

# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

## 方法的继承与重写

# 目录



PART 01  
**方法继承**



PART 02  
**方法重写**



# 方法继承

## 1.方法继承

## 方法继承

方法继承：父类的方法可以被子类继承使用。

```
//  
type Person struct {  
    name string //名字  
    sex  byte   //性别, 字符类型  
    age  int    //年龄  
}  
  
//Person类型, 实现了一个方法  
func (tmp *Person) PrintInfo() {  
    fmt.Printf("name=%s, sex=%c, age=%d\n", tmp.name, tmp.sex, tmp.age)  
}  
  
//有个学生, 继承Person字段, 成员和方法都继承了  
type Student struct {  
    Person //匿名字段  
    id     int  
    addr   string  
}  
  
func main() {  
    s := Student{Person{"mike", 'm', 18}, 666, "bj"}  
    s.PrintInfo()  
}
```

思路：

- 1、找出公共的属性，定义父类(结构体)
- 2、找出公共的方法，定义在父类(结构体)
- 3、找出独有的属性，定义在自己的结构体(类)中
- 4、找出独有的方法，定义在自己的结构体(类)中



# 方法重写

## 1.方法重写



## 方法重写

---

如果子类(结构体)中的方法名与父类(结构体)中的方法名同名，在调用的时候是先调用子类(结构体)中的方法，这就方法的重写。

所谓的重写：就是子类(结构体)中的方法，将父类中的相同名称的方法的功能重新给改写了。

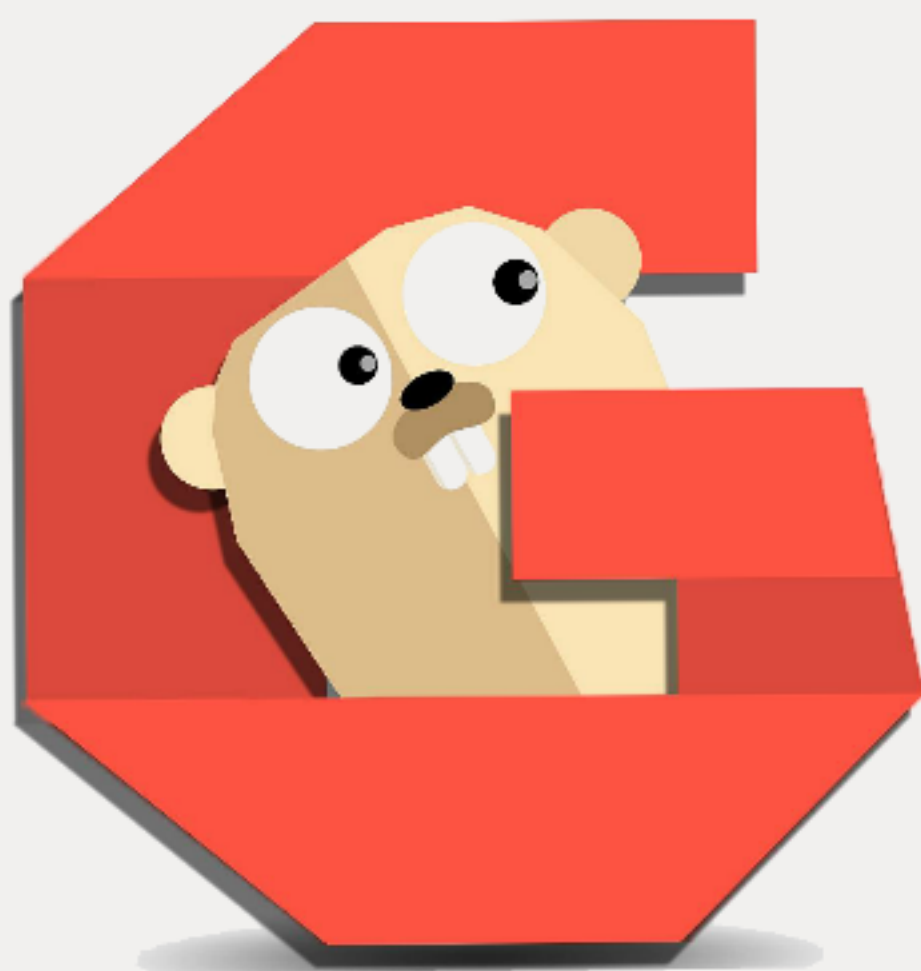
子类(结构体)继承父类(结构体)的方法，在调用对象继承方法的时候，调用和执行的是父类的实现。但有时候需要对子类中的继承方法有不同的实现方式。



## 实例演示







江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看