第三章

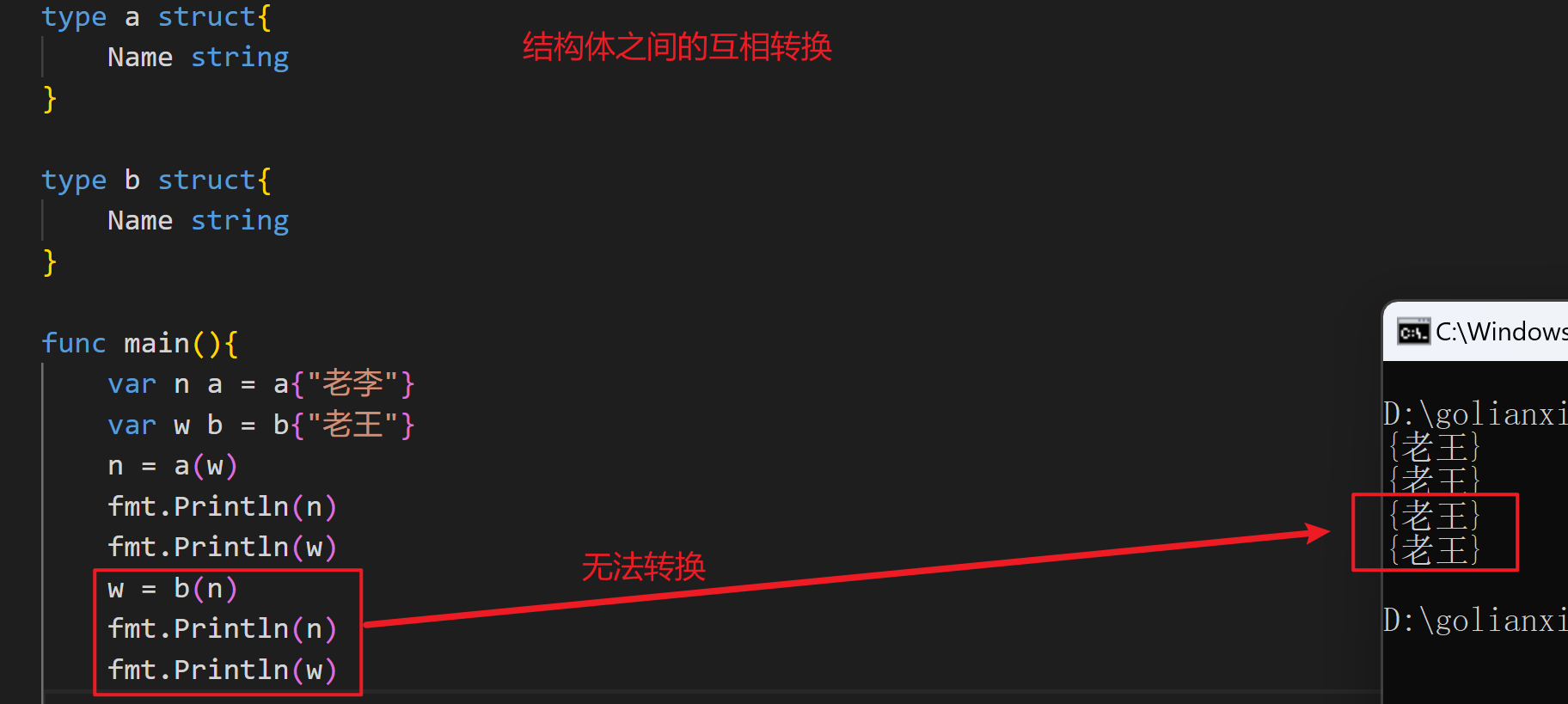
面向对象：结构体

【1】结构体赋值的四种方式：

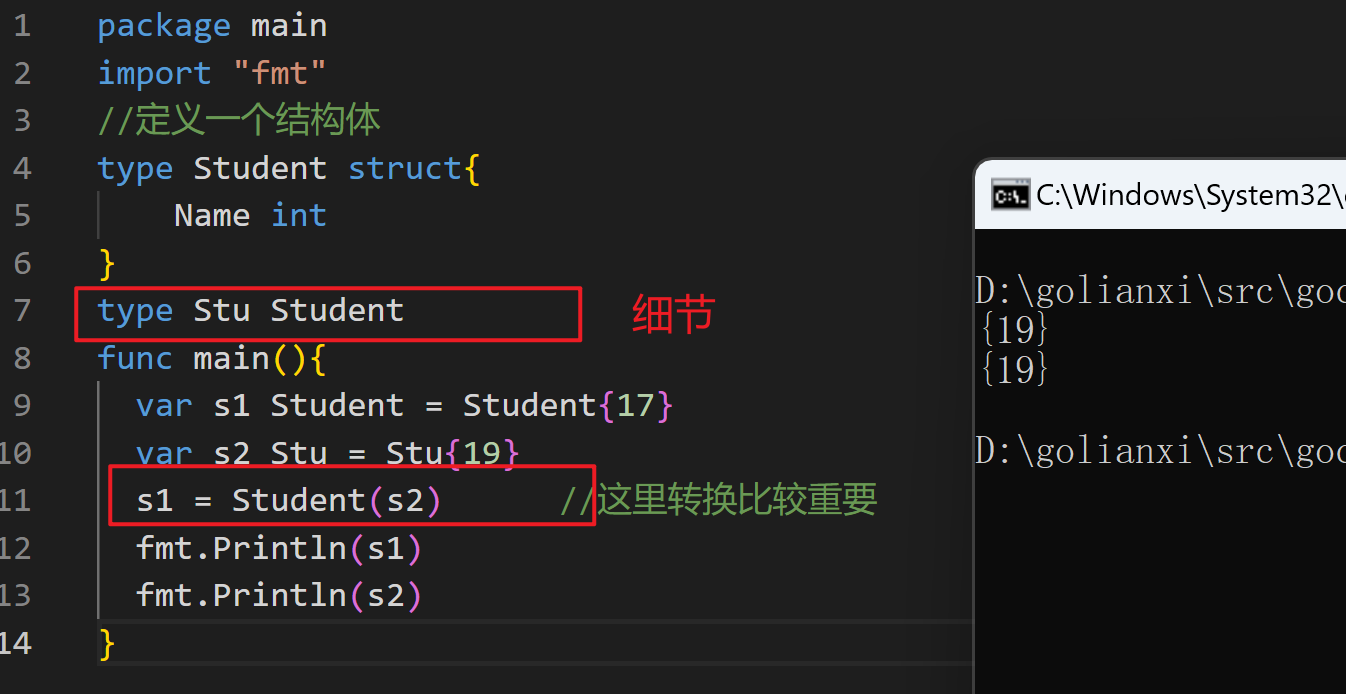
 【2】结构体是用户单独定义的类型，和其他类型进行转换时需要有完全相同的字段（名字，个数和类型）



