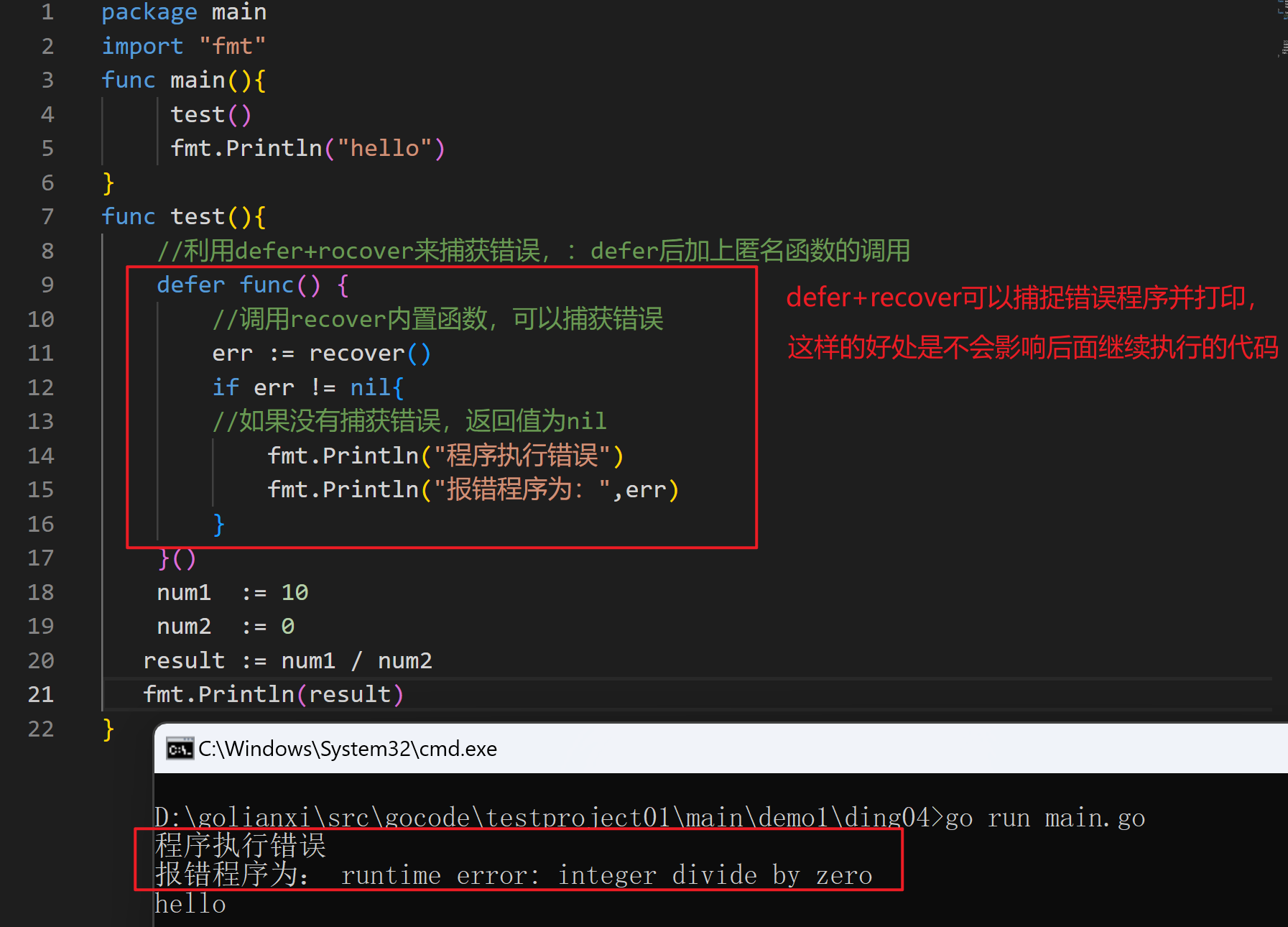
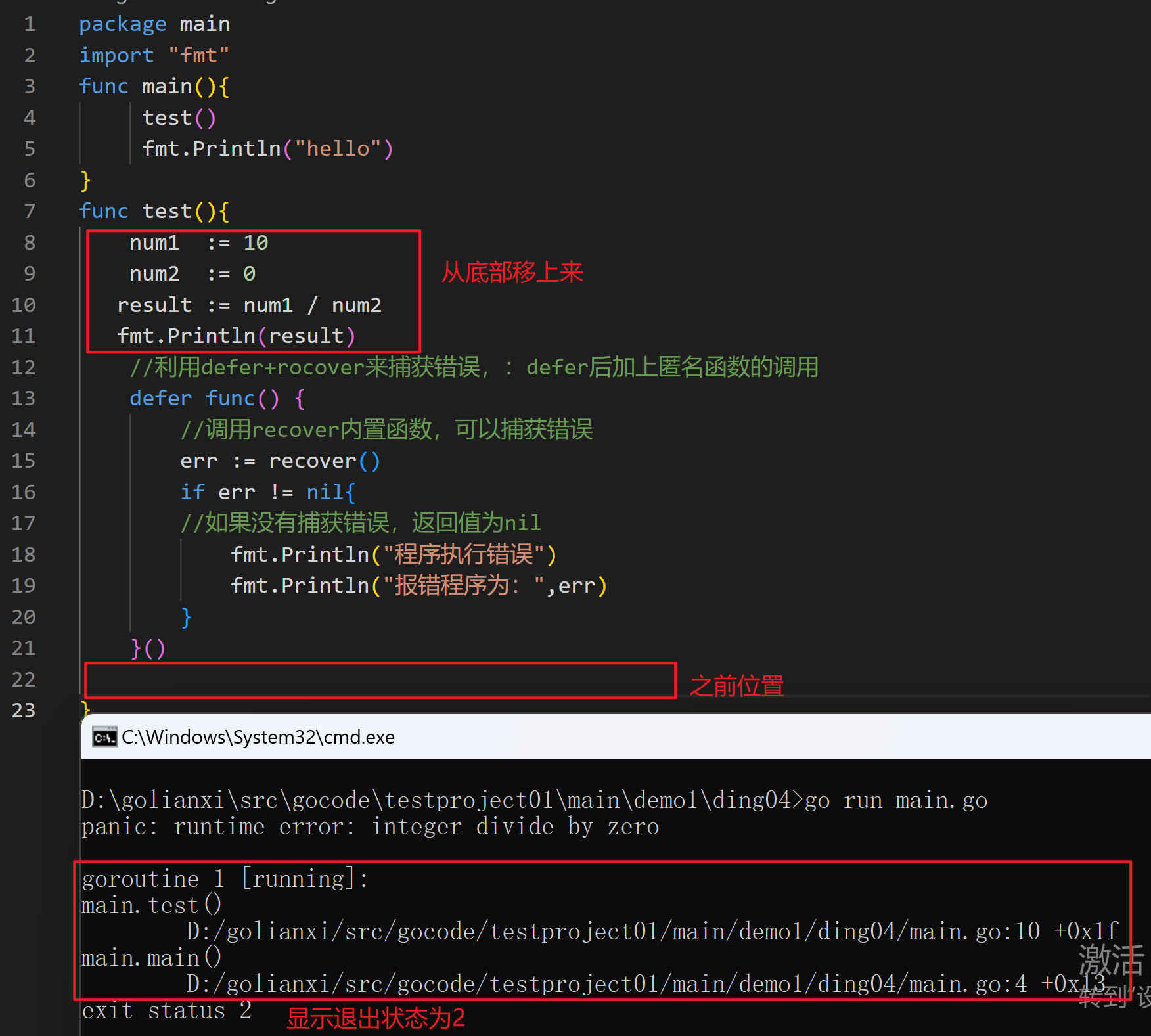
第二章

