



# Swift 2面向对象编程 (影印版)

Object-Oriented Programming with Swift 2

Gastón C. Hillar 著

[PACKT]  
PUBLISHING



东南大学出版社  
SOUTHEAST UNIVERSITY PRESS

# Swift 2 面向对象编程(影印版)

*Gastón C. Hillar* 著

南京 东南大学出版社

## 图书在版编目(CIP)数据

Swift 2 面向对象编程:英文/(意)加斯顿·C.希勒  
著. —影印本. —南京:东南大学出版社, 2017.4

书名原文: Object-Oriented Programming with Swift 2

ISBN 978-7-5641-7079-0

I. ①S… II. ①加… III. ①程序语言—程序设计—英文 IV. ①TP312

中国版本图书馆 CIP 数据核字(2017)第 051872 号

© 2016 by PACKT Publishing Ltd

Reprint of the English Edition, jointly published by PACKT Publishing Ltd and Southeast University Press, 2017.  
Authorized reprint of the original English edition, 2016 PACKT Publishing Ltd, the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 PACKT Publishing Ltd 出版 2016。

英文影印版由东南大学出版社出版 2017。此影印版的出版和销售得到出版权和销售权的所有者——PACKT Publishing Ltd 的许可。

版权所有,未得书面许可,本书的任何部分和全部不得以任何形式复制。

## Swift 2 面向对象编程(影印版)

---

出版发行:东南大学出版社

地 址:南京四牌楼 2 号 邮编:210096

出 版 人:江建中

网 址: <http://www.seupress.com>

电子邮件: [press@seupress.com](mailto:press@seupress.com)

印 刷:常州市武进第三印刷有限公司

开 本:787 毫米×980 毫米 16 开本

印 张:20.5

字 数:401 千字

版 次:2017 年 4 月第 1 版

印 次:2017 年 4 月第 1 次印刷

书 号:ISBN 978-7-5641-7079-0

定 价:74.00 元

---

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

# Credits

**Author**

Gastón C. Hillar

**Project Coordinator**

Nikhil Nair

**Reviewers**

Vinod Madigeri

Hugo Solis

**Proofreader**

Safis Editing

**Commissioning Editor**

Amarabha Banerjee

**Indexer**

Monica Ajmera Mehta

**Acquisition Editors**

Nadeem Bagban

Reshma Raman

**Graphics**

Disha Haria

**Content Development Editor**

Divij Kotian

**Production Coordinator**

Nilesh Mohite

**Technical Editor**

Parag Topre

**Cover Work**

Nilesh Mohite

**Copy Editor**

Shruti Iyer

# About the Author

**Gastón C. Hillar** is an Italian and has been working with computers since he was 8 years old. In the early 80s, he began programming with the legendary Texas TI-99/4A and Commodore 64 home computers. Gaston has a bachelor's degree in computer science and graduated with honors. He also holds an MBA in which he graduated with an outstanding thesis. At present, Gaston is an independent IT consultant and a freelance author who is always looking for new adventures around the world.

He has been a senior contributing editor at Dr. Dobb's and has written more than a hundred articles on software development topics. Gaston was also a former Microsoft MVP in technical computing. He has received the prestigious Intel® Black Belt Software Developer award seven times.

He is a guest blogger at Intel® Software Network (<http://software.intel.com>). You can reach him at [gastonhillar@hotmail.com](mailto:gastonhillar@hotmail.com) and follow him on Twitter at <http://twitter.com/gastonhillar>. Gastón's blog is <http://csharpmulticore.blogspot.com>.

He lives with his wife, Vanesa, and his two sons, Kevin and Brandon.

# About the Reviewers

**Vinod Madigeri** is a curious developer with a particular interest in object-oriented programming. He has worked in different industries (telecommunication, game technologies, and consumer electronics) writing software in C, C++, Objective-C, Swift, and C#.

Vinod is a passionate software engineer who writes code for fun. He has been doing this professionally for some 6 years and had been goofing with computers for 10 years before that.

**Hugo Solis** is an assistant professor in the physics department at University of Costa Rica. His current research interests are computational cosmology, complexity, and the influence of hydrogen on material properties. Hugo has wide experience with languages such as C/C++ and Python for scientific programming and visualization. He is a member of Free Software Foundation and has contributed code to a few free software projects. Hugo has also been a technical reviewer for *Mastering Object-Oriented Python*, *Learning Object-Oriented Programming* and *Kivy: Interactive Applications in Python* and the author of *Kivy Cookbook*, Packt Publishing. Currently, he is in charge of IFT, a Costa Rican scientific nonprofit organization for the multidisciplinary practice of physics (<http://iftucr.org>).

---

I'd like to thank my beloved mother, Katty Sanchez, for her support and vanguard thoughts.

---

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.com).

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

*To my sons, Kevin and Brandon, and my wife, Vanesa*



# Preface

Object-oriented programming, also known as OOP, is a required skill in any modern software developer job. It makes a lot of sense because object-oriented programming allows you to maximize code reuse and minimize maintenance costs. However, learning object-oriented programming is challenging because it includes too many abstract concepts that require real-life examples to be easy to understand. In addition, object-oriented code that doesn't follow best practices can easily become a maintenance nightmare.