问题：只能单方面转换，不能完全转换

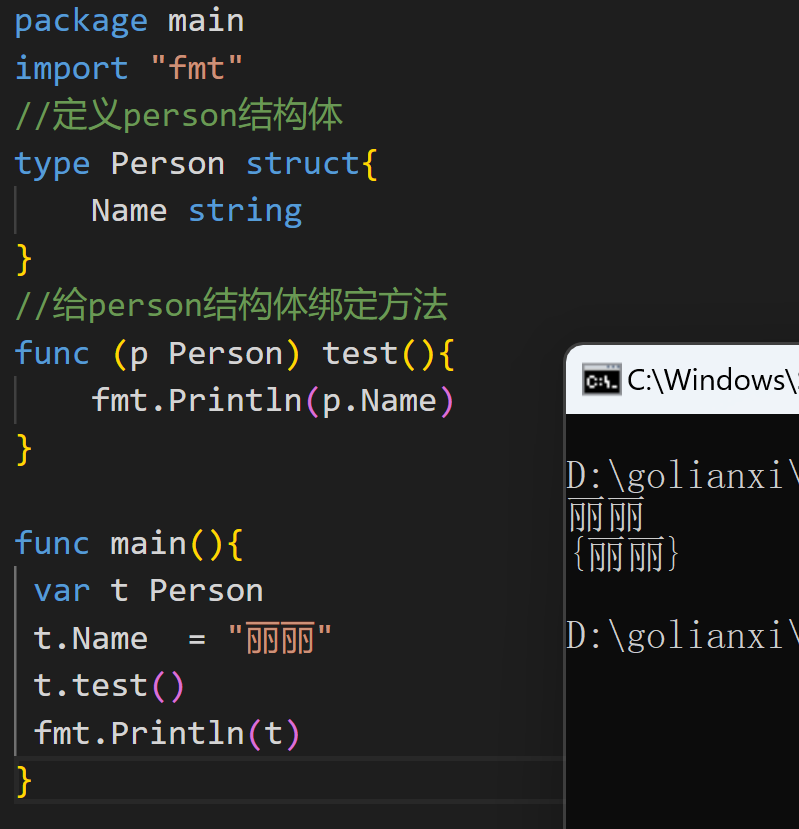


【3】结构体进行type重新定义（相当于取别名），go会认为是新的数据类型，但是相互间可以强制转换



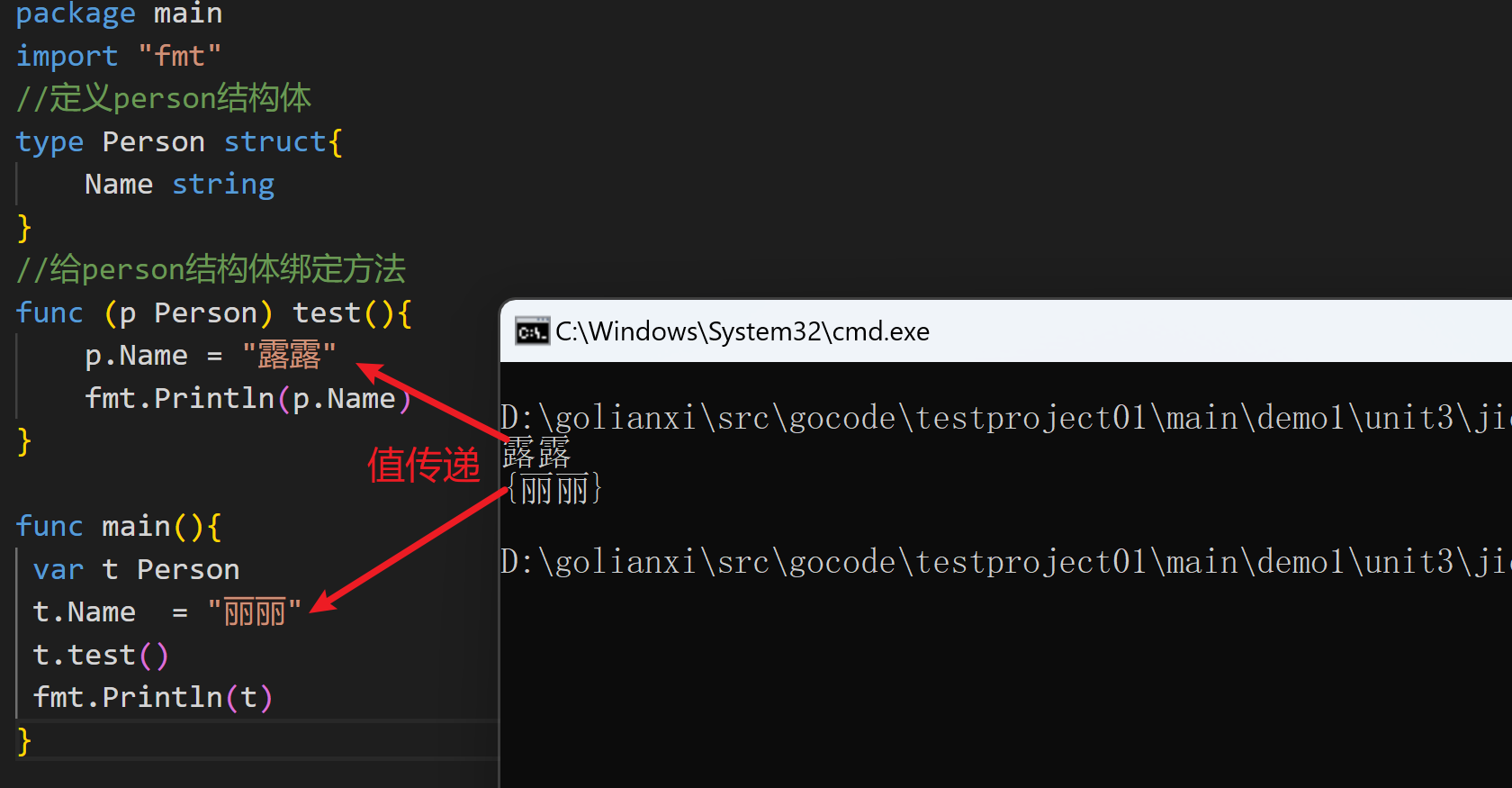
【4】方法是作用在指定的数据类型上，和指定的数据类型绑定，因此自定义类型，都可以有方法，而不仅仅是struct。