【1】错误处理捕获及打印

功能：不会影响后面代码的执行，而是将有问题的代码捕获并打印给程序员，提高代码健壮性

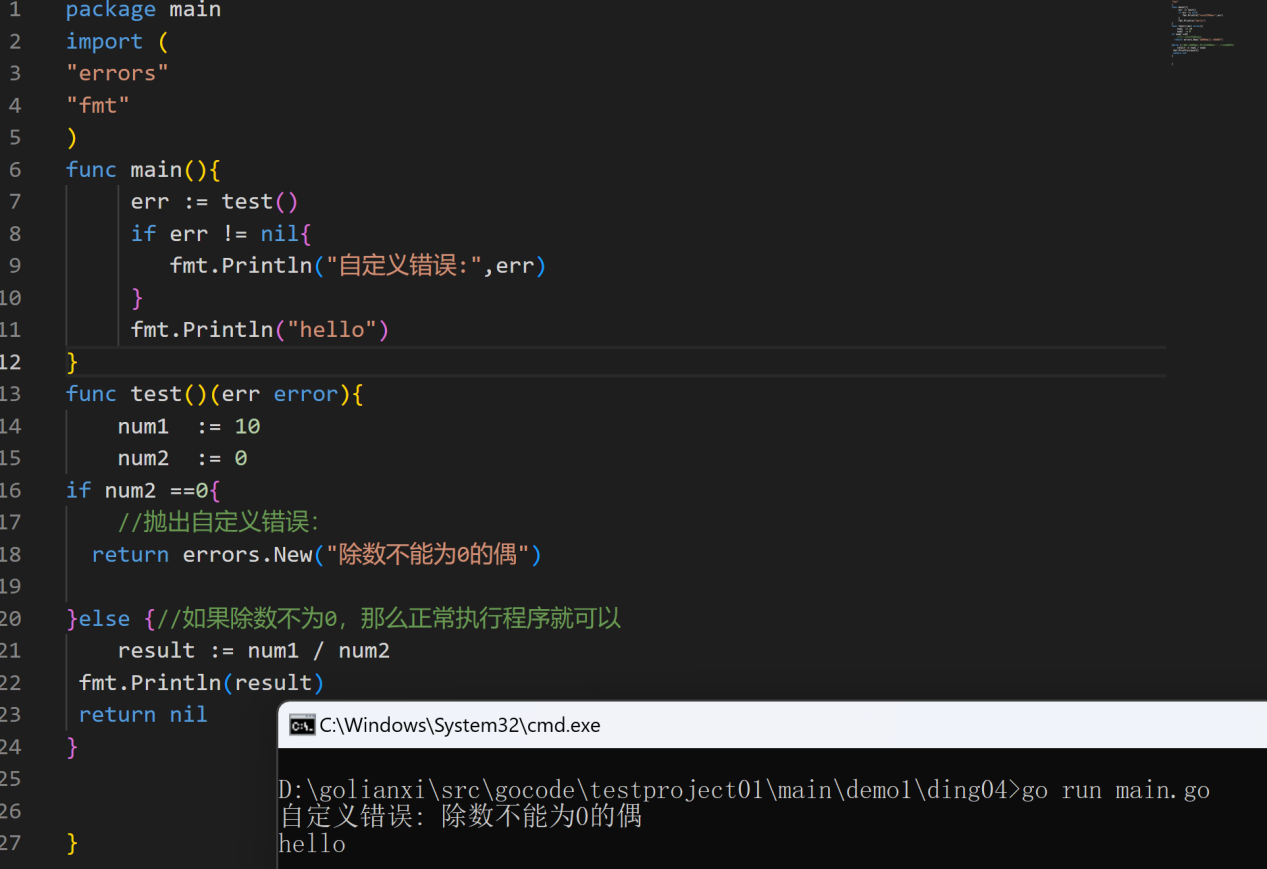


问题：



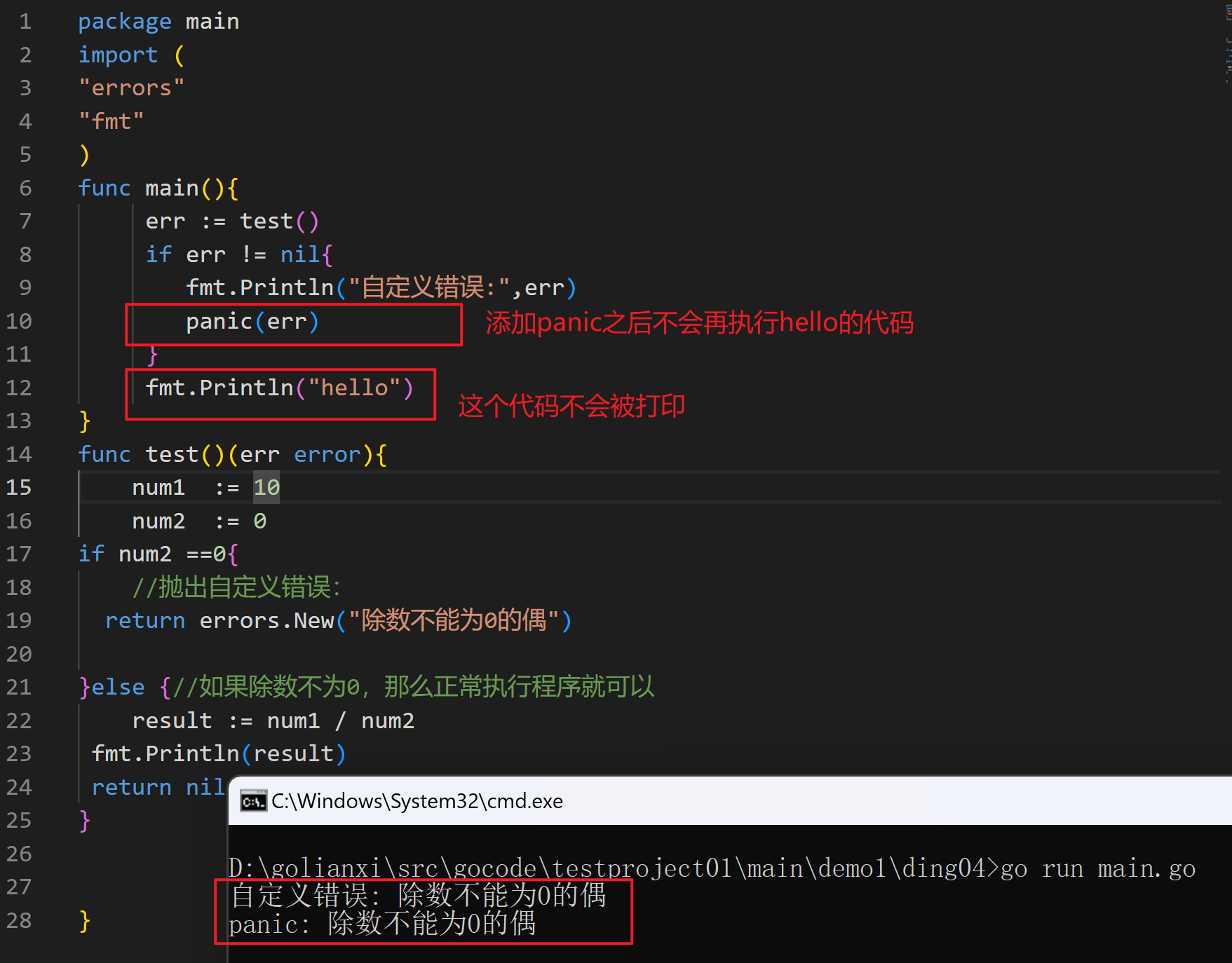
不知道什么原因导致

自定义错误



自定义错误有种情况：程序出现错误以后，后续代码没有必要继续执行，想让程序中断，退出程序

方法：可以借用builtin包下的的内置函数：panic



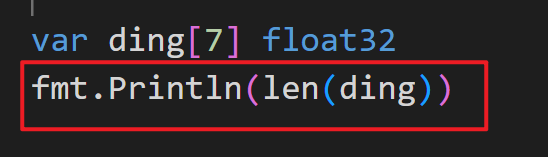
数组

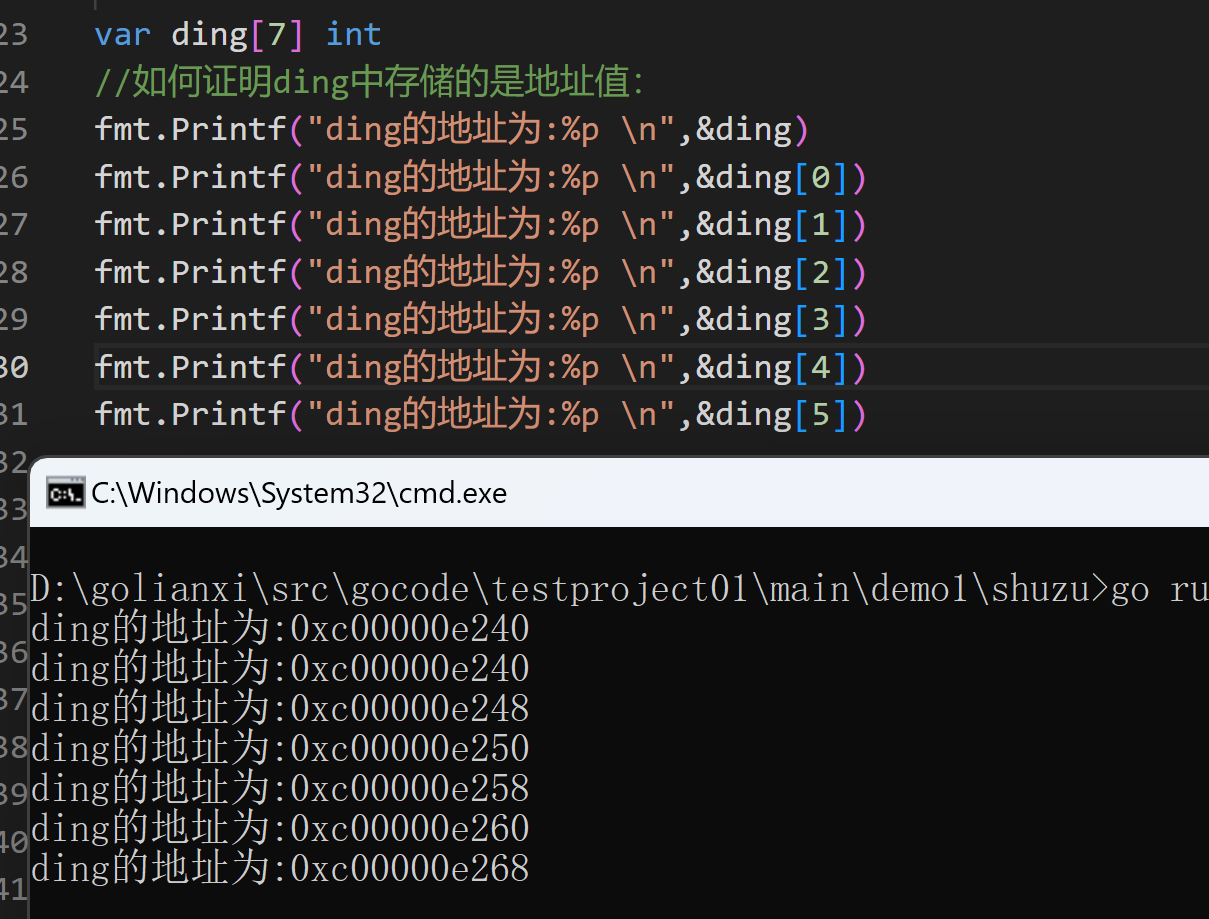
使用场景：

当变量定义个数太多，采用平时使用变量数量简单相加减的方式，导致代码数量多，人工输入费时费力，所以采用数组方式，可以存储相同类型的数据，简便高效

