

App-Development-with-Swift-Certified-User: App Development with Swift Certified User Exam Dumps & PassGuide App-Development-with-Swift-Certified-User Examen



APP DEVELOPMENT WITH SWIFT Certified User

Die Apple App-Development-with-Swift-Certified-User Zertifizierungsprüfung ist eine der wertvollsten zeitgenössischen Zertifizierungsprüfung. In den letzten Jahrzehnten ist die Computer-Ausbildung schon ein Fokus geworden. Sie ist ein notwendiger Bestandteil der Informations-Technologie im IT-Bereich. So legen viele IT-fachleute diese Prüfung ab, um ihr Wissen zu erweitern und einen Durchbruch in allen Bereichen zu verschaffen. Und unsere Fragen und Antworten zur Apple App-Development-with-Swift-Certified-User Zertifizierungsprüfung sind genau das, was sie brauchen. Dennoch ist es schwer, diesen Test zu bestehen. Wählen Sie die entsprechende Abkürzung, um Erfolg zu garantieren. Wählen Sie ZertSoft, kommt der Erfolg auf Sie zu. Die Fragen und Antworten zur Apple App-Development-with-Swift-Certified-User Zertifizierung von ZertSoft werden von den IT-Eliten nach ihren Erfahrungen und Praxien bearbeitet und haben die Zertifizierungserfahrung von mehr als zehn Jahren.

Wenn Sie unsere Lernmaterialien zur Apple App-Development-with-Swift-Certified-User Zertifizierungsprüfung benutzen wollen, werden sicher die Zeit und Wirtschaftskosten reduziert. Vorm Kauf unserer Apple App-Development-with-Swift-Certified-User Prüfungsfrage können Sie kostenlos unsere Fragen herunterladen. Sie sind in der Form von PDF und Software. Wenn Sie die Softwareversion brauchen, bitte setzen Sie sich in Verbindung mit unserem Kundenservice.

>> **App-Development-with-Swift-Certified-User German** <<

bestehen Sie App-Development-with-Swift-Certified-User Ihre Prüfung mit unserem Prep App-Development-with-Swift-Certified-User Ausbildung Material & kostenloser Dowload Torrent

Wenn Sie die richtige Methode benutzen, haben Sie schon halben Erfolg erhalten. Wir ZertSoft bieten Ihnen die effizienteste Methode für Apple App-Development-with-Swift-Certified-User Prüfung, die von unseren erfahrenen Forschungs-und Entwicklungsstellen hergestellt wird. Auf unserer offiziellen Webseite können Sie durch Paypal die Apple App-Development-with-Swift-Certified-User Prüfungsunterlagen gesichert kaufen. Wir werden Ihre Persönliche Informationen und Zahlungsinformationen gut bewahren und bieten Ihnen nach dem Kauf der Apple App-Development-with-Swift-Certified-User Unterlagen immer weiter

hochwertigen Dienst.

Apple App Development with Swift Certified User Exam App-Development-with-Swift-Certified-User Prüfungsfragen mit Lösungen (Q28-Q33):

28. Frage

Review the code snippet.

The " faces " dictionary contains emojis and their descriptions.

Which code will create an array named " emojis " that will copy all the emojis from the " faces " dictionary?

- A. let emojis List(faces.values())
- B. let emojis = Array(faces.values)
- C. let emojis = List(faces.keys())
- D. let emojis = Array(faces.keys)

Antwort: D

Begründung:

This question belongs to Swift Programming Language , specifically the objective on managing data using collection types , including dictionaries and arrays . In the dictionary shown, the emojis are the keys and the text descriptions are the values . Swift provides a keys property on dictionaries that returns a collection containing all dictionary keys. To convert that keys collection into an array, you use the Array(...) initializer.

Therefore, the correct code is let emojis = Array(faces.keys). Apple documents both the Dictionary.keys property and the Array type used to store a sequence of values of the same type.