Swift is a multi-paradigm programming language, and one of its most important paradigms is OOP. If you want to create great applications and apps for Mac, iPhone, iPad, Apple TV, and Apple Watch, you need to master OOP in Swift. In addition, as Swift also grabs the nice features found in functional programming languages, it is convenient to know how to mix OOP code with functional programming code.

This book will allow you to develop high-quality reusable object-oriented code in Swift 2.2. You will learn the object-oriented programming principles and how Swift implements them. You will learn how to capture objects from real-world elements and create object-oriented code that represents them. You will understand Swift's approach towards object-oriented code. You will maximize code reuse and reduce maintenance costs. Your code will be easy to understand, and it will work with representations of real-life elements.

## What this book covers

*Chapter 1, Objects from the Real World to Playground*, teaches you the principles of object-oriented paradigms. We will discuss how real-world objects can become part of the fundamental elements of code. We will translate elements into the different components of the object-oriented paradigm supported in Swift: classes, protocols, properties, methods, and instances.

*Chapter 2, Structures, Classes, and Instances*, starts generating blueprints to create objects. You will learn about an object's life cycle, and we will work with many examples to understand how object initializers and deinitializers work.

*Chapter 3, Encapsulation of Data with Properties*, introduces you to organizing data in the blueprints that generate objects. We will understand the different members of a class and how they are reflected by members of the instances generated from a class. You will learn the difference between mutable and immutable classes.

*Chapter 4, Inheritance, Abstraction, and Specialization*, introduces you to creating a hierarchy of blueprints that generate objects. We will take advantage of inheritance and many related features to specialize behavior.

*Chapter 5, Contract Programming with Protocols*, discusses how Swift works with protocols in combination with classes. We will declare and combine multiple blueprints to generate a single instance. We will declare protocols with different types of requirements, and then we will create classes that conform to these protocols.

*Chapter 6, Maximization of Code Reuse with Generic Code*, teaches you how to maximize code reuse by writing code capable of working with objects of different types—that is, instances of classes that conform to specific protocols or whose class hierarchy includes specific superclasses. We will work with protocols and generics.

*Chapter 7, Object-Oriented Programming and Functional Programming*, teaches you how to refactor existing code to take full advantage of object-oriented code. We will prepare the code for future requirements, reduce maintenance costs, and maximize code reuse. We will also work with many functional programming features included in Swift combined with object-oriented programming.

*Chapter 8, Extending and Building Object-Oriented Code*, puts together all the pieces of the object-oriented puzzle. We will take advantage of extensions to add features to types, classes, and protocols in which we don't have access to the source code. We will make sure that the code exposes only the things that it has to expose, and you will learn how everything you learned about object-oriented programming is useful in any kind of app we might create.

## What you need for this book

In order to work with Xcode and the Swift Playground, you will need a Mac OS computer capable of running OS X 10.10.5 or later with 8 GB of RAM.

In order to work with the Swift open source version on the Linux platform, you will need any computer capable of running Ubuntu 14.04 or later or Ubuntu 15.10 or later. These are the Linux distributions where the Swift open source binaries are built and tested. It is also possible to run the Swift compiler and utilities on other Linux distributions. You must take a look at the latest available documentation at the Swift open source website, <https://swift.org>.

## Who this book is for

If you are an iOS developer who has a basic idea of OOP and want to incorporate OOP concepts with Swift to optimize your application's performance, then this book is for you. This is a very useful resource for developers who want to shift from Objective C, C#, Java, Python, JavaScript, or other object-oriented languages to Swift.

## Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "We can assign 20 to `regularHexagon1.lengthOfSide` and 50 to `regularHexagon2.lengthOfSide`."