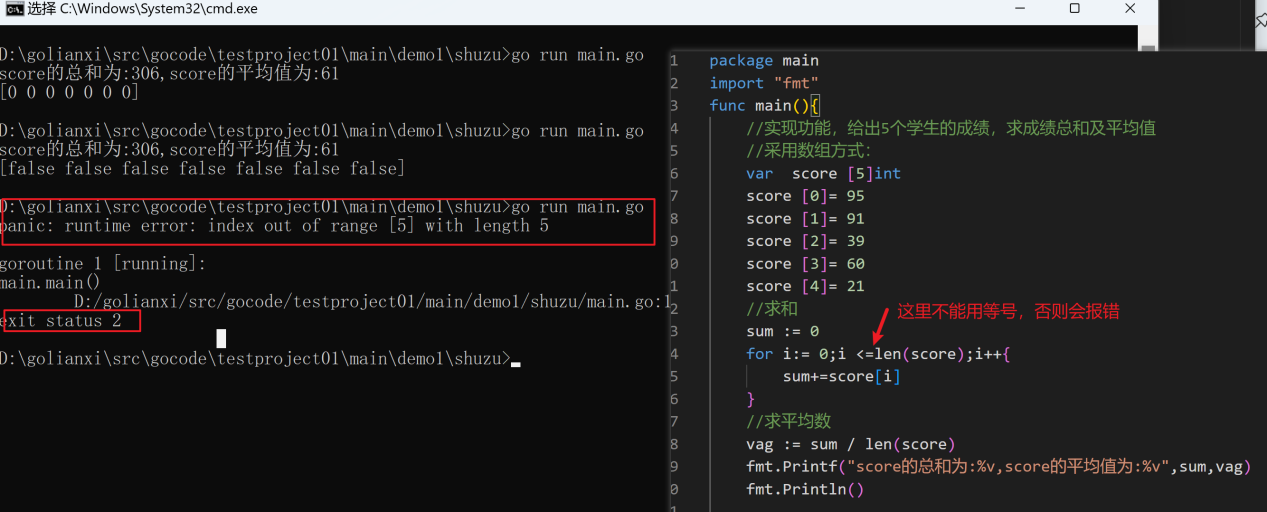


如果想看这个数组的长度及地址是多少，可以采用



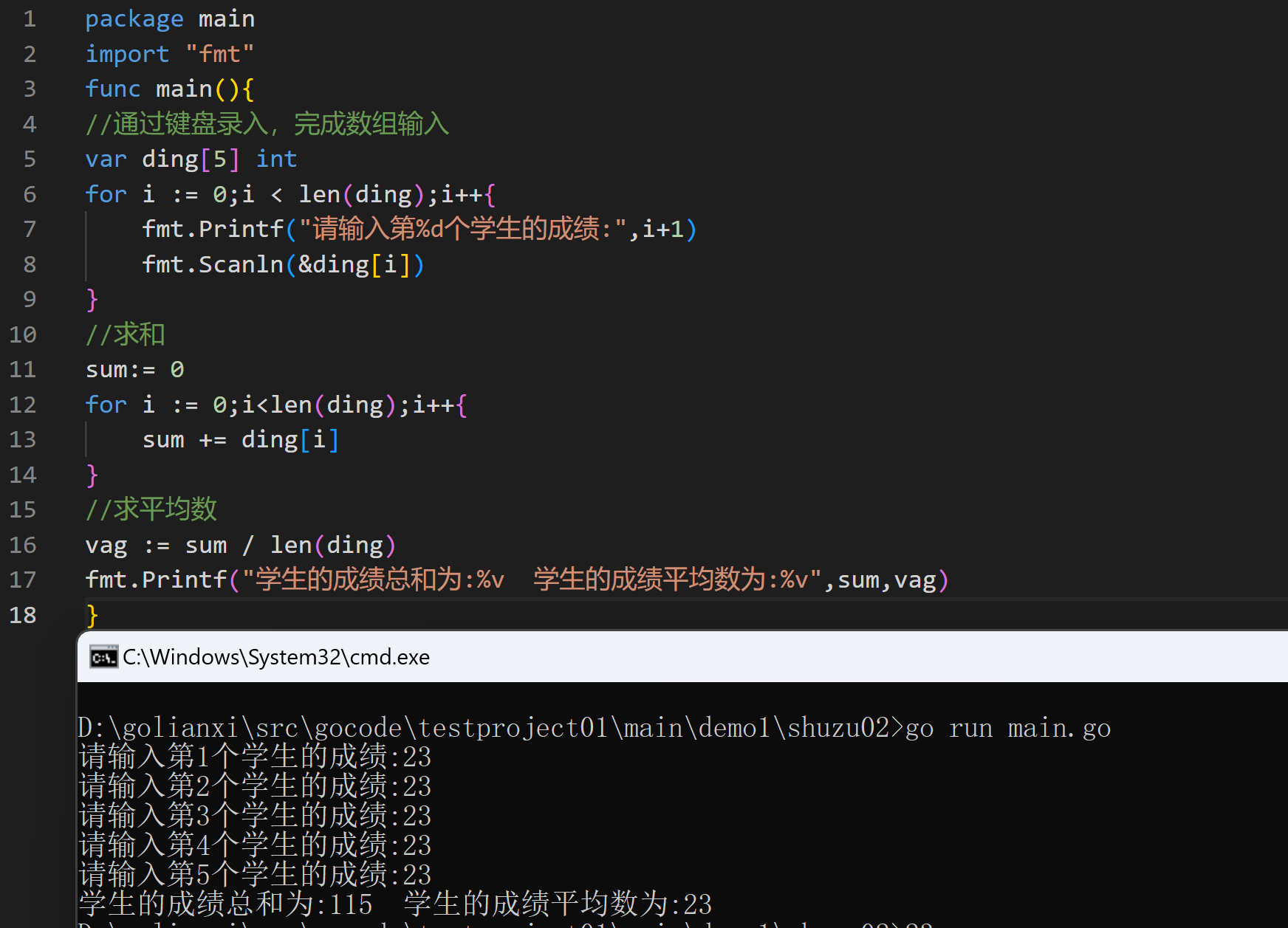


注意点：



For循环里面的等号要注意

通过for循环+终端输入的方式，实现代码如下



数组的遍历

【普通for循环】

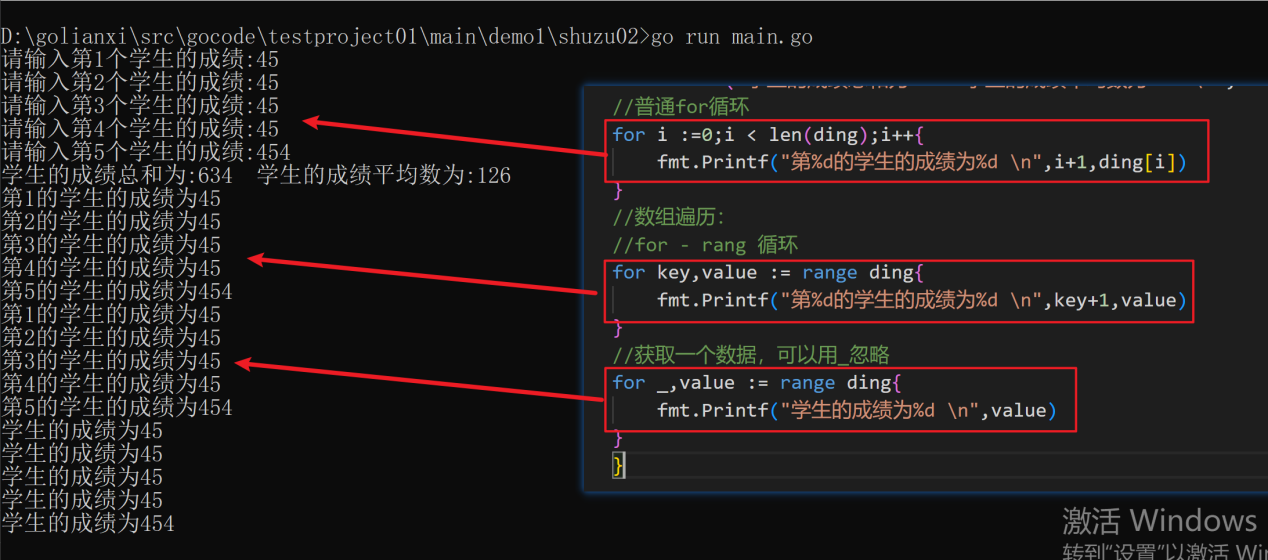
【键值循环】

例：for key,value := rang coll{

}

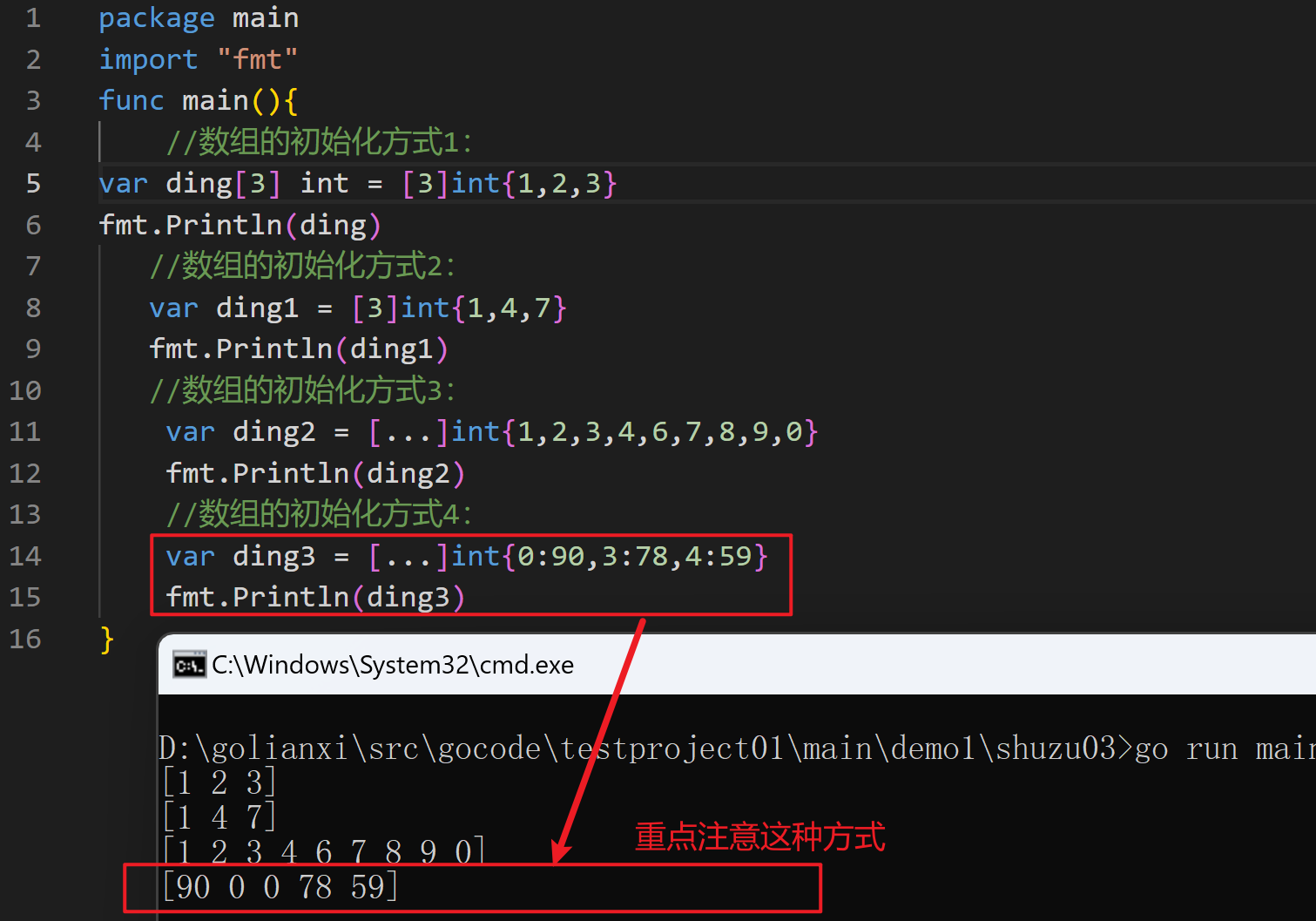
注意：

1. coll就是你要的数组
2. 每次遍历得到的索引用key接收，每次遍历得到的索引位置上的值用value接收
3. Key，value的名字可以随便起，自定义
4. Key，value属于在这个循环中的局部变量



