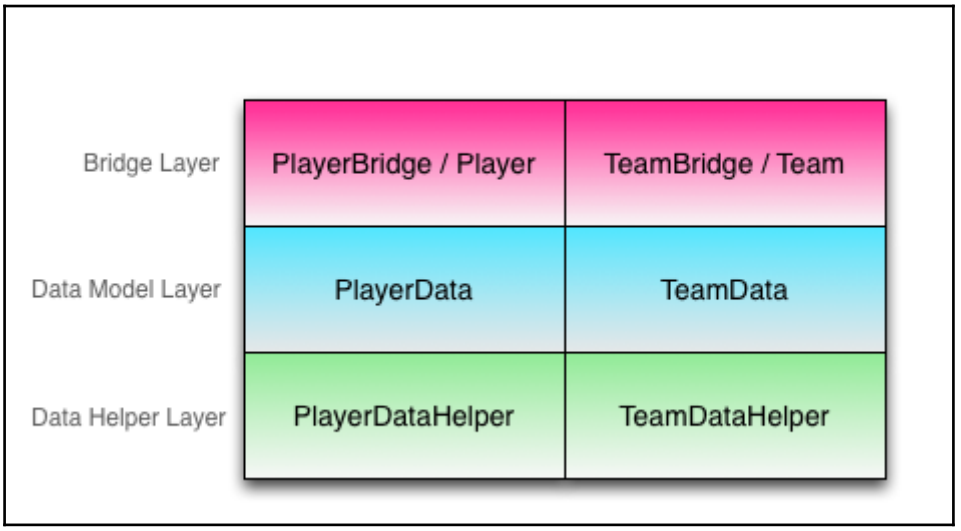


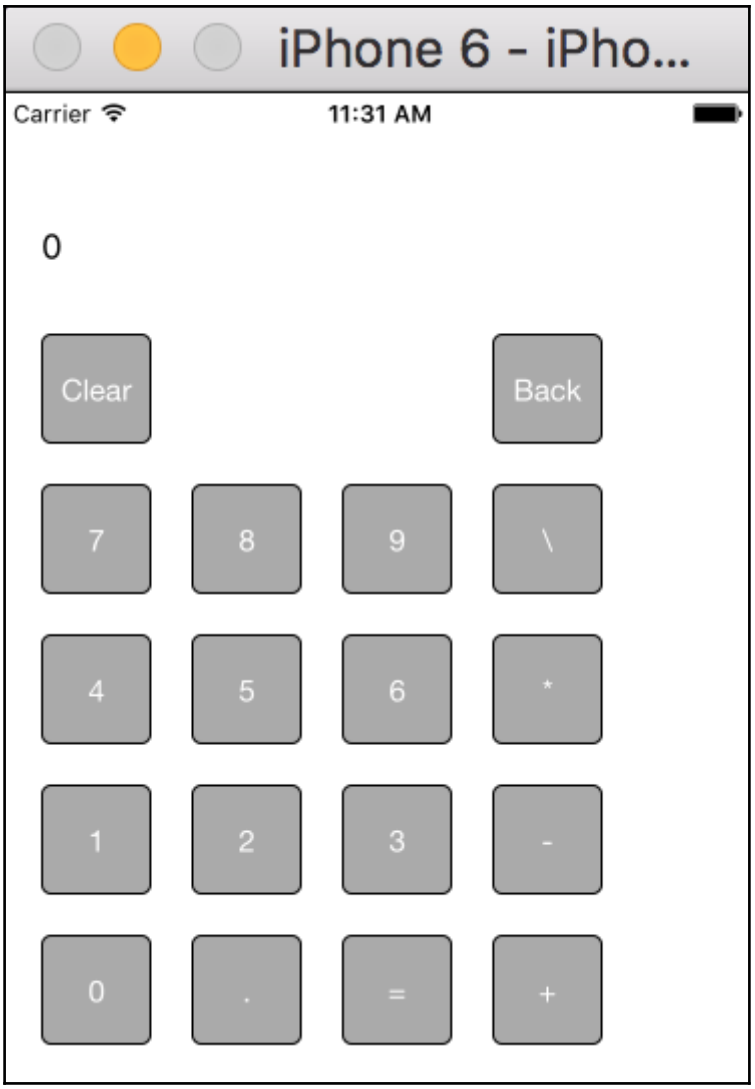
Chapter 6:Protocol-Oriented Programming



Chapter 8: Case Studies






iPhone 6 - iPho...
Carrier
11:31 AM
0
Clear
Back
7
8
9
\
4
5
6
*
1
2
3
-
0
.
=
+