Swift 枚举 Swift 类

Swift 结构体

Swift 结构体是构建代码所用的一种通用且灵活的构造体。

我们可以为结构体定义属性（常量、变量）和添加方法，从而扩展结构体的功能。

与 C 和 Objective C 不同的是：

- 结构体不需要包含实现文件和接口。
- 结构体允许我们创建一个单一文件，且系统会自动生成面向其它代码的外部接口。

结构体总是通过被复制的方式在代码中传递，因此它的值是不可修改的。

语法

我们通过关键字 struct 来定义结构体：

```
struct nameStruct {
   Definition 1
   Definition 2
   ……
   Definition N
}
```

实例

我们定义一个名为 MarkStruct 的结构体 ，结构体的属性为学生三个科目的分数，数据类型为 Int：

```
struct MarkStruct{
   var mark1: Int
   var mark2: Int
   var mark3: Int
}
```

我们可以通过结构体名来访问结构体成员。

结构体实例化使用 **let** 关键字：

```
import Cocoa

struct studentMarks {
   var mark1 = 100
   var mark2 = 78
   var mark3 = 98
}
```

```
let marks = studentMarks()
print("Mark1 是 \(marks.mark1)")
print("Mark2 是 \(marks.mark2)")
print("Mark3 是 \(marks.mark3)")
```

以上程序执行输出结果为：

```
Mark1 是 100
Mark2 是 78
Mark3 是 98
```

实例中，我们通过结构体名 'studentMarks' 访问学生的成绩。结构体成员初始化为mark1, mark2, mark3，数据类型为整型。

然后我们通过使用 **let** 关键字将结构体 studentMarks() 实例化并传递给 marks。

最后我们就通过 . 号来访问结构体成员的值。

以下实例化通过结构体实例化时传值并克隆一个结构体：

```
import Cocoa

struct MarksStruct {
   var mark: Int

   init(mark: Int) {
      self.mark = mark
   }
}
var aStruct = MarksStruct(mark: 98)
var bStruct = aStruct // aStruct 和 bStruct 是使用相同值的结构体!
bStruct.mark = 97
print(aStruct.mark) // 98
print(bStruct.mark) // 97
```

以上程序执行输出结果为：

```
98
97
```

结构体应用

在你的代码中，你可以使用结构体来定义你的自定义数据类型。

结构体实例总是通过值传递来定义你的自定义数据类型。

按照通用的准则，当符合一条或多条以下条件时，请考虑构建结构体：

- 结构体的主要目的是用来封装少量相关简单数据值。

- 有理由预计一个结构体实例在赋值或传递时，封装的数据将会被拷贝而不是被引用。
- 任何在结构体中储存的值类型属性，也将会被拷贝，而不是被引用。
- 结构体不需要去继承另一个已存在类型的属性或者行为。

举例来说，以下情境中适合使用结构体：

- 几何形状的大小，封装一个`width`属性和`height`属性，两者均为`Double`类型。
- 一定范围内的路径，封装一个`start`属性和`length`属性，两者均为`Int`类型。
- 三维坐标系内一点，封装`x`，`y`和`z`属性，三者均为`Double`类型。

结构体实例是通过值传递而不是通过引用传递。

```
import Cocoa

struct markStruct{
    var mark1: Int
    var mark2: Int
    var mark3: Int

    init(mark1: Int, mark2: Int, mark3: Int){
        self.mark1 = mark1
        self.mark2 = mark2
        self.mark3 = mark3
    }
}

print("优异成绩:")
var marks = markStruct(mark1: 98, mark2: 96, mark3:100)
print(marks.mark1)
print(marks.mark2)
print(marks.mark3)

print("糟糕成绩:")
var fail = markStruct(mark1: 34, mark2: 42, mark3: 13)
print(fail.mark1)
print(fail.mark2)
print(fail.mark3)
```

以上程序执行输出结果为：

```
优异成绩:
98
96
100
糟糕成绩:
```

```
34
42
13
```

以上实例中我们定义了结构体 markStruct，三个成员属性：mark1, mark2 和 mark3。结构体内使用成员属性使用 self 关键字。

从实例中我们可以很好的理解到结构体实例是通过值传递的。

← Swift 枚举 Swift 类 →