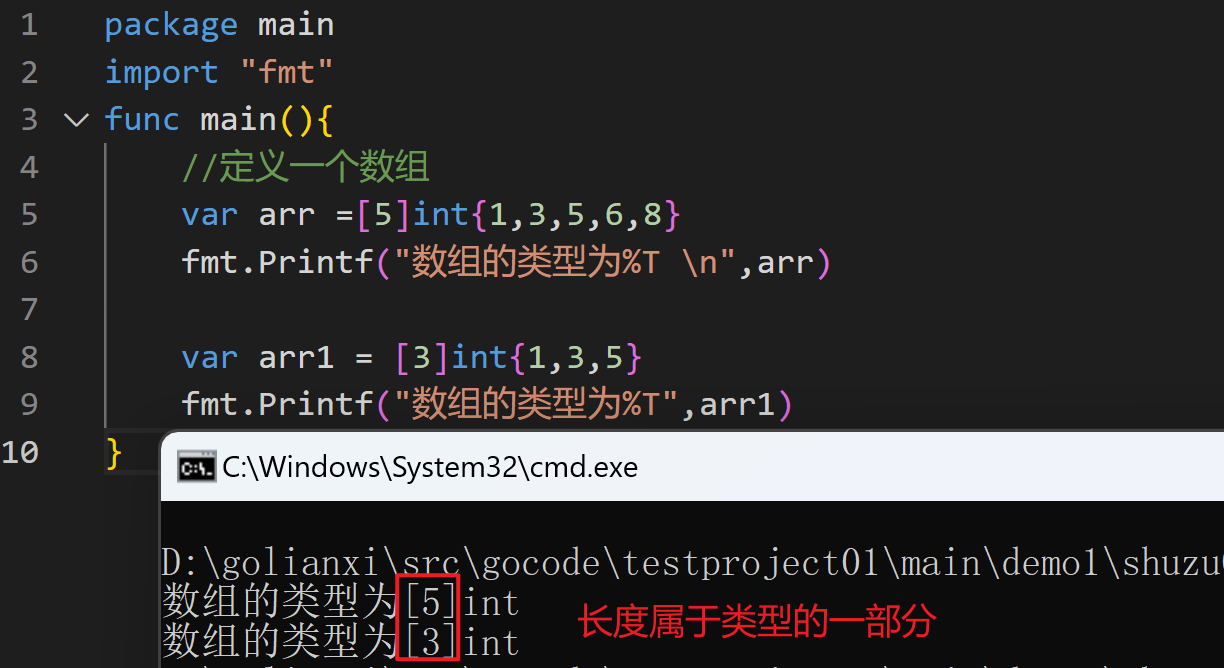
数组的初始化方式

数组的初始化有四种方式：



数组的注意事项：

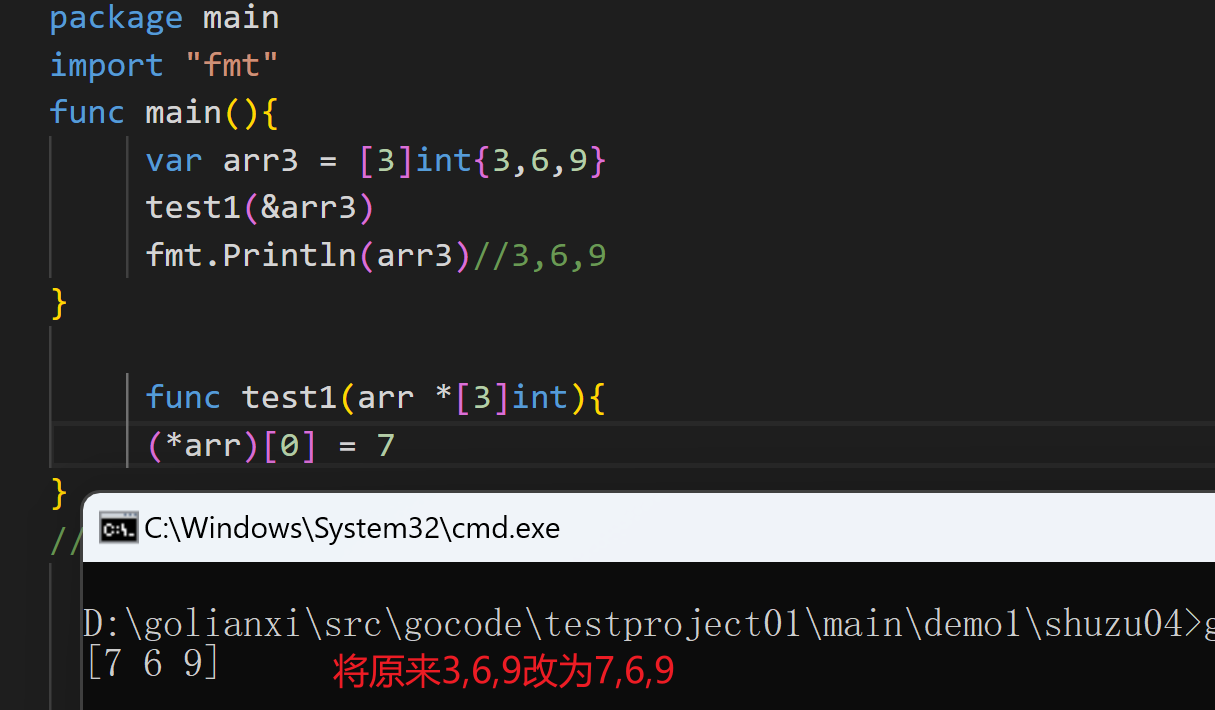
1. 长度属于类型的一部分



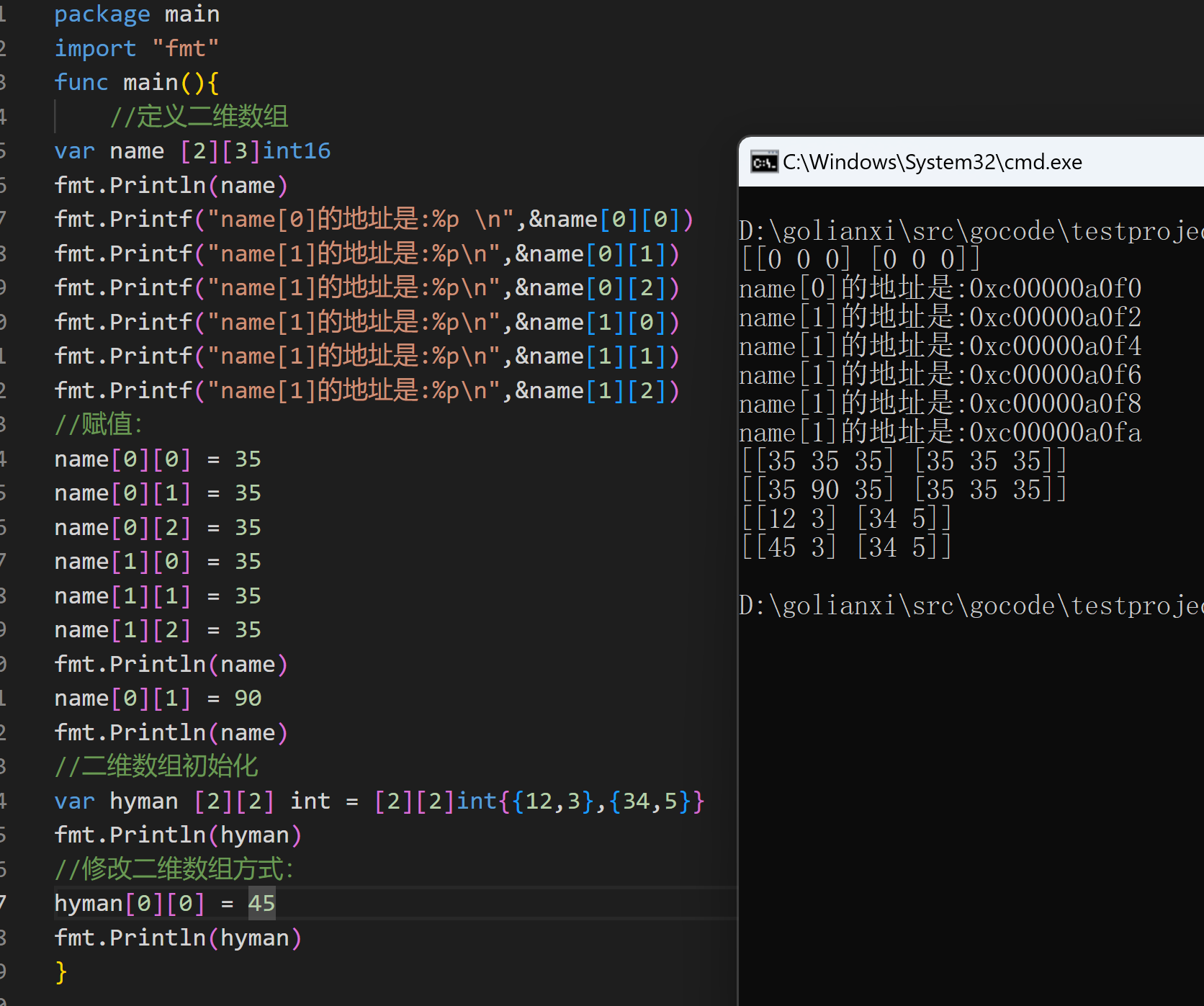
1. go中数组属于值类型，在默认情况下是值传递，因此会进行值拷贝



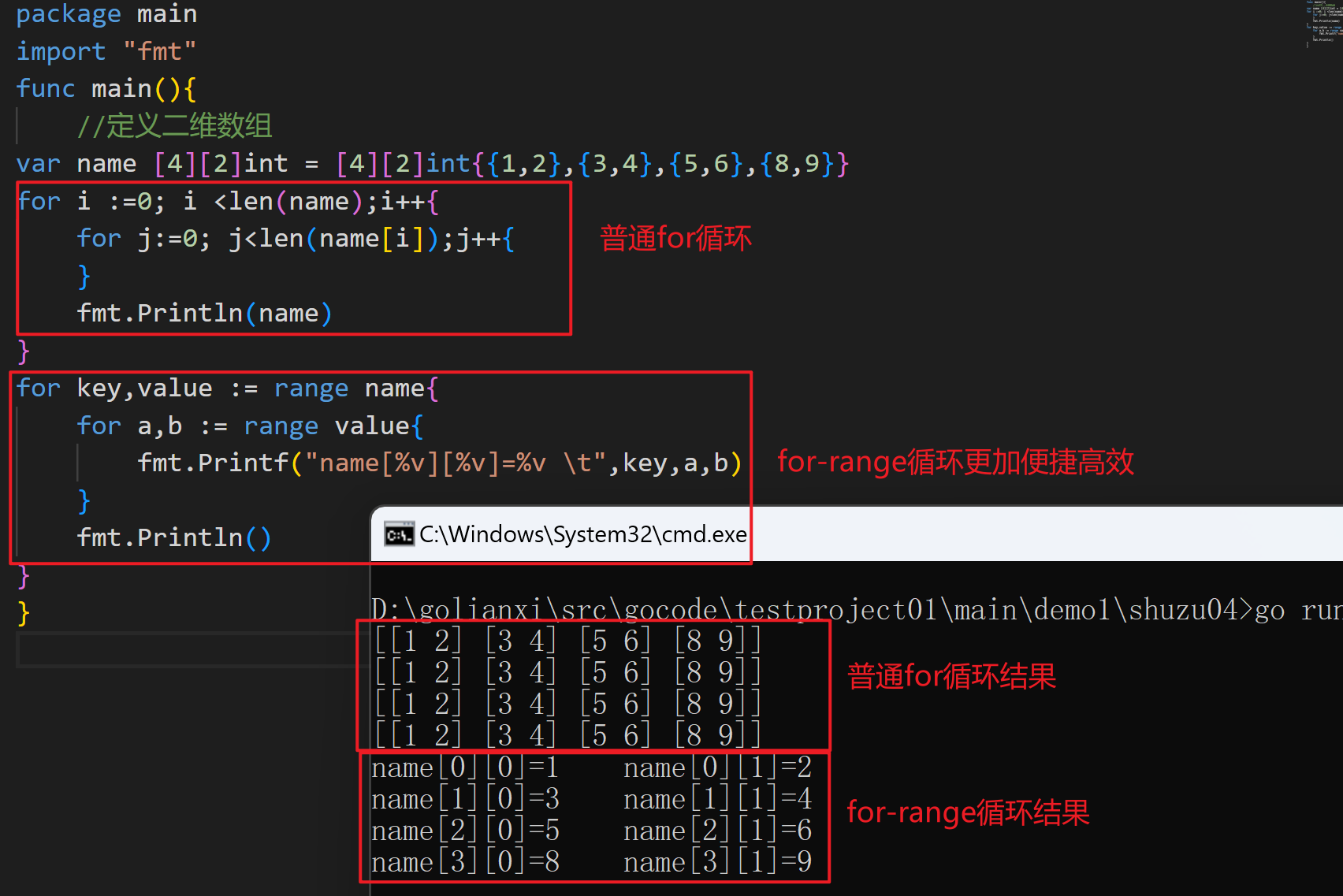
1. 如想在其他函数中，去修改原来的数组，可以使用引用传递（指针方式）



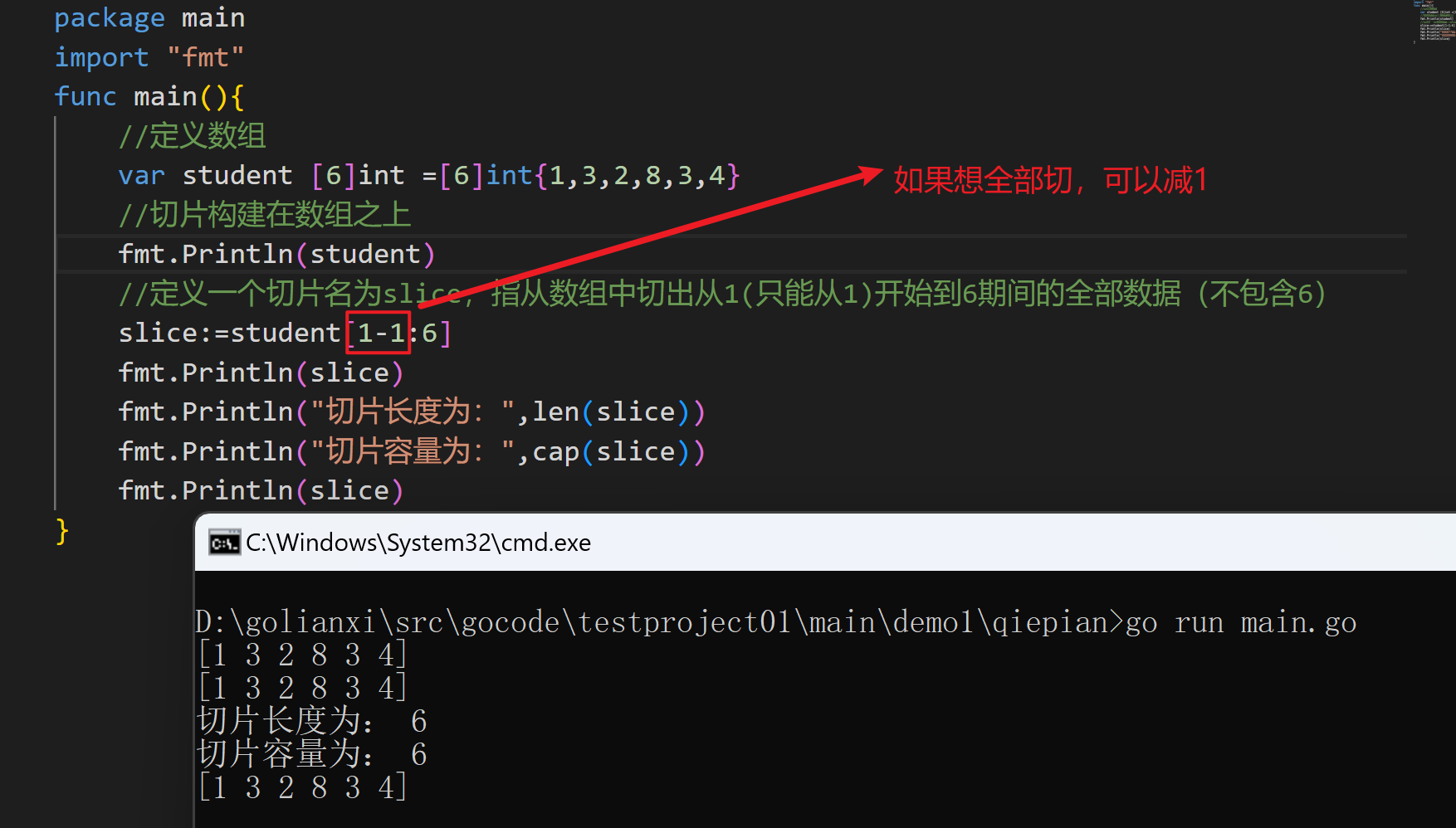
