1. 枚举与结构体

枚举也是一种数据类型，一般利用枚举的作用是将事物的选择限定到一个范围中。如一星期只有七天，在编程中就可以用枚举来限定星期数。

6.1枚举类型的创建与应用

swift中使用枚举类型，主要分为两种，一种是创建任意类型的枚举类型，另外一种是创建指定数据类型的枚举类型。以下是这两种枚举类型的定义：

1. 任意枚举类型

enum 枚举名{

}

任意枚举类型在枚举的相关值中用的比较多，一般就是不为枚举类型设置数据类型。

2、指定枚举类型

enum 枚举名:数据类型{

}

指定枚举类型在枚举的原始值中用的比较多，一般就是为枚举类型设置数据类型，来限定枚举中枚举成员的数据类型。

例如我们定义表示星期的枚举：

enum week{

case SUNDAY

case MONDAY

case TUESDAY

case WEDNESDAY

case THURSDAY

case FRIDAY

case SATURDAY}

var weekDay = week.SUNDAY

weekDay = .THURSDAY

switch weekDay{

case .SUNDAY: print("星期天")

case .Monday: print("星期一")

case .TUESDAY: print("星期二")

case .WEDNESDAY: print("星期三")

case .THURSDAY: print("星期四")

case .FRIDAY: print("星期五")

case .SATURDAY: print("星期六")

default: print("没有匹配值")}

以上程序执行输出结果为：

星期四

在示例中，需要注意到几个点，如果一个变量的类型已经确认为某个枚举类型，那么开发者在进行变量赋值的时候，是可以直接将枚举名省略掉，直接使用点语法获取枚举值即可。

6.2枚举的原始值与相关值

枚举的原始值就是给定义的枚举成员，再赋上一个值。对于枚举类型是整数型，默认的原始值是从零开始逐渐递增，而枚举类型是其他类型的，需要开发者指定初始值。

6.2.1枚举的原始值

前面创建的枚举中，并没有为枚举类型初始化一个值，而枚举的原始值就是在初始化枚举类型时，为其附上一个初始值。示例如下：

//为枚举类型指定一个原始值

enum original:Int{

case a = 1

case b = 2

case c = 3

case d = 4

}

通过枚举类型提供的rawValue属性来获取枚举的原始值，示例如下：

//创建枚举变量

var num = original.a

//获取num枚举变量的原始值“a”

var rawValue = num.rawValue

6.2.2枚举的相关值

在Swift语言中，枚举的相关值是在枚举中声明的枚举可以是包含不同的数据类型，且值不是事先设定的。相关值是当你在创建一个基于枚举成员的新常量或变量时才会被设置，即每次值都可以不同。

以下实例中我们定义一个名为Student的枚举类型，它可以是Name的一个字符串(String),或者是Mark的一个相关值(Int,Int,Int)。示例如下：

enum Student{

case Name(String)

case Mark(Int,Int,Int)

}

var stuDetails = Student.Name(“HDU”)

var studMarks = Student.Mark(10,10,10)

switch studmarks{

case .Name(let studName):

print(“学生的名字:\(studName)”)

case .Mark(let Mark1,let Mark2,let Mark3):

print(“学生的成绩:\(Mark1),\(Mark2),\(Mark3)”)

}

以上程序执行输出结果为：

学生的成绩是：10，10，10

6.3Swift结构体

在Swift语言中，结构体是与类的使用方法特别的相似。

与其他的语言不同的是，Swift语言的结构体不仅仅只是可以定义属性，而且还可以添加方法，与类的功能特别的相似。

语法

我们通过关键字struct来定义结构体：

struct nameStruct{

属性

方法

}

实例如下：

struct dog{

var name:String

var color:String

func bark(){

print(“汪汪”)

}

}

从以上的实例可以看出，Swift中的结构体与类的用法特别的相似，但是结构体和类还是有本质的区别的。

在结构体中，数据的传递是以复制的方式进行传递的。虽然结构体中能够声明方法，但是结构体是没有继承和多态的性质的。

本章小节

本章主要给读者介绍了Swift语言中的枚举和结构体，相对与其他语言中的枚举类型，Swift中的枚举显得比较灵活。而对于结构体来说是比较大的变化，结构体中能够声明方法，使其许多的应用上和类非常的相似。