



绪论

沈心逸 计算机科学与技术学院

18888921102

shenxinyi@zju.edu.cn

目录



- iOS
- Xcode
- Swift

iOS



- 移动平台操作系统
- 现在移动终端操作系统已经是安卓与 iOS 的天下，因为 iOS 仅苹果一家使用，其它所有品牌使用的全部是安卓或基于安卓的操作系统。
- Android 一家独大，占据74%的市场份额。
- 排名第二的是 iOS，占据26%的市场份额。

iOS



- iOS（原名iPhone OS，自iOS 4后改名为iOS）是苹果公司为移动设备所开发的专有移动操作系统，所支持的设备包括 iPhone、iPod touch 和 iPad。2016年9月发布用于 Apple Watch 的 WatchOS，2019年6月的 WWDC 上 iPadOS 从 iOS 中分离，针对 iPad 做了包括支持鼠标接入等优化。
- 目前最新的版本是 iOS 14

目录



- iOS
- Xcode
- Swift

Xcode



- iOS 集成开发工具
- 运行在 macOS 上
- 目前最新的版本是 Xcode 12.4
- Xcode 具有统一的用户界面设计，编码、测试、调试都在一个简单的窗口内完成。



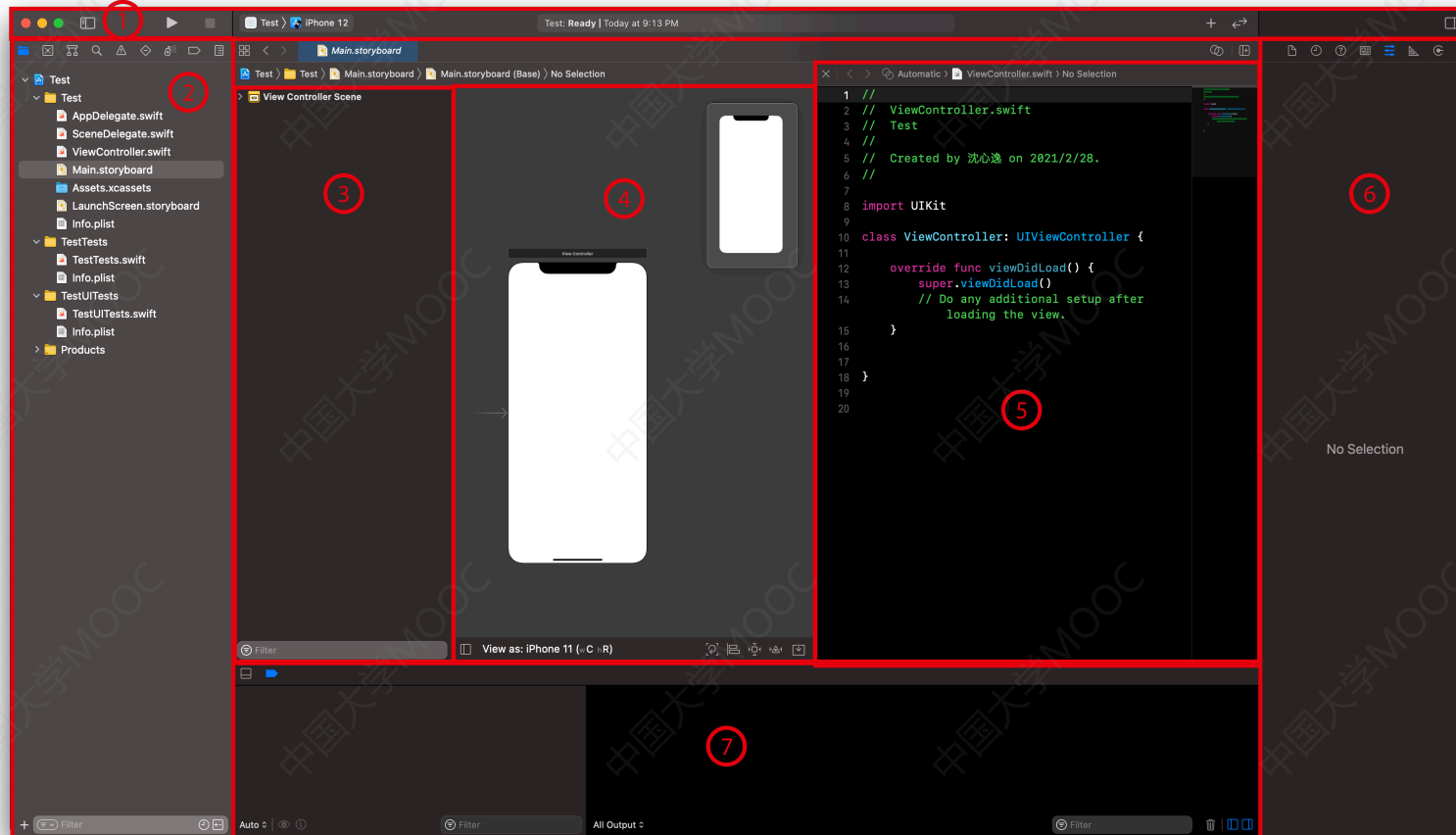
Xcode



1: 主选项区域，从上往下依次是创建新工程、从 Git 上下载已有工程、打开本地已有工程。

2: 历史记录区域，保存了之前打开过的工程，点击可快速打开。

Xcode



Xcode



- 1: 状态栏, 可运行 / 停止 App, 选择模拟器或者真机设备。
- 2: 导航面板, 显示工程下目录结构。
- 3: 大纲面板, 显示 App 界面的层次结构。
- 4: 界面布局区域, 预览手机上查看的效果。
- 5: 代码编辑区域。
- 6: 属性面板, 调整控件的大小、颜色、字体、事件等属性。
- 7: 调试区域, 显示运行中的调试信息。

目录



- iOS
- Xcode
- Swift

Swift



- Swift 是一种支持多编程范式和编译式的编程语言，是用来撰写 macOS、iOS、iPadOS、watchOS 和 tvOS 的语言之一。2014年，其在苹果开发者年会（WWDC）发布。
- 特点：快速、现代、安全、互动
- 目前 Swift 最新的版本是 5.4



Swift



- iOS 开发两大语言
 - Objective-C
 - Swift
- iOS 开发工具
 - Xcode
 - macOS 操作系统的电脑
 - iOS 设备（可选）

Swift



- Swift 的优点
 - 语法简洁
 - 类型安全
 - 函数式编程的支持，极大提高编程效率。
- Swift 的缺点
 - 缺乏 OC 的 Runtime 机制
 - 有历史包袱的大厂仍然在使用 OC，推广不够全面。

Swift - 语法简洁



```
print("Hello, world!")
```

Swift - 语法简洁



- 常量和变量

```
var a = 1  
let b = 2
```

```
a = 3
```



```
b = 4
```



Swift - 语法简洁



- 命名习惯：驼峰式

```
let maximumNumberOfLoginAttempts = 10  
var currentLoginAttempt = 0
```


Swift - 类型安全



- 基本类型

```
let eight: Int = 8
let pi: Double = 3.14
let name: String = "Coco"
let lightOn: Bool = false
```

Swift - 类型安全



```
let a: Int = 1
var b: Double = 2.0
let c = a + b
```



```
let a: Int = 1
var b: Double = 2.0
let c = Double(a) + b
```