二维数组基本概况



二维数组的遍历

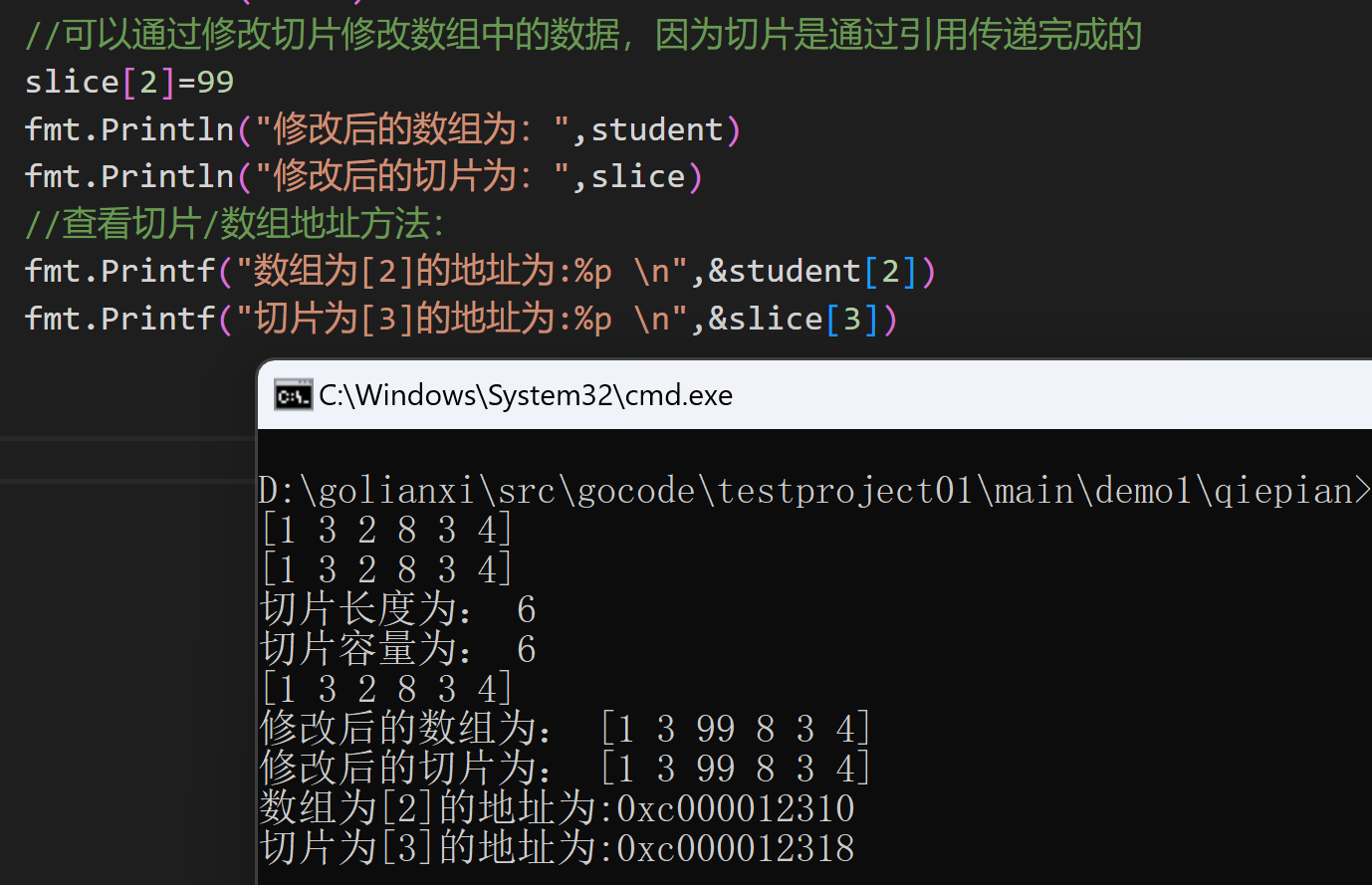


切片

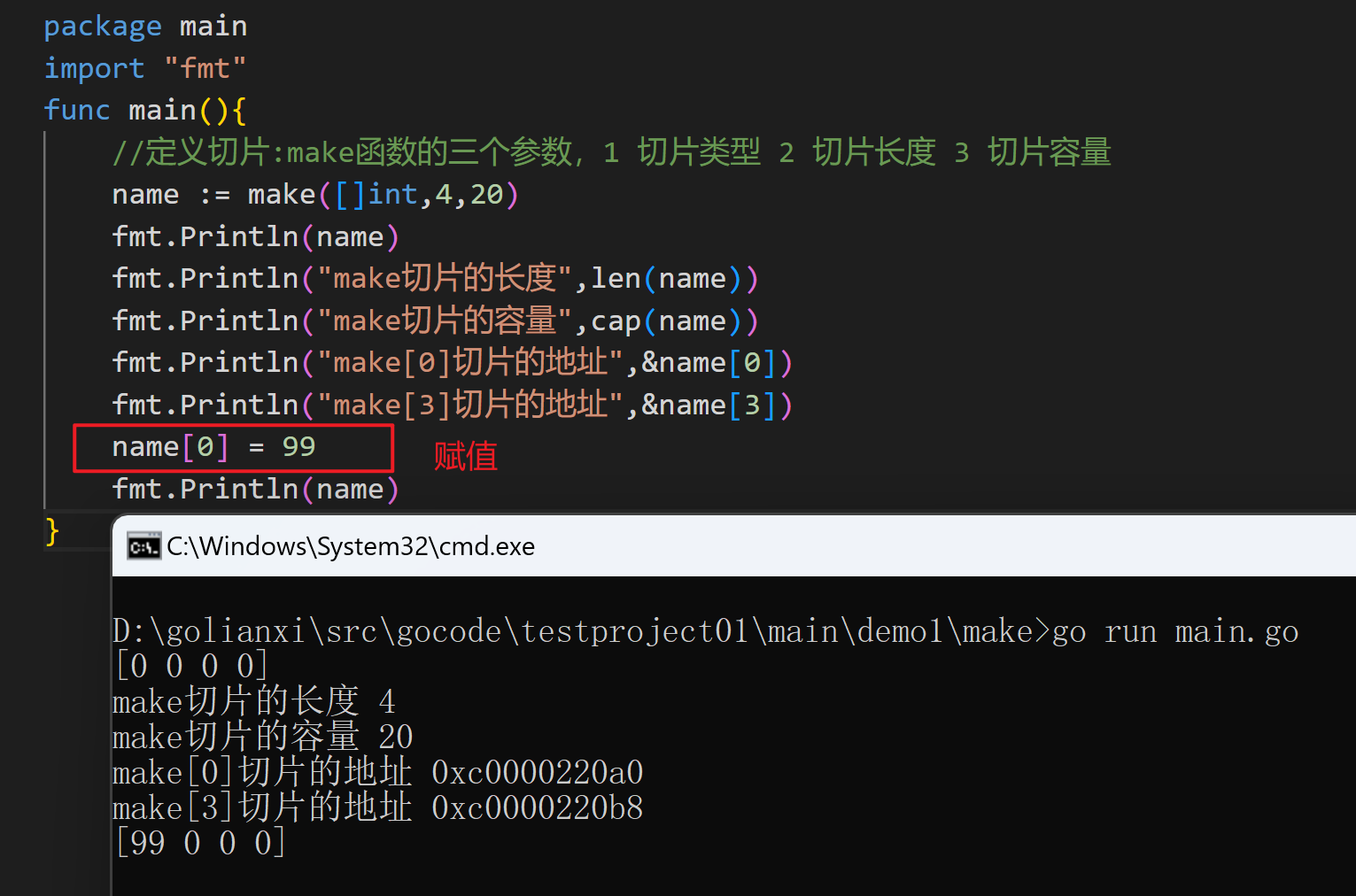


通过切片修改数组中的数据

代码如下：

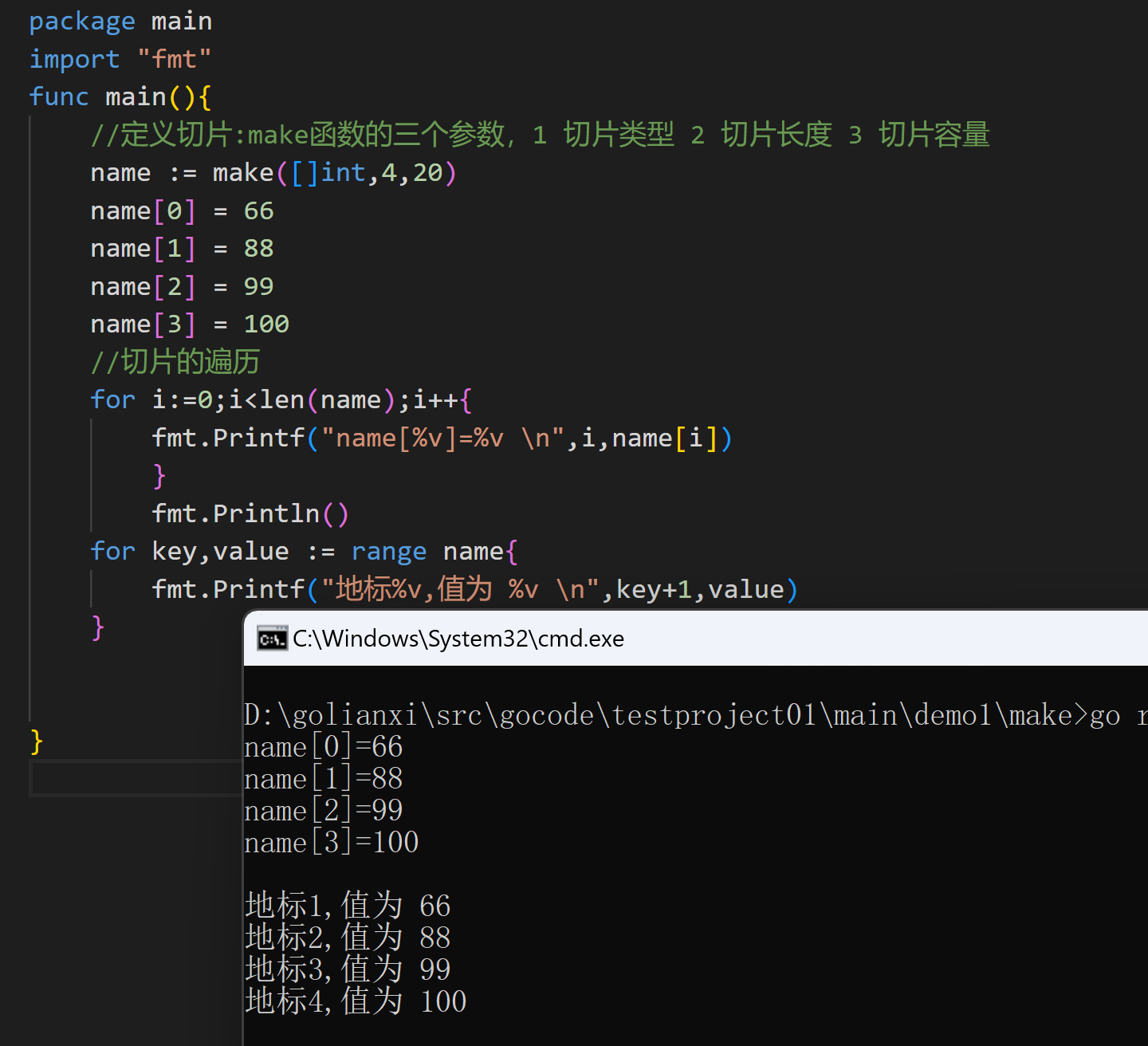


通过make内置函数创建切片。

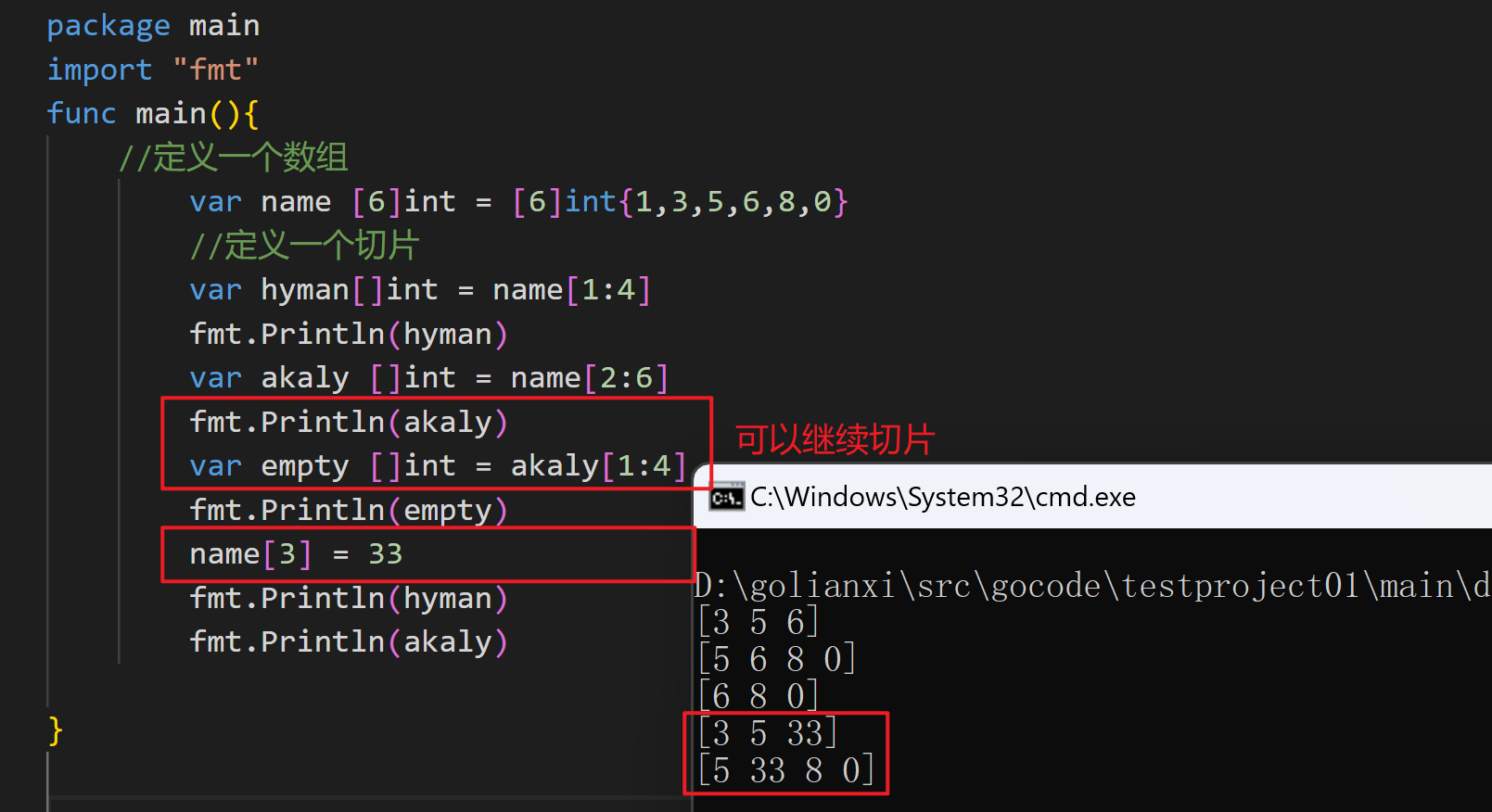


Ps：make底层创建一个数组，对外不可见，所以不可以直接操作这个数组，要通过slice去间接访问各个元素，不可以直接对数组进行维护/操作

切片遍历方式



切片的注意事项