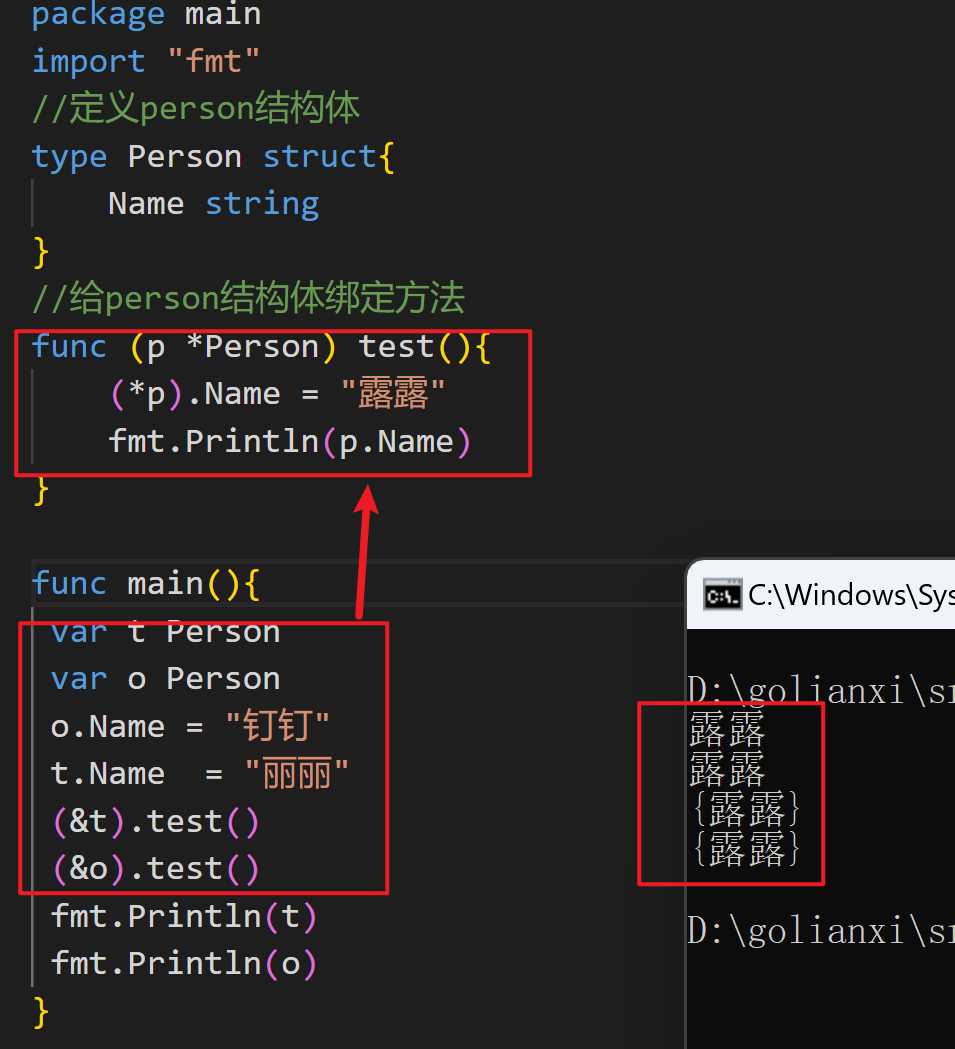
方法的声明和调用格式



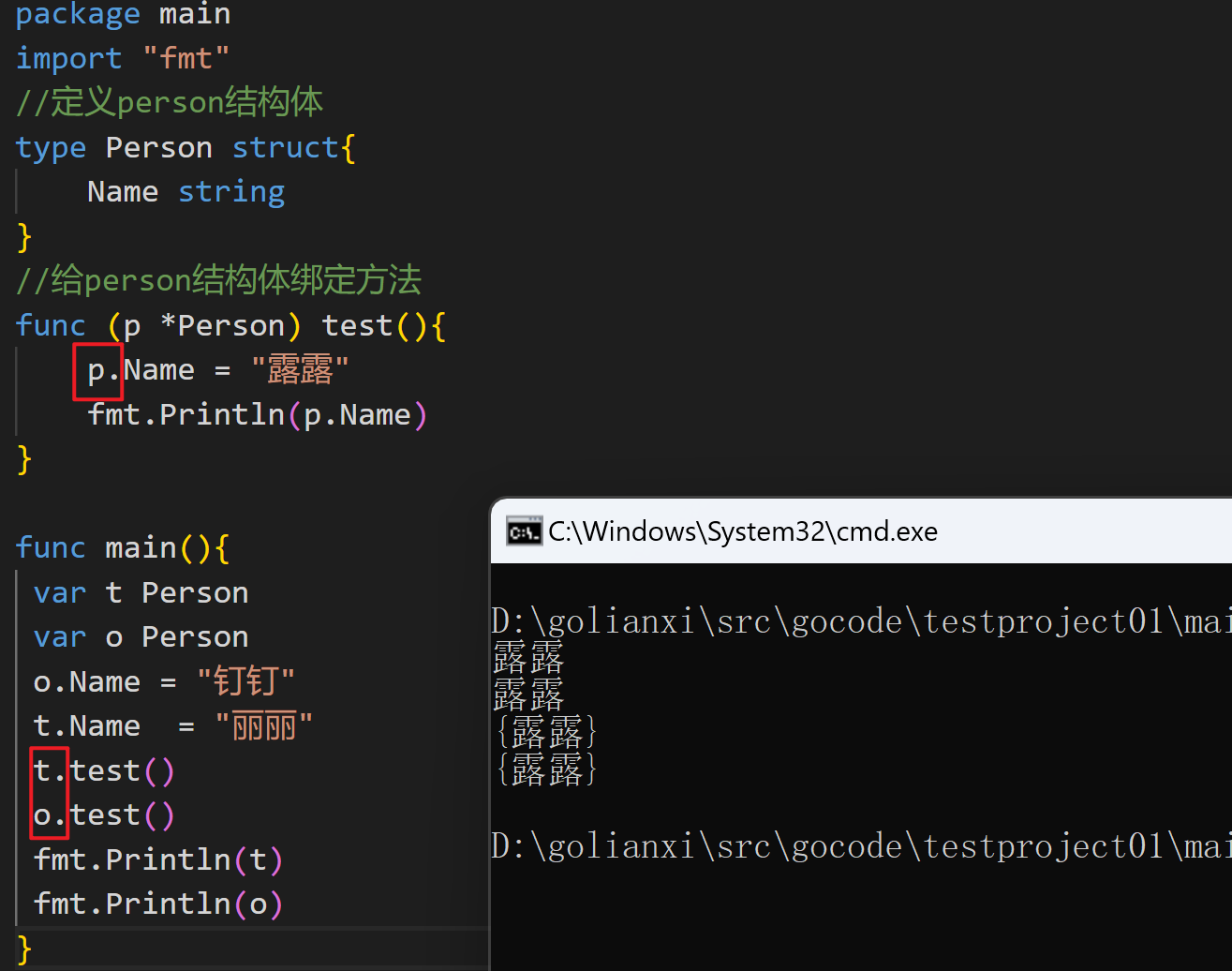
Ps：

1. test方法中参数名字可以自定义
2. 结构体person和test方法绑定，调用test方法必须靠指定的类型：person
3. test方法已经确定，如果用其他类型变量调用test方法一定会报错
4. 结构体对象传入test方法中，是值传递，和函数参数传递一致

