

# Apple App-Development-with-Swift-Certified-User復習 内容 & App-Development-with-Swift-Certified-User全真 模擬試験



## APP DEVELOPMENT WITH SWIFT Certified User

ITエリートになるという夢は現実の世界で叶えやすくありません。しかし、AppleのApp-Development-with-Swift-Certified-User認定試験に合格するという夢は、PassTestに対して、絶対に掴められます。PassTestは親切なサービスで、AppleのApp-Development-with-Swift-Certified-User問題集が質の良くて、AppleのApp-Development-with-Swift-Certified-User認定試験に合格する率も100パーセントになっています。PassTestを選ぶなら、私たちは君の認定試験に合格するのを保証します。

PassTestは、魅力的なキャラクターで世界中の試験受験者を招きます。当社の専門家は彼らの卓越性に大きく貢献しました。したがって、試験をシミュレートするApp-Development-with-Swift-Certified-Userが最良であると率直に言うことができます。App-Development-with-Swift-Certified-User学習教材のコンテンツを作成する取り組みは、学習ガイドの開発につながり、完成度を高めます。関心を集め、いくつかの難しい点を簡素化するために、当社の専門家は当社のApp-Development-with-Swift-Certified-User学習資料を設計し、App-Development-with-Swift-Certified-User学習ガイドをよりよく理解できるように最善を尽くします。

>> Apple App-Development-with-Swift-Certified-User復習内容 <<

## App-Development-with-Swift-Certified-User全真模擬試験 & App- Development-with-Swift-Certified-Userテストサンプル問題

タスクを効率的に完了できない場合は、App-Development-with-Swift-Certified-User学習教材の使用をお勧めします。特定の状況を評価することにより、合理的なスケジュールを提供し、App-Development-with-Swift-Certified-User試験トレーニングガイドの拡張可能なバージョンを提供し、短時間でより多くの知識をすばやく把握できます。同時に、あなたはあなたの周りの人々以上のことをします。これがApp-Development-with-Swift-Certified-Userテストガイドでできることです。App-Development-with-Swift-Certified-User学習ガイドは、効率を向上させ、より高い品質でタスクを完了するためのものです。

## Apple App Development with Swift Certified User Exam 認定 App-Development-with-Swift-Certified-User 試験問題 (Q32-Q37):

### 質問 # 32

Which two statements about building an app are true? (Choose 2.)

- A. Your phone must always be physically attached to your Mac to run your apps from Xcode on it.
- **B. You can preview a View in the Canvas without running your app.**
- C. You need a paid Apple Developer account in order to run your app on your phone.
- D. You can run your app in the simulator with Generic iOS Device chosen.
- **E. You can run an app on your phone and get debug information in Xcode.**

正解: B、E

解説:

Comprehensive and Detailed Explanation From App Development with Swift domains:

This question belongs to Xcode Developer Tools , especially the objectives about using the Xcode interface, building and running an app, and debugging. A is true because Xcode supports SwiftUI previews in the canvas, allowing you to see a view's interface directly in Xcode without fully launching the entire app in the normal run workflow. Apple's documentation states that Xcode can display a preview of a custom SwiftUI view in the preview canvas and keep it updated as you make code changes.

D is also true because when you run an app from Xcode on a device, Xcode opens a debugging session in the debug area. Apple explicitly documents that after a successful build, Xcode runs the app and opens a debugging session, which means you can view debug information while the app is running on the phone.

The other options are false. B is false because a phone does not have to be physically attached at all times; modern Xcode workflows support device pairing and wireless development after setup. C is false because Generic iOS Device is not an actual simulator run target for launching the app like a specific simulator device. E is false because you do not need a paid Apple Developer Program membership merely to run an app on your own device for development; Apple provides support for development testing on devices with the required setup such as pairing and Developer Mode.

### 質問 # 33

Review the code snippet.

□ The code snippet does not compile.

Which two actions will fix the errors? (Choose 2.)