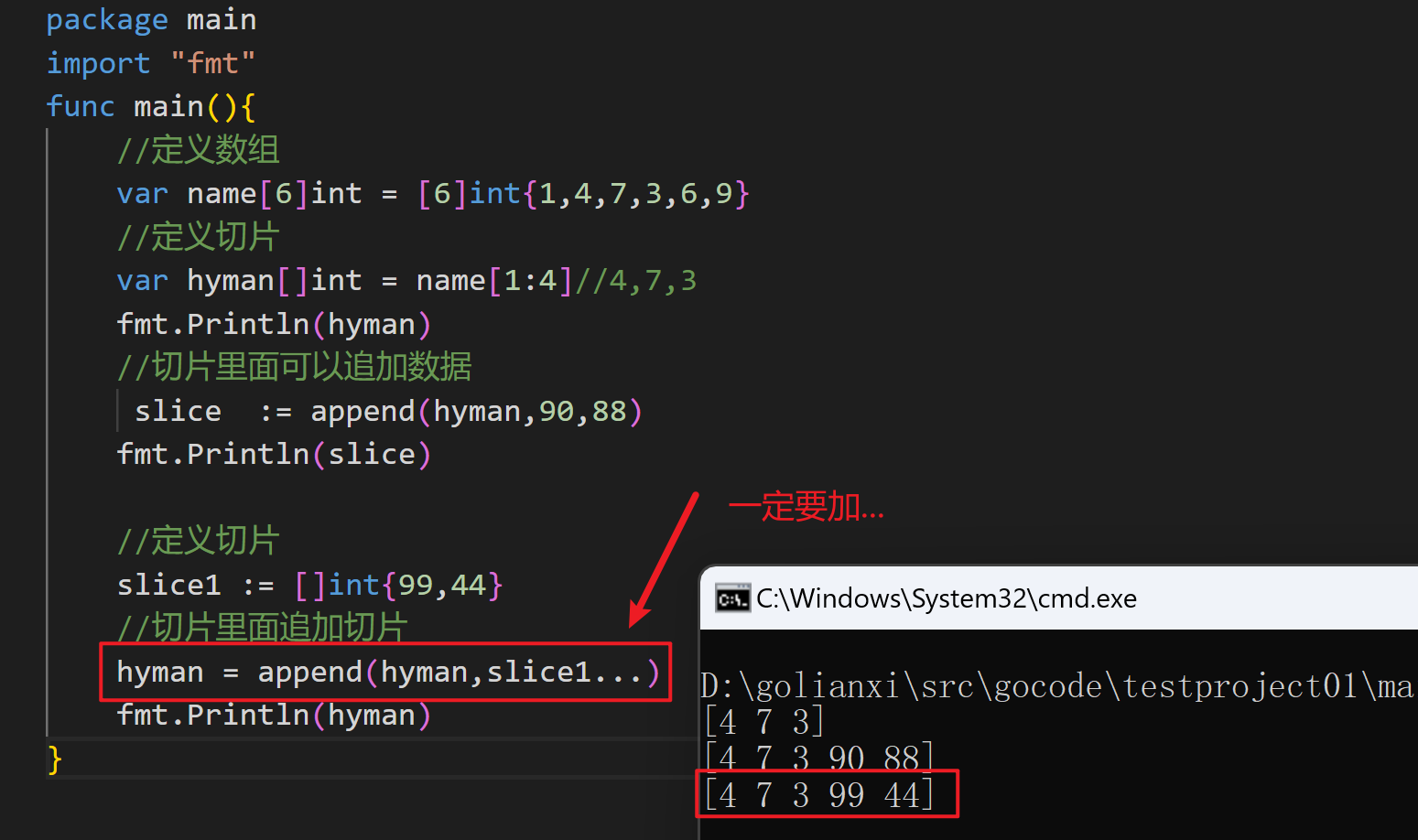
切片里面追加数据的方法：用append



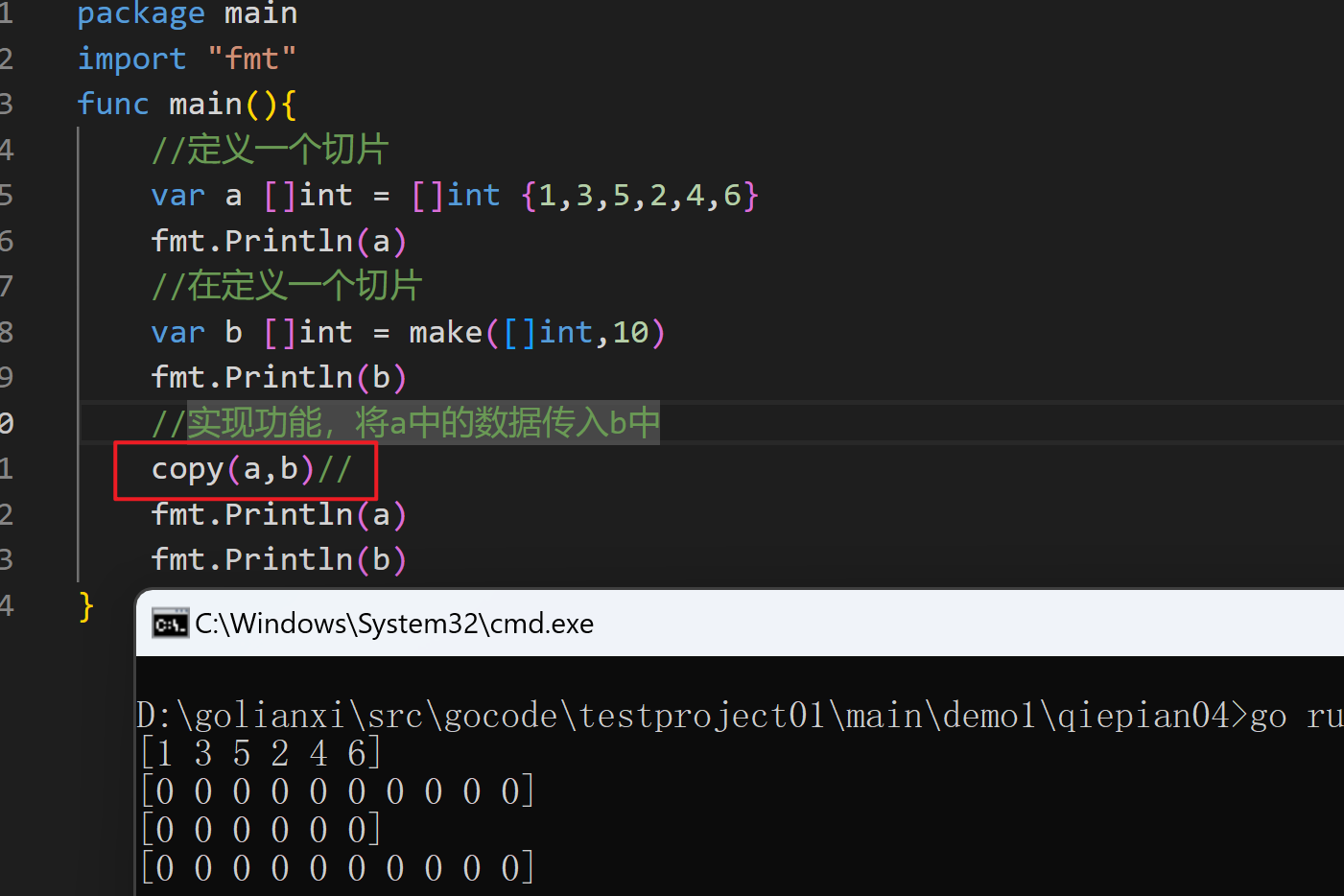
Append底层原理：

1. 底层追加元素的时候对切片进行扩容，老切片扩容为新切片
2. 创建一个新切片，将老切片中的4,7,3复制到新切片中，在新切片中追加90,88
3. Slice底层数组的指向，指向的是新数组
4. 在使用追加的时候，其实想要做的效果是给name追加
5. 底层的新数组还是不能直接维护，需要通过切片间接维护操作

切片后面可以追加切片



切片中copy的用法：



映射（Map）的引用