 （5）如果程序员希望在方法中，改变结构体变量的值，可以通过结构体指针的方式实现

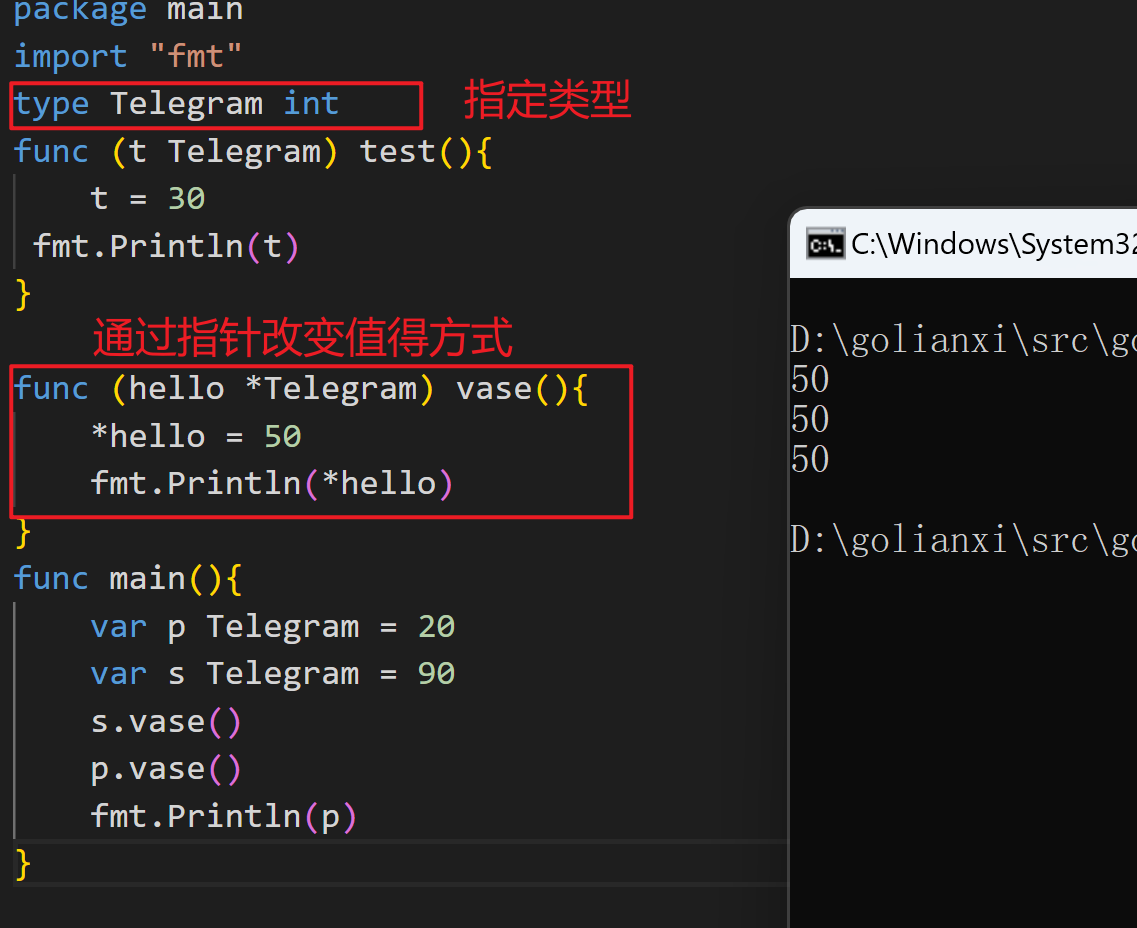


通过指针改变结构体变量的方式可以简化



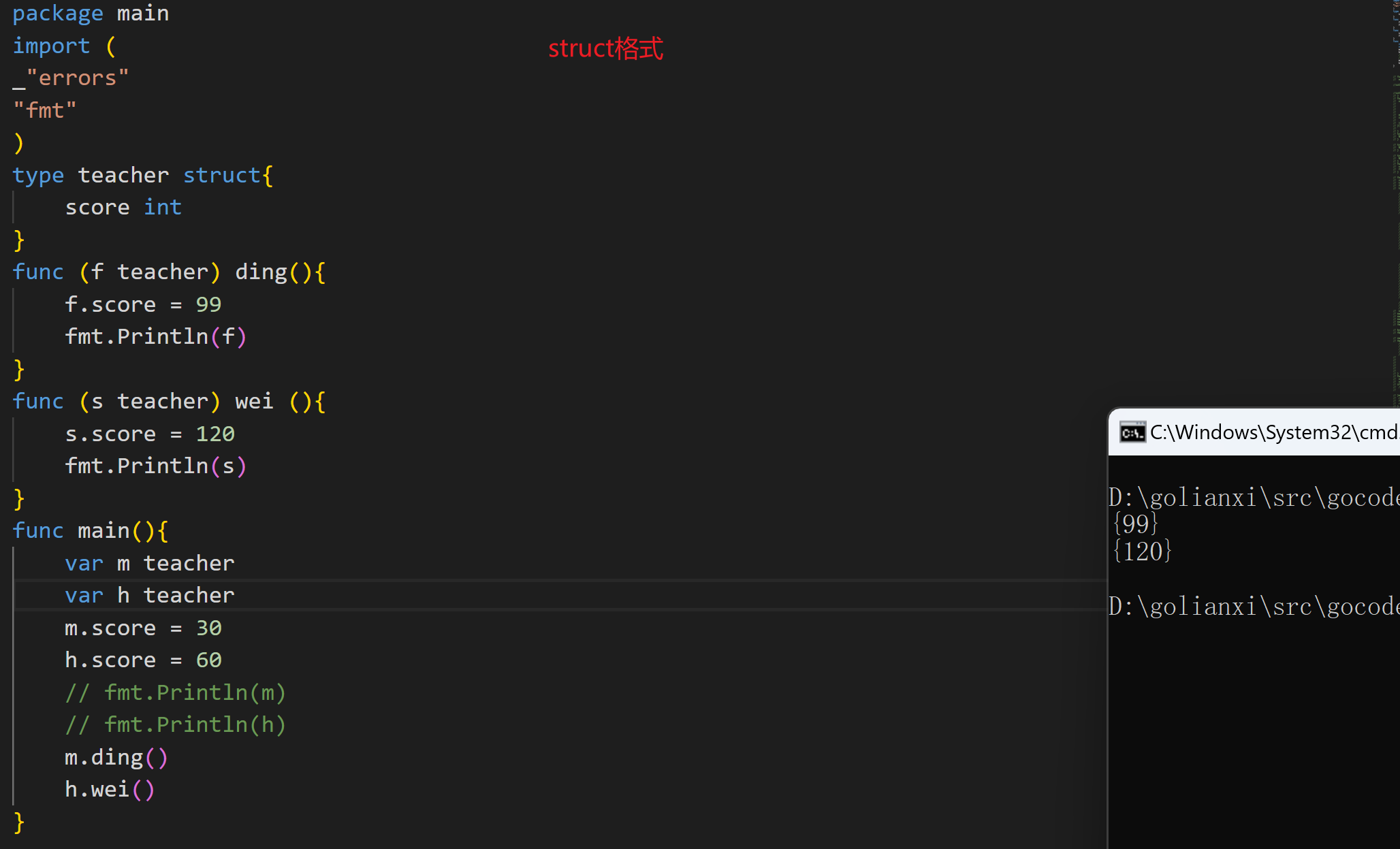
Ps：因为底层编译器做了优化，会自动转化

1. golang中的方法作用在指定的数据类型上，和指定的数据类型绑定，因此自定义类型都可以有方法，而不仅仅是struct，比如int，float32等都可以有方法

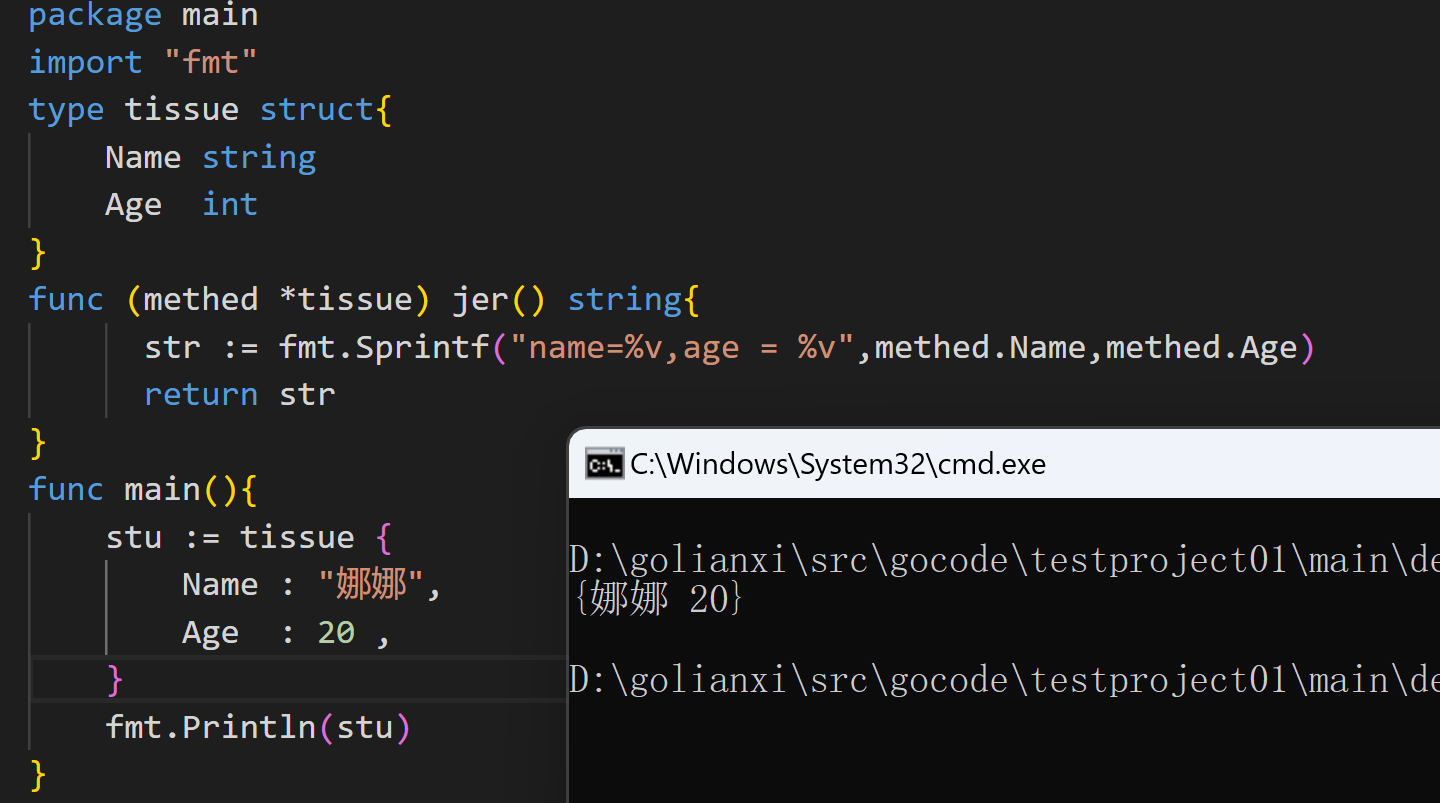


Ps：这种格式要注意与struct格式的差异点





1. 方法的访问范围控制的规则，和函数一样。方法首字母大写，可以在本包和其他包访问，方法首字母小写，只能在本包访问
2. 如果一个类型实现了string()这个方法，那么fmt.println默认会调用这个变量的string()进行输出



