



# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

## 匿名字段

# 目录



PART 01  
匿名字段



PART 02  
同名字段



PART 03  
指针类型匿名字段



PART 04  
多重继承



# 匿名字段

1.匿名字段的定义

2.创建与初始化

3.成员操作



### 什么叫匿名字段？

定义一个struct，定义的时候是字段名与其类型一一对应，实际上Go语言支持只提供类型，而不写字段名的方式，也就是匿名字段，或称为嵌入字段。



## 匿名字段创建与初始化

```
type Person struct {  
    id    int  
    name  string  
    age   int  
}  
type Student struct {  
    Person ← 匿名字段  
    score float64  
}
```

以上代码通过匿名字段实现了继承，将公共的属性封装在Person中，在Student中直接包含Person,那么Student中就有了Person中所有的成员，Person就是匿名字段。

**注意：**Person匿名字段，只有类型，没有名字。

给Student赋值，具体初始化的方式如下：

```
func main() {  
    //顺序初始化  
    var s1 Student = Student{Person{101, "mike", 18}, 98.5}  
    fmt.Println("s1 = ", s1)  
}
```



注意



## 成员操作

创建完成对象后，可以根据对象来操作对应成员属性，是通过“.”运算符来完成操作的。

案例：

```
var s1 Student = Student{Person{101, "mike", 18}, 98.5}
s1.age = 20
s1.Person.id = 120
s1.score = 99
fmt.Println(s1)
```

由于Student继承了Person，所以Person具有的成员，Student也有，所以根据Student创建出的对象可以直接对age成员项进行修改。

由于在Student中添加了匿名字段Person，所以对象s1,也可以通过匿名字段Person来获取age，进行修改。当然也可以进行如下修改：

```
var s1 Student = Student{Person{101, "mike", 18}, 98.5}
s1.Person = Person{112, "zhangsan", 20}
fmt.Println(s1)
```

直接给对象s1中的Person成员（匿名字段）赋值。

通过以上案例我们可以总结出，根据类（结构体）可以创建出很多的对象，这些对象的成员（属性）是一样的，但是成员（属性）的值是可以完全不一样的。





# 同名字段

## 1.同名字段

## 同名字段

```
type Person struct {  
    id    int  
    name  string  
    age   int  
}  
type Student struct {  
    Person  
    name  string //和Person同名了  
    score float64  
}
```

思考：在Student中也加入了一个成员name，这样与Person重名了，那么如下代码是给Student中name赋值还是给Person中的name 进行赋值？

```
var s1 Student  
s1.name = "zhangsan"  
fmt.Printf("s1 = %+v\n", s1)  
/*
```

在操作同名字段时有一个基本的原则：

如果能够在自己对象所属的类（结构体）中找到对应的成员，那么直接进行操作，如果找不到就去对应的父类（结构体）中查找。这就是所谓的就近原则。





# 指针类型匿名字段

## 1. 指针类型匿名字段

## 指针类型匿名字段

结构体（类）中的匿名字段的类型，也可以是指针。

```
type Person struct {  
    id int  
  
    name string  
    age  int  
}  
type Student struct {  
    *Person  
  
    score float64  
}  
  
func main() {  
  
    var s Student  
    s.Person = new(Person)  
    s.id = 103  
    s.name = "wangwu"  
    s.age = 19  
    s.score = 89  
    fmt.Println(s.id, s.name, s.age, s.score)  
}
```

使用new分配空间

`new()`的作用是分配空间，`new()`函数的参数是一个类型，这里为Person结构体类型，返回值为指针类型，所以赋值给\*Person，这样\*Person也就指向了结构体Person的内存。



# 多重继承

## 1.多重继承



## 多重继承

---

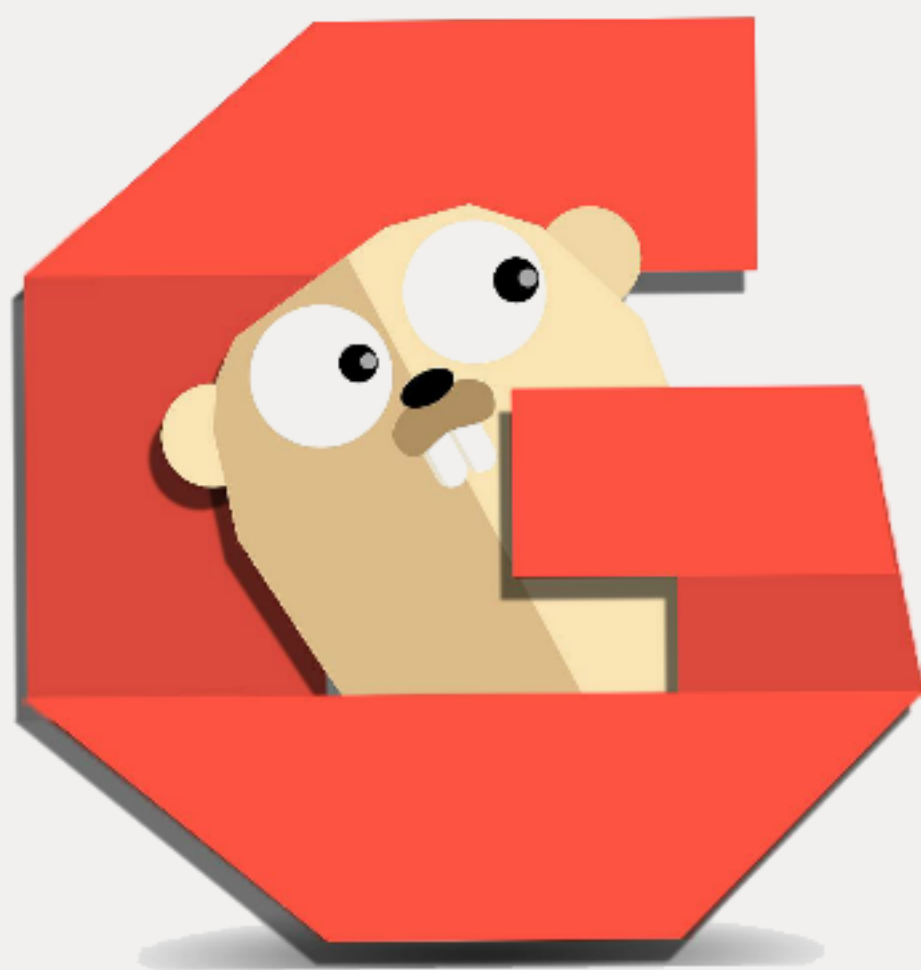
多重继承指的是一个类可以继承另外一个类，而另外一个类又可以继承别的类，比如A类继承B类，而B类又可以继承C类，这就是多重继承。

注意：尽量在程序中，减少多重继承，否则会增加程序的复杂度。



## 实例演示





# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看