- A. Change shipping from let to var to make it mutable.
- B. Change the initial value of totalCost from 0 to 0.0.
- **C. Change the type of quantity from int to Double .**
- D. Change the type of unitPrice from Double to Int.
- **E. Change totalCost from let to var to make it mutable.**

正解: C、E

解説:

This question belongs to Swift Programming Language , especially the domains covering basic Swift types , operators , and constants versus variables .

There are two compile problems in the snippet. First, unitPrice and shipping are inferred as Double, while quantity is inferred as Int. In Swift, arithmetic operands must have compatible types; Swift does not automatically mix Int and Double in one arithmetic expression. So unitPrice \* quantity fails unless quantity is changed to Double or explicitly converted. That makes A a correct fix.

Second, the line totalCost += ... uses the compound assignment operator +=, which stores a new value back into the left-hand side. Swift requires the left-hand side of += to be mutable, so totalCost must be declared with var, not let. That makes D the second correct fix.

The other choices do not solve the actual compile issues. B is unnecessary because totalCost is already explicitly declared as Double, so 0 is valid there. C would still leave shipping as Double, so the mixed-type arithmetic problem remains. E is irrelevant because shipping is never reassigned. Therefore, the two correct answers are A and D

### 質問 # 34

You have created a view which includes some formatted text:

□ You decide to extract this formatted text into a subview so that you can reuse the formatting for other texts.

You highlight the Text and choose Extract subview.  
You refactor ExtractedView to BigGreenTextView.  
You add the line let text: String to the extracted view above the body.  
You replace "That ' s all folks " with a reference to text  
How should you call this extracted View from the original View?

- A. BigGreenTextView(text)
- **B. BigGreenTextView(text: " That ' s all folks ")**
- C. BigGreenTextView(text: text)
- D. BigGreenTextView( " That ' s all folks ")

**正解: B**

解説:

This question belongs to View Building with SwiftUI , specifically the domain on extracting subviews to simplify the structure of an overlarge View .

When you create a custom SwiftUI view and add a stored property like:

let text: String

that property becomes part of the view's initializer. Since the property name is text, Swift expects you to pass the value using the parameter label text: when creating the view. So the correct call is:

BigGreenTextView(text: " That ' s all folks ")

That makes B the correct answer.

Why the others are wrong:

\* A is incorrect because this is not using the expected parameter label.

\* C would only be correct if there were already another variable named text in the calling scope and you wanted to pass that variable instead of the literal string shown in the question.

\* D is incorrect because it omits the parameter label.

This is a standard SwiftUI pattern: extract reusable formatting into a separate custom View, give it an input property, and then pass the needed value through the initializer using the property label.

**質問 # 35**

Select the location to set this app to run on an iPhone 14 in the simulator.

□

**正解:**

解説:

□

Explanation:

□

This question belongs to Xcode Developer Tools , specifically the domain for building and running an app on the iOS simulator . In Xcode, the place where you choose whether the app runs on a simulator or a connected device is the run destination / scheme destination selector in the toolbar. This control appears near the top center of the Xcode window and displays the current destination, such as a simulator model or device target. To run the app on iPhone 14 , you click that selector and choose iPhone 14 from the available simulator list. Apple's Xcode documentation describes choosing a run destination before building and running the app in Simulator or on a device.

So, for the hotspot, the correct place is the device/simulator dropdown in the top toolbar , not the preview canvas controls, not the project navigator, and not the code editor. That selector determines where the app launches when you press Run.

**質問 # 36**

Complete the code that conforms to the View protocol by selecting the correct option from each drop-down list.

Note: You will receive partial credit for each correct answer.

□

**正解:**

解説:

□

Explanation:

□

This question belongs to View Building with SwiftUI , especially the domain covering positioning and/or layout a single SwiftUI View with standard Views and modifiers and the foundational structure of a SwiftUI view. In SwiftUI, a custom screen is typically declared as a struct that conforms to the View protocol. Apple's SwiftUI documentation shows the standard pattern:

struct ScreenView: View {

```
var body: some View {
  Text( " Hello ")
}
}
```

Here, struct is required because SwiftUI views are commonly defined as structures. View is required after the colon because the type must conform to the View protocol. body is the required computed property that returns the content of the view as some View. Apple documents that every conforming View type must provide a body property that describes its content.

