CLASS AND STRUCT

SWIFT PROGRAMMING

@eiwaithein

CLASS And STRUCT

Class and Struct have many common. Both have:

- properties to store values
- methods to provide functionality
- initializer to initialize values
- conform to protocols

Class has more capabilities

- One class can inherit of another

- Type casting can check class instance at runtime.

- Deinitializer can free up any resources.

Class has more capabilities

- Reference counting allow more than one reference to class instance. (*Struct always copied when they passed and do not use reference counting.*)

- Class is Reference type and Struct is value type.

Definition syntax

use **class** keyword for class and use struct for struct.

```
class SampleClass {

}

struct SampleStruct {

}
```

```swift
// 1. Struct and class defination
struct Resolution {
    var width = 10
    var height = 10
}

class VideoMode {
    var resolution = Resolution() // struct
    var interlaced = false
    var frameRate = 0.0
    var name: String?
}

// 2. Struct and class instance
let resolutionStruct = Resolution()
let videoClass = VideoMode()

// 3. Accessing properties.
resolutionStruct.width
videoClass.resolution.width
```

```
{width 10, height 10}
{{width 10, height 10} int...


10
10
```

```swift
// 4. Classes are reference types
let highRes = Resolution(width: 1024, height: 768)
// create an instance for avideoMode
let avideoMode = VideoMode()
avideoMode.resolution = highRes
avideoMode.interlaced = true
avideoMode.frameRate = 60.0
avideoMode.name = "iPad non retina"

// create another instance for avideoMode
var anotherVideoMode = VideoMode()
anotherVideoMode.frameRate = 30.0

anotherVideoMode.frameRate
avideoMode.frameRate

// reference type
anotherVideoMode = avideoMode
anotherVideoMode.frameRate = 90

anotherVideoMode.frameRate
avideoMode.frameRate
```

```
{width 1,024, height 768}

{{width 10, height 10} int...
{{width 1,024, height 76...
{{width 1,024, height 76...
{{width 1,024, height 76...
{{width 1,024, height 76...

{{width 10, height 10} int...
{{width 10, height 10} int...

30.0
60.0

{{width 1,024, height 76...
{{width 1,024, height 76...

90.0
90.0
```

Identity Operators

- Identical to (===) Not Identical to (!==)

- Use it to check whether two constants or variables refers to same single instance.

```
// 5. Identity Operator === , not identical !==
println("Hello world")
if (avideoMode === anotherVideoMode) {
    println("same reference type")
}
```

Properties - lazy stored property

- its initialized value is not calculated until the first time is used. use @lazy keyword before its declaration.

- Global constants and variables are lazy stored property but local variables are not.

@lazy var firstName = “Michael”

Properties - computed property

- not actually store a value. They provide getter and setter.

- If a computer property's setter doesn't define new value to be set, default name **"newValue"** will be used.

- A computer property with getter but no setter is read-only property.

Properties - Property Observers

- Property observers observe and respond to changes in a property’s value. Property observers are called every time a property’s value is set, even if the new value is the same as the property’s current value.

- can add property observer to any stored properties except lazy stored properties.

Properties - Property Observers

- willSet : is called just before the value is stored

- didSet : is called immediately after the new value is stored.

- observer won’t be called in init function.

When to use struct and class?

- The size of geometric shape with width and height property.
- ranges with start and length property.
- 3D coordinating system with x,y,z property.

- Other cases, use CLASS!!

Prevent Overriding

- use final keyword to prevent overriding

- @final var, @final func, @final class func, and @final subscript

Initialization

```
init() {
}
```

DeInitialization

```
deinit {
}
```

Reference

- https://docs.google.com/spreadsheets/d/1CuOfAyymk2B0sp7bQCyjTQo4RMv9C-iA25sIRO-orZk/edit?usp=sharing

- https://developer.apple.com/swift/

- http://eiwaithein.wordpress.com/