Ps：以后定义结构体的话，常定义string()作为输出结构体信息的方法，在fmt.println会自动调用

1. 方法和函数的区别

【1】绑定指定类型：

方法：需要绑定指定数据类型

函数：不需要绑定指定数据类型

【2】调用方式不一样：

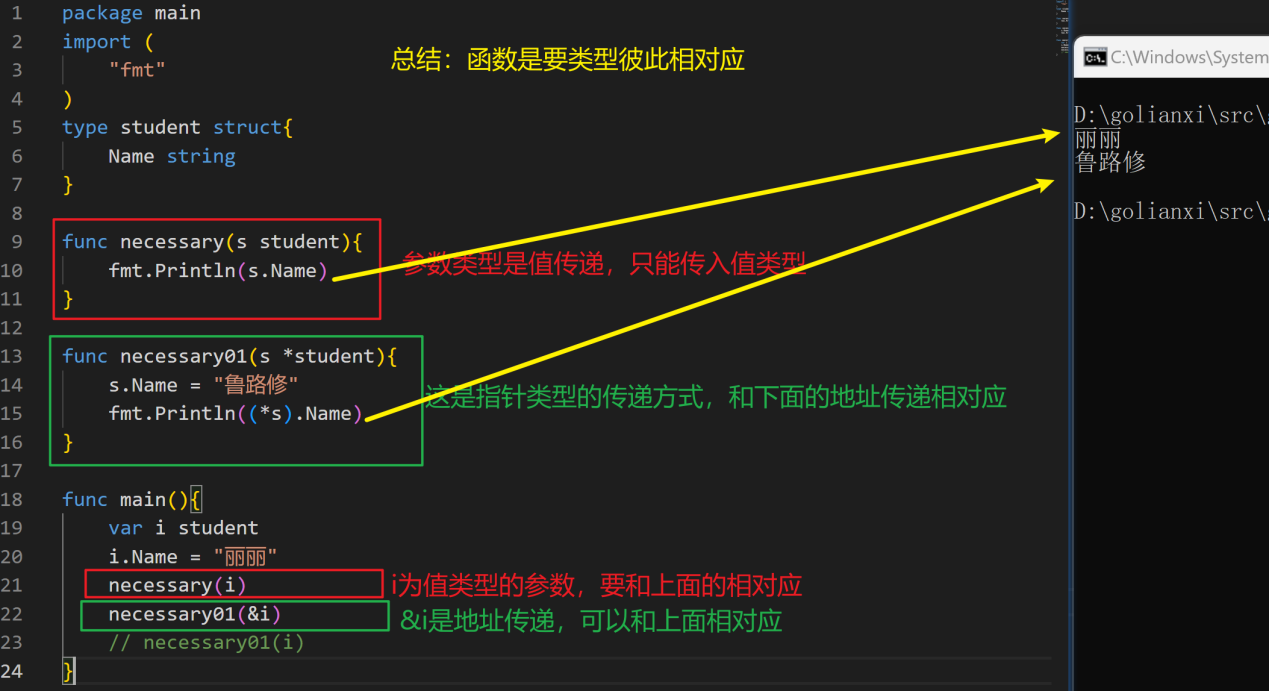
函数的调用方式：

函数名（实参列表）

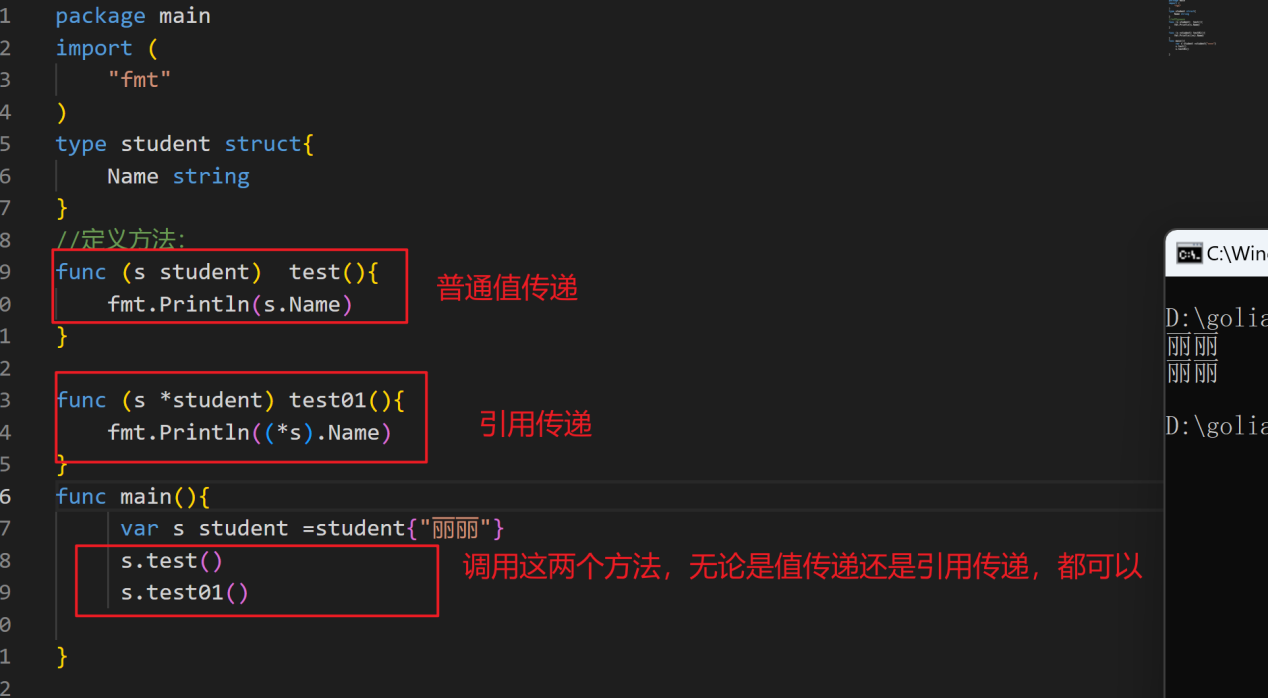
方法的调用方式：变量.方法名（实参列表）



【3】对于函数来说，参数类型对应是什么就要传入什么

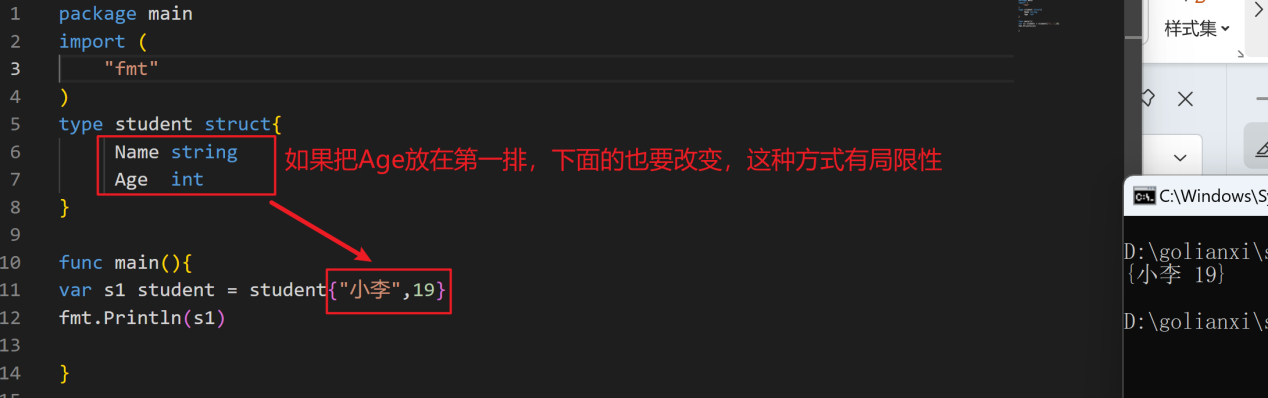


【4】对于方法来说，接受者为值类型，可以传入指针类型，接收者为指针类型，可以传入值类型

