Go语言学习(三)枚举和类型

原创 2016年03月23日 21:44:07 4998

package main

import "fmt"

//1.枚举的使用

// 枚举指一系列相关的常量，比如下面关于一个星期中每天的定义。通过上一节的例子，我们

// 看到可以用在 const 后跟一对圆括号的方式定义一组常量，这种定义法在Go语言中通常用于定义

// 枚举值。Go语言并不支持众多其他语言明确支持的 enum 关键字。

// 下面是一个常规的枚举表示法，其中定义了一系列整型常量：

const(

Sunday = iota //0

Monday //1

Tuesday //2

Wedenesday //3

Thursday //4

Friday //5

Saturday //6

//同Go语言的其他符号（symbol）一样，以大写字母开头的常量在包外可见。

//以上例子中 numberOfDays 为包内私有，其他符号则可被其他包访问。

numberOfDays

)

//2.类型

/\*

Go语言内置以下这些基础类型

布尔类型 : bool

整型: int,int8,int16,int32,int64,uint,uint8,uint16,uint32,uint64,uintptr

浮点类型:float32,float64

复数类型: complex,complex64,complex128

字符串类型:string

字符类型:rune

错误类型: error

此外,Go语言也支持以下符合类型

指针:pointer

数组:array

切片:slice

字典:map

通道:chan

结构体:struct

接口:interface

\*/

func main() {

fmt.Println("Sunday=",Sunday)

fmt.Println("Monday=",Monday)

fmt.Println("Tuesday=",Tuesday)

fmt.Println("Wedenesday=",Wedenesday)

fmt.Println("Thursday=",Thursday)

fmt.Println("Friday=",Friday)

fmt.Println("Saturday=",Saturday)

fmt.Println("numberOfDays=",numberOfDays)

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

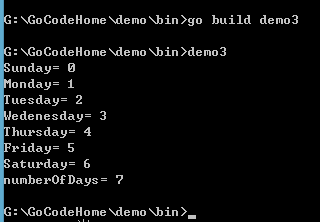
49

50

51

52

运行结果：



文章来源：<http://blog.csdn.net/mchenys/article/details/50966938>

版权声明：本文为博主原创文章，未经博主允许不得转载。