So the completed code is:

```
import SwiftUI
struct ScreenView: View {
  var body: some View {
    Text( " Hello ")
  }
}
```

This is the canonical SwiftUI view declaration pattern and is one of the most fundamental concepts in App Development with Swift.

## 質問 # 37

.....

PassTestはApple試験問題集を提供するウェブサイトで、ここによく分かります。最もよくて最新で資料を提供いたします。こうして、君は安心してApp-Development-with-Swift-Certified-User試験の準備を行ってください。弊社の資料を使って、100%に合格を保証いたします。

**App-Development-with-Swift-Certified-User全真模擬試験**: <https://www.passtest.jp/Apple/App-Development-with-Swift-Certified-User-shiken.html>

App-Development-with-Swift-Certified-Userクイズ準備を使用する場合は、デモについて学ぶ機会があります、Apple App-Development-with-Swift-Certified-User復習内容 それに、あなたに極大な便利と快適をもたらせます、App-Development-with-Swift-Certified-User認定試験が大変難しいと感じて、多くの時間を取らなければならないとしたら、ツールとしてPassTestのApp-Development-with-Swift-Certified-User問題集を利用したほうがいいです、試験の準備をするためにPassTestのAppleのApp-Development-with-Swift-Certified-User試験トレーニング資料を買うのは冒険的行為と思ったとしたら、あなたの人生の全てが冒険なことになります、App-Development-with-Swift-Certified-User試験準備のPCバージョンは、Windowsユーザー向けです、App-Development-with-Swift-Certified-Userトレーニング資料の詳細と保証、連絡方法、当社製品に関するクライアントの評価、および関連する試験。

もしかして、ヨハン様はカークの・はい、ふ・くすぐったいような感触にお尻を揺らす、App-Development-with-Swift-Certified-Userクイズ準備を使用する場合は、デモについて学ぶ機会があります、それに、あなたに極大な便利と快適をもたらせます、App-Development-with-Swift-Certified-User認定試験が大変難しいと感じて、多くの時間を取らなければならないとしたら、ツールとしてPassTestのApp-Development-with-Swift-Certified-User問題集を利用したほうがいいです。

## App-Development-with-Swift-Certified-User試験の準備方法 | 最新の App-Development-with-Swift-Certified-User復習内容試験 | 素敵な App Development with Swift Certified User Exam全真模擬試験

試験の準備をするためにPassTestのAppleのApp-Development-with-Swift-Certified-User試験トレーニング資料を買うのは冒険的行為と思ったとしたら、あなたの人生の全てが冒険なことになります、App-Development-with-Swift-Certified-User試験準備のPCバージョンは、Windowsユーザー向けです。

- App-Development-with-Swift-Certified-User予想試験 □ App-Development-with-Swift-Certified-User対策学習 □ App-Development-with-Swift-Certified-User練習問題 □ ➡ App-Development-with-Swift-Certified-User □□□の試験問題は ⇒ [www.japancert.com](http://www.japancert.com) ⇐ で無料配信中 App-Development-with-Swift-Certified-User日本語版問題集
- App-Development-with-Swift-Certified-User受験料過去問 □ App-Development-with-Swift-Certified-User予想試験 □ App-Development-with-Swift-Certified-User受験記 □ ➡ [www.goshiken.com](http://www.goshiken.com) □ から簡単に ➡ App-Development-with-Swift-Certified-User □ を無料でダウンロードできます App-Development-with-Swift-Certified-User全真問題集
- ユニークな App-Development-with-Swift-Certified-User復習内容試験-試験の準備方法-便利な App-Development-with-Swift-Certified-User全真模擬試験 □ ✨ [www.mogixam.com](http://www.mogixam.com) □ ✨ □ サイトにて最新 ➡ App-Development-with-Swift-Certified-User □ 問題集をダウンロード App-Development-with-Swift-Certified-User試験感想