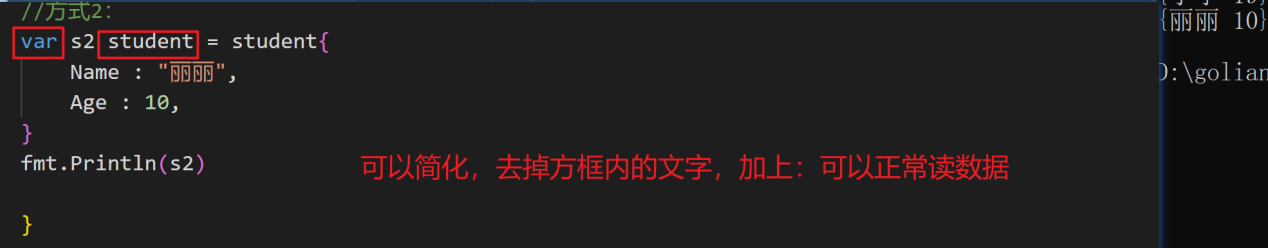


创建结构体实例时指定字段值

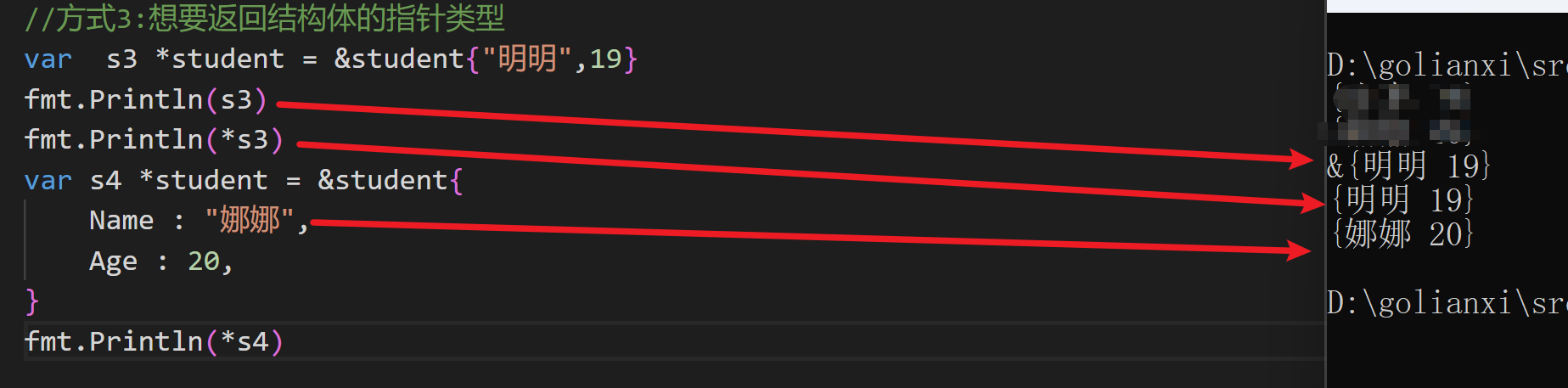
【1】方式1：按照顺序赋值操作



【2】方式2：

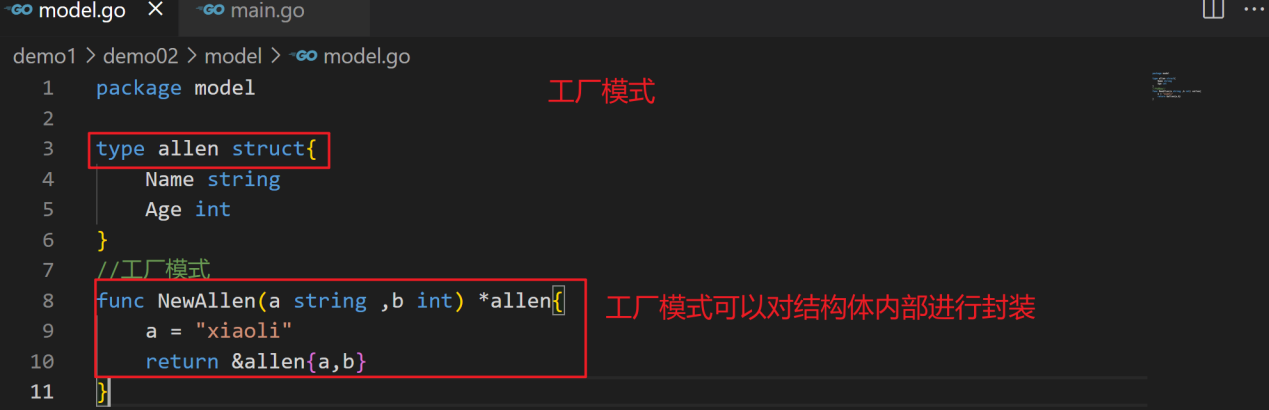


【3】返回结构体对应的指针类型



工厂模式：

工厂模式主要用于对结构体进行操作，可以在不影响原本结构体的前提下，通过重新命名对本来的结构体进行调用





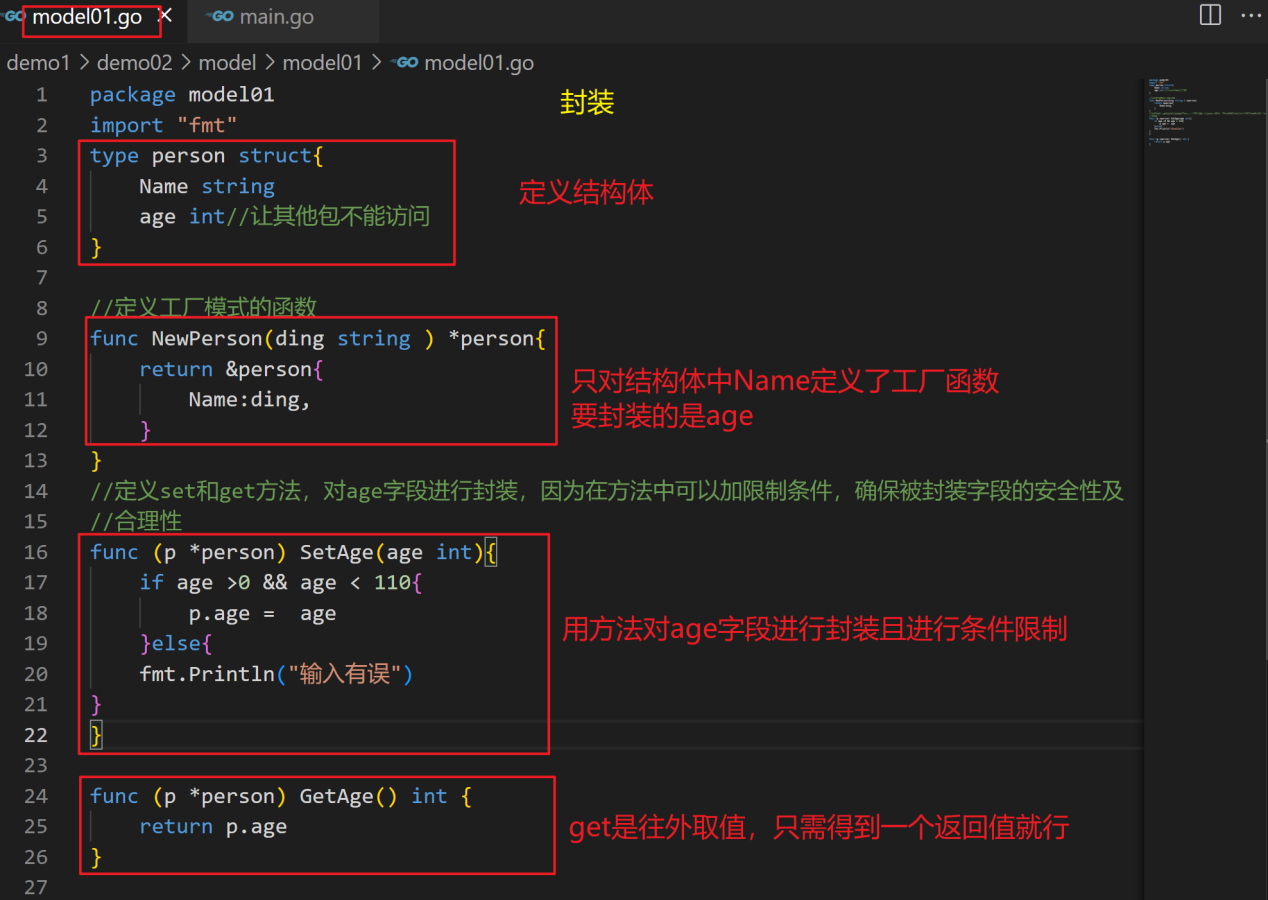
封装的实现：

【1】什么是封装：

封装就是把抽象的字段和对字段的操作封装在一起，数据被保护在内部，程序的其它包只能在通过被授权的操作方法，才能对字段进行操作