Option B is incorrect because faces.values would return the descriptions like " grinning " , " thinking " , and " happy " , not the emoji keys. Options A and C are incorrect because List is a SwiftUI view type, not the correct collection type for creating an array from dictionary contents. Also, the dictionary interface uses properties like .keys and .values, not method calls like .keys() or .values().

Apple's dictionary documentation makes clear that keys is a property returning a collection of the dictionary's keys.

29. Frage

Refer to this image to complete the code.

Note: You will receive partial credit for each correct answer

Antwort:

Begründung:

Explanation:

This question belongs to View Building with SwiftUI , especially the objectives for using List views to iterate through collections and structuring views with standard SwiftUI containers. The screenshot shows two grouped sets of rows: one headed MY FRIENDS and one headed MY PETS . In SwiftUI, the correct container for a scrollable table-style presentation of rows is List, and the correct way to divide that list into labeled groups is Section. Apple documents List as a container that presents data in a single-column row- based layout, and Section as a way to organize list content into grouped areas with headers and optional footers. That is exactly the structure shown in the image. (developer.apple.com , developer.apple.com) The ForEach(names, id: \.self) and ForEach(pets, id: \.self) lines are already iterating through the arrays, so each ForEach should be wrapped inside a Section. The section labels such as " My Friends " and " My Pets "

" are provided with the header: label. So the intended code structure is:

```
List {
  Section {
    ForEach(names, id: \.self) { name in Text(name) }
  } header: {
    Text( " My Friends " )
  }
  Section {
    ForEach(pets, id: \.self) { pet in Text(pet) }
  } header: {
    Text( " My Pets " )
  }
}
```

This matches the UI shown in the image and aligns directly with SwiftUI list and section composition patterns in App Development

with Swift.

30. Frage

Review the code snippet.

Move each item from the list on the left to the correct code segment on the right. You may use each item only once.

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

Antwort:

Begründung:

Explanation:

This question belongs to Swift Programming Language , specifically the domain covering structs, properties, methods, and initializers

A computed property does not store a value directly. Instead, it returns a value calculated from other data.

That is why description is a computed property: it returns a string based on content.

A memberwise initializer is automatically provided by Swift for structs when their stored properties are initialized through parameters. So Document(content: "Greetings! ") is using the struct's memberwise initializer.

A type property belongs to the type itself rather than to an instance. In Swift, static var docCount = 0 is a type property because it is declared with static.

An instance method is a function that belongs to an instance of the struct or class. The display() method uses the instance's content, so it is an instance method.

A type method is a method declared with static and belongs to the type itself. So static func increment() is a type method because it changes the shared type property docCount.

31. Frage

Given the function definition, which two statements call the function correctly? (Choose 2.)

Based on the image provided, here is the text for each of the multiple-choice options:

- A. schedule(who: " Jane Doe ", from: " 9:30am ", to: " 10:30am ", " Office ")
- B. D. schedule(name: " Jane Doe ", starting: " 9:30am ", ending: " 10:30am ", place: " Office ")
- C. E. schedule(who: " Jane Doe ", from: " 9:30am ", to: " 10:30am ")
- D. schedule(who name: " Jane Doe ", from starting: " 9:30am ", to ending: " 10:30am ")
- E. schedule(who: " Jane Doe ", from: " 9:30am ", to: " 10:30am ", place: " Office ")

Antwort: C,D

Begründung:

This question belongs to Swift Programming Language , specifically the objective on functions , including internal and external parameter names and default parameter values .

The function is defined as:

```
func schedule(who name: String, from starting: String, to ending: String, _ place: String = "Zoom") { print( " Appointment: meeting \  
(name) from \  
(starting) to \  
(ending) at \  
(place) " )  
}
```

This means:

- * the external parameter names are who, from, and to
- * the internal parameter names are name, starting, and ending
- * the last parameter uses _, which means it has no external label
- * the last parameter also has a default value of " Zoom "

Now evaluate the options:

* A is incorrect because it uses place: as an external label, but _ place means no external label is allowed.

* B is correct because it uses the required external names who, from, and to, and it omits the last parameter, which is allowed because it has a default value.

