#### 结构体

题一、

package main  
  
import (  
 "fmt"  
)  
  
  
type student struct {  
 name string  
 age int  
}  
func main() {  
 m := make(map[string]\*student)  
 stus := []student{  
 {name: "ljy", age:30},  
 {name: "liujiadong", age: 27},  
 {name: "dx", age: 26},  
 }  
 //m[stus[0].name] = &stus[0] //①  
 //m[stus[1].name] = &stus[1]  
 //m[stus[2].name] = &stus[2]  
 for \_, s := range stus{ // ②  
 //fmt.Println(s.name)  
 m[s.name] = &s  
 fmt.Printf("%s->%p\n",s.name, &s)  
 }  
 fmt.Println(m)  
 for k, v := range m {  
 fmt.Println(k, "==>", v.name) //③  
 }  
}

上题中, 在②中， s变量的指针已经固定， 循环体中赋予m[s.name]的值为地址， 所以在循环三次后， s的地址不变， 但是值已经更改为最后一次的值， 导致m[s.name]的值也更改为最后一次的值&{name:"dx", age:26},所以在③中， v.name的值是固定的， 但采用①的方式给m赋值的话，在③时， 打印出的值为对应的赋值， 所以在循环遍历数组，通过指针给变量赋值时， 应采用索引的方式，如

for i=0;i<len(stus);i++ {  
 m[stus[i].name] = &stus[i]  
}

##### 指针类型的接收者

指针类型的接收者由一个结构体的指针组成， 由于指针的特性，调用方法时修改接收者的任意成员变量， 其值都能有效被修改，如果非指针类型的接收者， 调用方法修改成员变量仅在该方法内有效，

#### 基础数据类型

##### 整型