【2】封装的好处：

1. 隐藏实现细节
2. 可以对数据进行验证，保证安全合理





继承的引入：

【1】继承

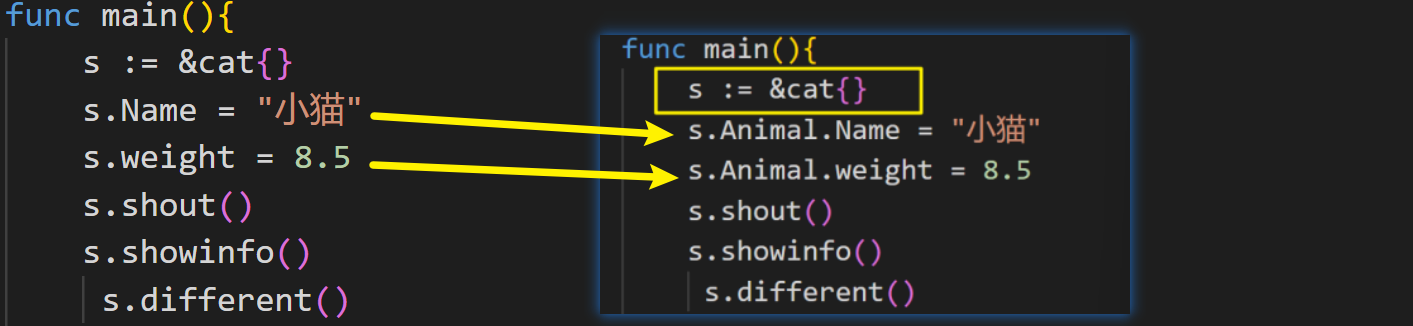
当多个结构体存在相同的属性（字段）和方法时，可以从这些结构体中抽象出结构体，在该结构体中定义这些相同的属性和方法，其他的结构体不需要重新定义这些属性和方法，只需嵌套一个匿名结构体即可。也就是说：在golang中，如果一个struct嵌套了另一个匿名结构体，那么这个结构体可以直接访问匿名结构体的字段和方法，从而实现了继承特性。



继承的注意事项

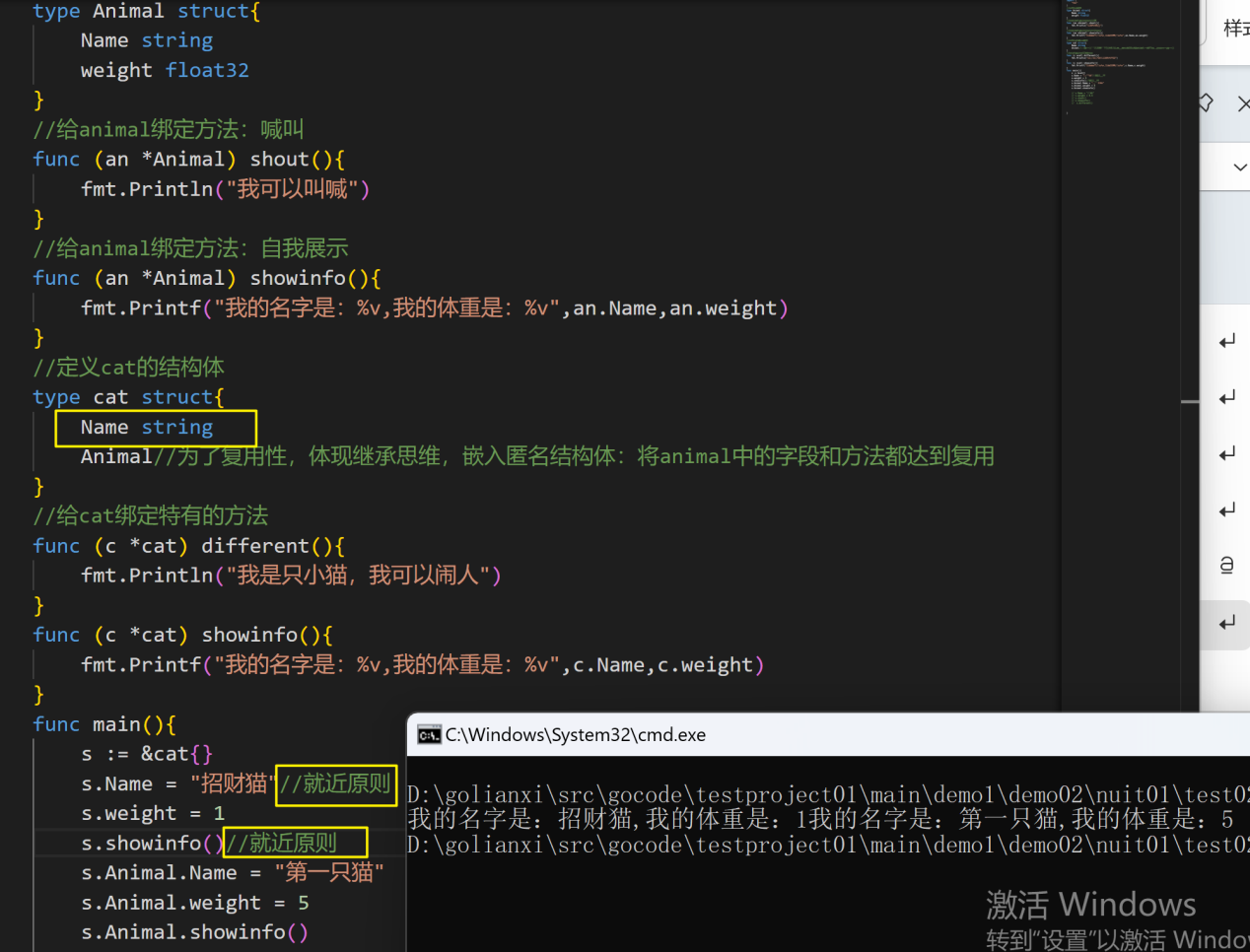
【1】结构体可以使用嵌套匿名结构体所有的字段和方法，即：首字母大写或者小写的字段，方法，都可以使用