* C is incorrect because it uses who:, from:, and to: correctly, but this function's first three parameters are not declared that way in the provided option set; the valid matching call style from the choices is not this one because the function's labels are paired with internal names in the declaration syntax shown in the question.

* D is incorrect because it uses the internal names name, starting, and ending as if they were external labels.

* E is correct because it uses the external labels who, from, and to, and omits the final unlabeled parameter, letting Swift use the

default "Zoom".

So the two correct answers are B and E.

32. Frage

Review the code snippet.

□ What value does the code output?

Antwort:

Begründung:

Answer the question by typing in the box.

2

Explanation:

This question belongs to Swift Programming Language, specifically the objectives covering functions, control flow, and default parameter values. The function is declared as `func getCategory(_ age: Int = 20) -> Int`, which means if no argument is supplied, Swift uses the default value 20. Apple's Swift documentation explains that you can define a default value for any parameter, and that value is used when the caller omits that argument. Since the code calls `getCategory()` with no parameter, the function executes using `age = 20`.

The conditional logic is then evaluated in order:

* `if age > 64` # false, because 20 is not greater than 64

* `else if age > 19` # true, because 20 is greater than 19

* so the function returns 2

Because Swift's `if/else if` control flow stops at the first true condition, the later checks are never reached once `age > 19` succeeds. Apple describes Swift as supporting standard control flow including conditional branching, and this example is a direct use of that branching behavior.

Therefore, `print(getCategory())` outputs 2, which corresponds to option B.

33. Frage

.....

Nur kontinuierlich zu verbessern kann man immer an der führenden Stelle stehen. Und es ist auch unsere Firmenphilosophie. Deshalb prüfen wir regelmäßig nach, ob die Apple App-Development-with-Swift-Certified-User Prüfung aktualisiert hat. Wenn sie aktualisiert hat, informieren wir unsere Kunden sofort darüber. Dadurch lassen Sie die neueste Informationen über Apple App-Development-with-Swift-Certified-User Prüfung erfahren. Aller Kundendienst der Aktualisierung nach der Kauf der Apple App-Development-with-Swift-Certified-User Software ist kostenlos innerhalb einem Jahr.

App-Development-with-Swift-Certified-User Online Prüfungen: <https://www.zertsoft.com/App-Development-with-Swift-Certified-User-pruefungsfragen.html>

Apple App-Development-with-Swift-Certified-User German Das zeigt die Fähigkeiten unseres Expertenteams, Apple App-Development-with-Swift-Certified-User German Mit unseren Test-Dumps können Sie richtig studieren und werden Sie doppelte Ergebnisse mit der Hälfte der Anstrengung bekommen, Wir ZertSoft beachten immer die vielfältige Veränderungen der Apple App-Development-with-Swift-Certified-User Prüfung, Apple App-Development-with-Swift-Certified-User German Sie ist ein hilfreicher Studienführer, mit dem das Lernen erleichtert wird.

Alle standen ängstlich auf, Binder erhob den App-Development-with-Swift-Certified-User Online Prüfungen Stock, das Tier machte einige Sätze, blieb aber auf einmal stehen, schnupperte, trabte auf Caspar zu, der bleich und stille App-Development-with-Swift-Certified-User Buch saß, wedelte mit dem Schweif und leckte die herabhängende Hand des Jünglings.

Seit Neuem aktualisierte App-Development-with-Swift-Certified-User Examfragen für Apple App-Development-with-Swift-Certified-User Prüfung

Der Verstand findet also in diesem nicht etwa schon eine dergleichen App-Development-with-Swift-Certified-User Verbindung des Mannigfaltigen, sondern bringt sie hervor, indem er ihn affiziert, Das zeigt die Fähigkeiten unseres Expertenteams.

Mit unseren Test-Dumps können Sie richtig studieren und werden Sie doppelte Ergebnisse mit der Hälfte der Anstrengung bekommen, Wir ZertSoft beachten immer die vielfältige Veränderungen der Apple App-Development-with-Swift-Certified-User Prüfung.