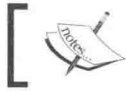
A block of code is set as follows:

```
let degCUnitFromStr = HKUnit(fromString: "degC")
let degFUnitFromStr = HKUnit(fromString: "degF")
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
Animal created
Mammal created
DomesticMammal created
Dog created
TerrierDog created
SmoothFoxTerrier created
I am 7 years old.
I am 14 years old.
I am 21 years old.
I am 4 years old.
I am 5 years old.
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Start Xcode, navigate to **File** | **New** | **Playground...**"



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

## Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## eBooks, discount offers, and more

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [customercare@packtpub.com](mailto:customercare@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

## Questions

If you have a problem with any aspect of this book, you can contact us at [questions@packtpub.com](mailto:questions@packtpub.com), and we will do our best to address the problem.

# Table of Contents

<b>Preface</b>	<b>v</b>
<b>Chapter 1: Objects from the Real World to Playground</b>	<b>1</b>
Installing the required software	1
Capturing objects from the real world	4
Generating classes to create objects	11
Recognizing variables and constants to create properties	14
Recognizing actions to create methods	17
Organizing classes with UML diagrams	20
Working with API objects in the Xcode Playground	26
Exercises	31
Test your knowledge	31
Summary	32
<b>Chapter 2: Structures, Classes, and Instances</b>	<b>33</b>
Understanding structures, classes, and instances	33
Understanding initialization and its customization	34
Understanding deinitialization and its customization	36
Understanding automatic reference counting	36
Declaring classes	37
Customizing initialization	38
Customizing deinitialization	41
Creating the instances of classes	45
Exercises	46
Test your knowledge	46
Summary	47
<b>Chapter 3: Encapsulation of Data with Properties</b>	<b>49</b>
Understanding the elements that compose a class	49
Declaring stored properties	51
Generating computed properties with setters and getters	54

Combining setters, getters, and a related property	62
Understanding property observers	65
Transforming values with setters and getters	69
Using type properties to create values shared by all the instances of a class	70
Creating mutable classes	74
Building immutable classes	78
Exercises	81
Test your knowledge	81
Summary	82
<b>Chapter 4: Inheritance, Abstraction, and Specialization</b>	<b>83</b>
Creating class hierarchies to abstract and specialize behavior	83
Understanding inheritance	88
Declaring classes that inherit from another class	90
Overriding and overloading methods	96
Overriding properties	101
Controlling whether subclasses can or cannot override members	103
Working with typecasting and polymorphism	108
Taking advantage of operator overloading	121
Declaring operator functions for specific subclasses	126
Exercises	128
Test your knowledge	128
Summary	129
<b>Chapter 5: Contract Programming with Protocols</b>	<b>131</b>
Understanding how protocols work in combination with classes	131
Declaring protocols	133
Declaring classes that adopt protocols	137
Taking advantage of the multiple inheritance of protocols	142
Combining inheritance and protocols	144
Working with methods that receive protocols as arguments	152
Downcasting with protocols and classes	155
Treating instances of a protocol type as a different subclass	159
Specifying requirements for properties	162
Specifying requirements for methods	164
Combining class inheritance with protocol inheritance	166
Exercises	178
Test your knowledge	179
Summary	180

<b>Chapter 6: Maximization of Code Reuse with Generic Code</b>	<b>181</b>
Understanding parametric polymorphism and generic code	181
Declaring a protocol to be used as a constraint	183
Declaring a class that conforms to multiple protocols	184
Declaring subclasses that inherit the conformance to protocols	188
Declaring a class that works with a constrained generic type	190
Using a generic class for multiple types	195
Combining initializer requirements in protocols with generic types	201
Declaring associated types in protocols	202
Creating shortcuts with subscripts	204
Declaring a class that works with two constrained generic types	206
Using a generic class with two generic type parameters	209
Inheriting and adding associated types in protocols	213
Generalizing existing classes with generics	214
Extending base types to conform to custom protocols	223
Test your knowledge	225
Exercises	226
Summary	227
<b>Chapter 7: Object-Oriented Programming and Functional Programming</b>	<b>229</b>
Refactoring code to take advantage of object-oriented programming	229
Understanding functions as first-class citizens	241
Working with function types within classes	243
Creating a functional version of array filtering	245
Writing equivalent closures with simplified code	247
Creating a data repository with generics and protocols	248
Filtering arrays with complex conditions	253
Using map to transform values	256
Combining map with reduce	259
Chaining filter, map, and reduce	262
Solving algorithms with reduce	262
Exercises	264
Test your knowledge	265
Summary	266
<b>Chapter 8: Extending and Building Object-Oriented Code</b>	<b>267</b>
Putting together all the pieces of the object-oriented puzzle	267
Adding methods with extensions	269
Adding computed properties to a base type with extensions	273
Declaring new convenience initializers with extensions	278
Defining subscripts with extensions	280



<b>Working with object-oriented code in apps</b>	<b>281</b>
<b>Adding an object-oriented data repository to a project</b>	<b>290</b>
<b>Interacting with an object-oriented data repository through</b>	
<b>Picker View</b>	<b>294</b>
<b>Exercises</b>	<b>299</b>
<b>Test your knowledge</b>	<b>299</b>
<b>Summary</b>	<b>300</b>
<b>Appendix: Exercise Answers</b>	<b>301</b>
<b>Chapter 1, Objects from the Real World to Playground</b>	<b>301</b>
<b>Chapter 2, Structures, Classes, and Instances</b>	<b>301</b>
<b>Chapter 3, Encapsulation of Data with Properties</b>	<b>302</b>
<b>Chapter 4, Inheritance, Abstraction, and Specialization</b>	<b>302</b>
<b>Chapter 5, Contract Programming with Protocols</b>	<b>302</b>
<b>Chapter 6, Maximization of Code Reuse with Generic Code</b>	<b>302</b>
<b>Chapter 7, Object-Oriented Programming and Functional</b>	
<b>Programming</b>	<b>303</b>
<b>Chapter 8, Extending and Building Object-Oriented Code</b>	<b>303</b>
<b>Index</b>	<b>305</b>

---