Swift - 函数式编程



- 例子：计算每个学生各科的平均分，并输出第一名的姓名。

```
struct Student {  
    let name: String  
    let scores: [科目: Int]  
}
```

```
enum 科目: String, CaseIterable {  
    case 语文, 数学, 英语, 物理  
}
```

```
let s1 = Student(  
    name: "Jane",  
    scores: [.语文: 86, .数学: 92, .英语: 73, .物理:  
88]  
)
```

```
let s2 = Student(  
    name: "Tom",  
    scores: [.语文: 99, .数学: 52, .英语: 97, .物理:  
36]  
)
```

```
let s3 = Student(  
    name: "Emma",  
    scores: [.语文: 91, .数学: 92, .英语: 100, .物理:  
99]  
)
```

```
let students = [s1, s2, s3]
```

Swift - 函数式编程



- 传统的指令式编程

```
var best: (Student, Double)?
for s in students {
    var totalScore = 0
    for key in 科目.allCases {
        totalScore += s.scores[key] ?? 0
    }
    let averageScore = Double(totalScore) / Double(科目.allCases.count)
    if let temp = best {
        if averageScore > temp.1 {
            best = (s, averageScore)
        }
    } else {
        best = (s, averageScore)
    }
}
if let best = best {
    print("最高平均分: \(best.1), 姓名: \(best.0.name)")
} else {
    print("students 为空")
}
```

Swift - 函数式编程



- 函数式编程

```
func average(_ scores: [科目: Int]) -> Double {  
    return Double(scores.values.reduce(0, +)) /  
    Double(科目.allCases.count)  
}
```

```
let bestStudent = students  
    .map { ($0, average($0.scores)) }  
    .sorted { $0.1 > $1.1 }  
    .first
```

学习资源



- 官方教程 the swift programming language
 - Swift 一手资料
 - 英文版: <https://docs.swift.org/swift-book/>
 - 中文版: <https://swiftgg.gitbook.io/swift/>
- App Development with Swift
 - UIKit 开发优秀教程
 - 外区 Apple 图书商店免费下载

学习资源



- 中国大学 MOOC 《Swift 创新导论》
 - <https://www.icourse163.org/course/ZJU-1450024180>
 - 理论 + 大量实践
 - 对话、案例分享和专题讨论
- Stanford University: Developing iOS 11 Apps with Swift
 - <https://itunes.apple.com/cn/course/id1309275316>