【1】映射（map），go语言中内置的一种类型，它将键值对相关联，可以通过键（key）来获取对应的值（value），如学生的学号与姓名，学号为键（key），姓名为值（value），map的作用是可以通过键（key）将整个映射的数据查找出来，类似其他语言的集合

【2】基本语法

var map 变量名 map[key type] value type

Ps: key,value的类型：bool，数字，string，指针，channel，还可以是只包含前面几个类型的接口，结构体，数组

Ps： key通常为int，string类型，value通常为数字（整数，浮点数），string，map，结构体

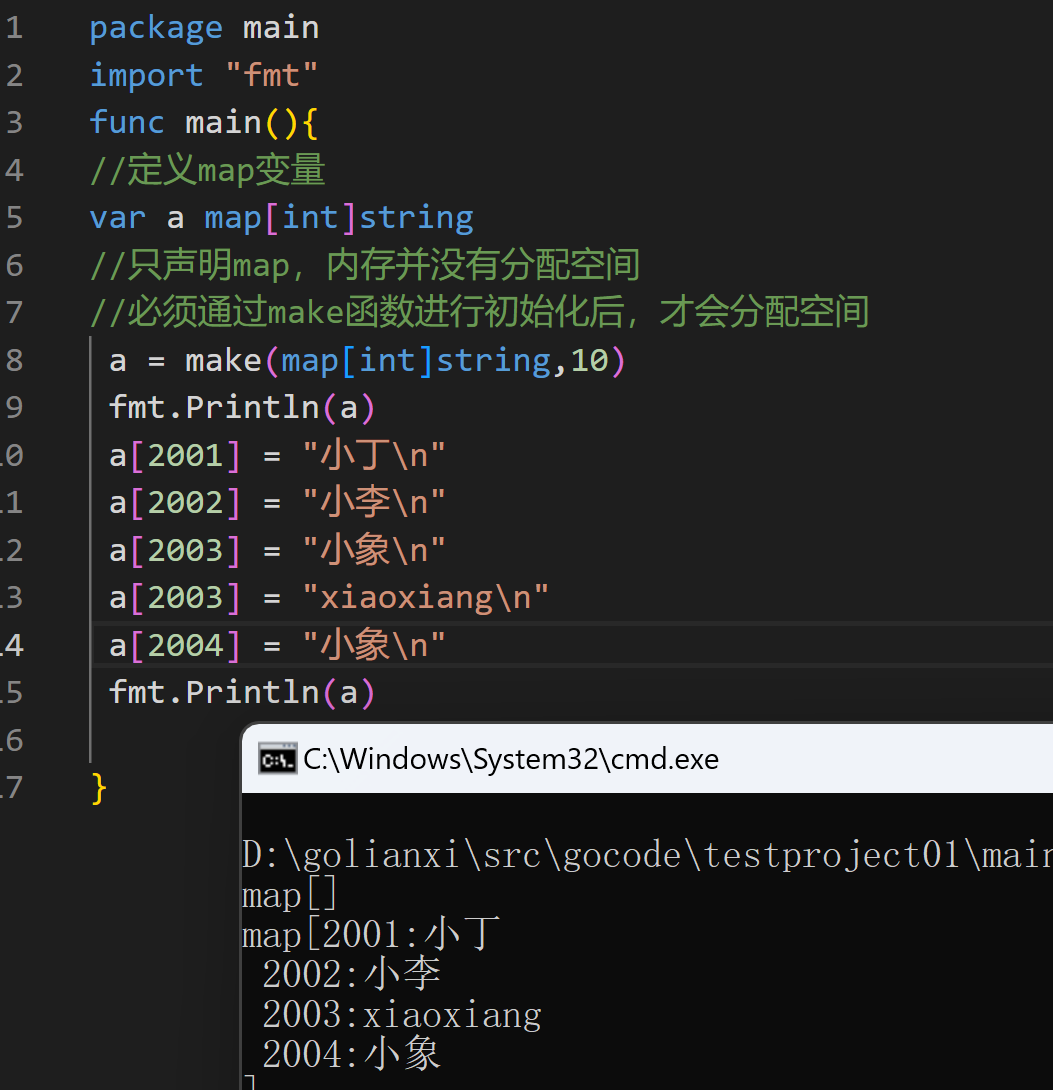