- App-Development-with-Swift-Certified-User全真問題集 □ App-Development-with-Swift-Certified-User日本語版問題集 □ App-Development-with-Swift-Certified-User受験料過去問 □ [ [www.goshiken.com](http://www.goshiken.com) ]を入力して「App-Development-with-Swift-Certified-User」を検索し、無料でダウンロードしてくださいApp-Development-with-Swift-Certified-User的中合格問題集
- App-Development-with-Swift-Certified-User必殺問題集 □ App-Development-with-Swift-Certified-User必殺問題集 □ App-Development-with-Swift-Certified-User練習問題 □ 今すぐ[ [www.passtest.jp](http://www.passtest.jp) ]を開き、▶ App-Development-with-Swift-Certified-User □を検索して無料でダウンロードしてくださいApp-Development-with-Swift-Certified-Userテスト対策書
- App-Development-with-Swift-Certified-User復習教材 □ App-Development-with-Swift-Certified-User資格講座 □ App-Development-with-Swift-Certified-Userテスト対策書 □ 【 [www.goshiken.com](http://www.goshiken.com) 】の無料ダウンロード{ App-Development-with-Swift-Certified-User }ページが開きますApp-Development-with-Swift-Certified-User試験感想
- App-Development-with-Swift-Certified-User復習資料 □ App-Development-with-Swift-Certified-Userテスト対策書 □ App-Development-with-Swift-Certified-User練習問題 □ ▶ App-Development-with-Swift-Certified-User □の試験問題は★ [www.japancert.com](http://www.japancert.com) □★□で無料配信中App-Development-with-Swift-Certified-User必殺問題集
- App-Development-with-Swift-Certified-User資格講座 □ App-Development-with-Swift-Certified-User復習教材 □ App-Development-with-Swift-Certified-User受験料過去問 □★ [www.goshiken.com](http://www.goshiken.com) □★□サイトにて《 App-Development-with-Swift-Certified-User 》問題集を無料で使おうApp-Development-with-Swift-Certified-User難易度
- 100%合格App-Development-with-Swift-Certified-User復習内容と完璧なApp-Development-with-Swift-Certified-User全真模擬試験 □ 今すぐ★ [www.it-passports.com](http://www.it-passports.com) □★□で“App-Development-with-Swift-Certified-User”を検索し、無料でダウンロードしてくださいApp-Development-with-Swift-Certified-User試験感想
- 試験の準備方法-有効的なApp-Development-with-Swift-Certified-User復習内容試験-最新のApp-Development-with-Swift-Certified-User全真模擬試験 □ ( [www.goshiken.com](http://www.goshiken.com) )に移動し、( App-Development-with-Swift-Certified-User )を検索して、無料でダウンロード可能な試験資料を探しますApp-Development-with-Swift-Certified-User合格体験記
- 試験App-Development-with-Swift-Certified-User復習内容 - 一生懸命にApp-Development-with-Swift-Certified-User全真模擬試験 | 最高のApp-Development-with-Swift-Certified-Userテストサンプル問題 □ サイト▶ [www.xhs1991.com](http://www.xhs1991.com) □で□ App-Development-with-Swift-Certified-User □問題集をダウンロードApp-Development-with-Swift-Certified-User最新な問題集
- [www.competize.com](http://www.competize.com), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [albsaer.alalawidesigner.com](http://albsaer.alalawidesigner.com), [www.notebook.ai](http://www.notebook.ai), [connect.garmin.com](http://connect.garmin.com), [p.mepage.com](http://p.mepage.com), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [www.stes.tyc.edu.tw](http://www.stes.tyc.edu.tw), [learn.csisafety.com.au](http://learn.csisafety.com.au), [unilisto.com](http://unilisto.com), Disposable vapes