【2】匿名结构体字段访问可以简化



s.Name--->（s:=&cat）s对应的结构体中找是否有Name字段，如果有直接使用，如果没有就去嵌入的结构体类型中的Name。

【3】当结构体和匿名结构体有相同的字段或者方法时，编译器采用就近访问原则，如希望访问匿名结构体的字段和方法，可以通过匿名结构体来区分

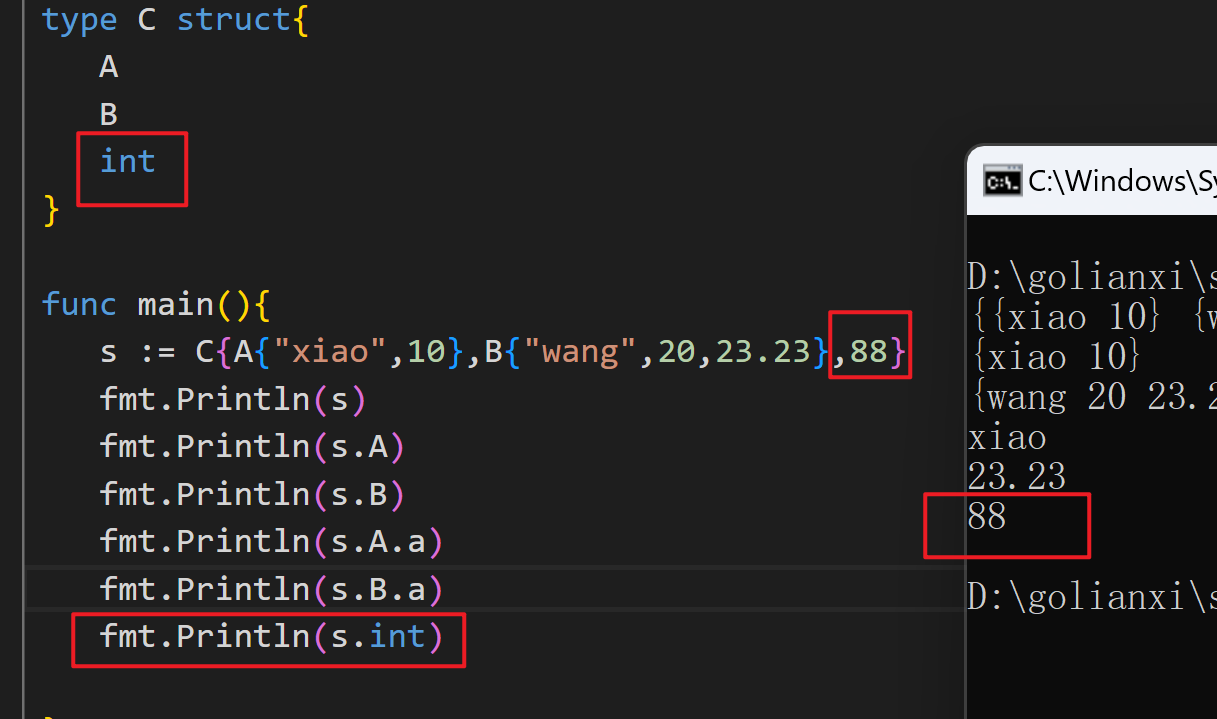


【4】golang中支持多继承，如一个结构体嵌套了多个匿名结构体，那么该结构体可以直接访问嵌套的匿名结构体的字段和方法，从而实现了多重继承，为了保证代码的简洁性，不建议使用多重继承，很多语言已将多重继承去除了，但是go中仍有保留。

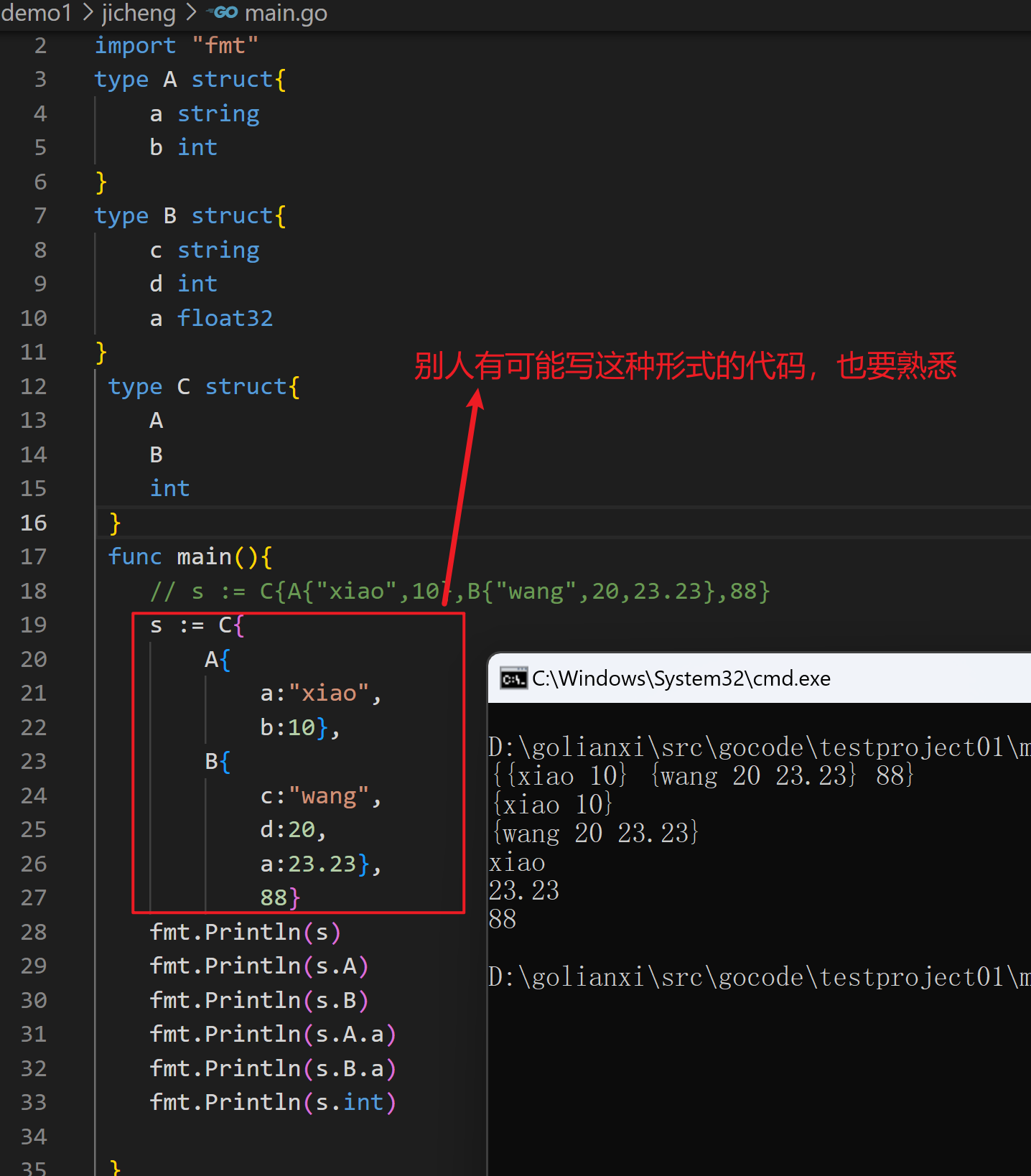


【5】如嵌入的匿名结构体有相同的字段名或者方法名，则在访问时，需要通过匿名结构体类型名来区分

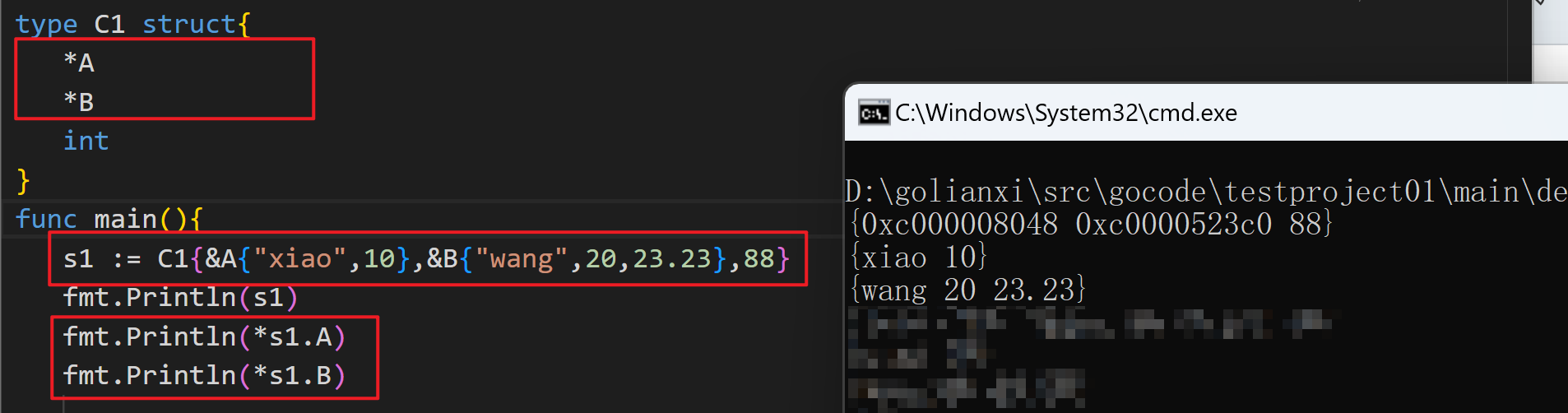
【6】结构体的匿名字段可以是基本数据类型，如



【7】嵌套匿名结构体后，也可以在创建结构体变量（实例）时，直接指定各个匿名结构体字段的值



【8】嵌入匿名结构体的指针也是可以的



【9】结构体的字段可以是结构体类型的。（组合模式）

