Go语言学习(十二)面向对象编程-结构体

原创 2016年05月09日 22:26:46

标签：

[go语言](http://so.csdn.net/so/search/s.do?q=go%E8%AF%AD%E8%A8%80&t=blog" \t "http://blog.csdn.net/mchenys/article/details/_blank)

922

1.结构体的初始化方式

例如自定义一个结构体

package main

import(

"fmt"

)type Rect struct{ //type和struct为关键字

x,y float64 //结构体成员

widh,height float64

}

func (r \*Rect) Area() float64{

return r.width \* r.height

}

func main(){

//初始结构体的几种方式:

rect1 := new(Rect)

rect2 := &Rect{}

rect3 := &Rect{0,0,100,200}

rect4 := &Rect{widh:100,height:200}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

在Go语言中，未进行显式初始化的变量都会被初始化为该类型的”零值”，   
例如bool类型的零值为false,int类型的零值为0,string 类型的零值为空字符串。

在Go语言中没有构造函数的概念,对象的创建通常交由一个全局的创建函数来完成,以   
NewXXX来命名,表示“构造函数”：

func newRect(x,y,widh,height float64) \*Rect{

return &Rect{x,y,widh,height}

}

1

2

3

这一切非常自然，开发者也不需要分析在使用了 new 之后到底背后发生了多少事情。在Go   
语言中，一切要发生的事情都直接可以看到。

2.嵌套结构体

package mainimport(

"fmt"

)func main(){

t := space{plane{line{3}, 5}, 7}

fmt.Println(t.x, t.y, t.z) //3 5 7

}

type line struct {

x int

}type plane struct {

line

y int

}type space struct {

plane

z int

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

3.结构体的特点:

1)结构体可以储存不同的类型   
2)在内存中占据连续的内存空间   
3)结构体每一个项所占用的内存大小不一定相同   
4)结构体支持组合,即结构体可以保护结构体   
5)通过操作结构体的项名t.x、t.y、t.z来获取字段值   
6)判断两个结构体是否相同,需要看结构体的类型是否相同，   
然后看项的顺序、项的名称、项的类型等等.   
7)结构体的成员初始化是通过操作字段赋值来完成

文章来源：http://blog.csdn.net/mchenys/article/details/51356824

版权声明：本文为博主原创文章，未经博主允许不得转载。