Ps：key部分不可以是slice，map，function

【3】代码：

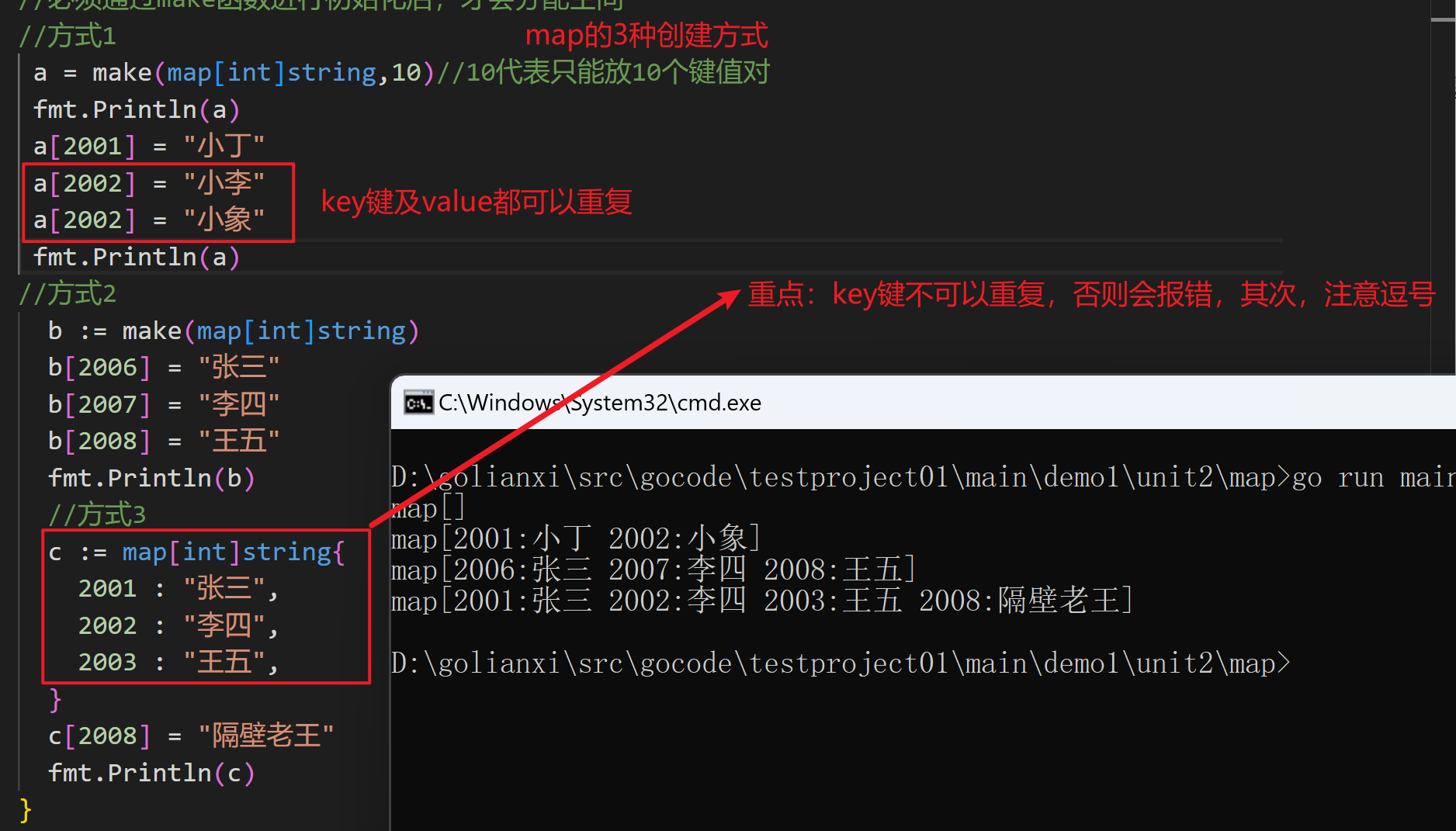
map的特点：

1. map集合在使用前一定要make
2. Map的key和value是无序的
3. Key如果有重复，那么value部分会是后一个value替换前一个value
4. Key如果没有重复，会有两条相同的Value
5. Make函数的第二个参数size可以省略，默认就分配一个内存





【2】map函数的三种创建方式：



【3】Map的基本操作事项

1. 增加和修改操作

增加：如果没有key值，添加新的key值为增加

修改：如果有key值，增加新的相同key值，覆盖旧key值即可

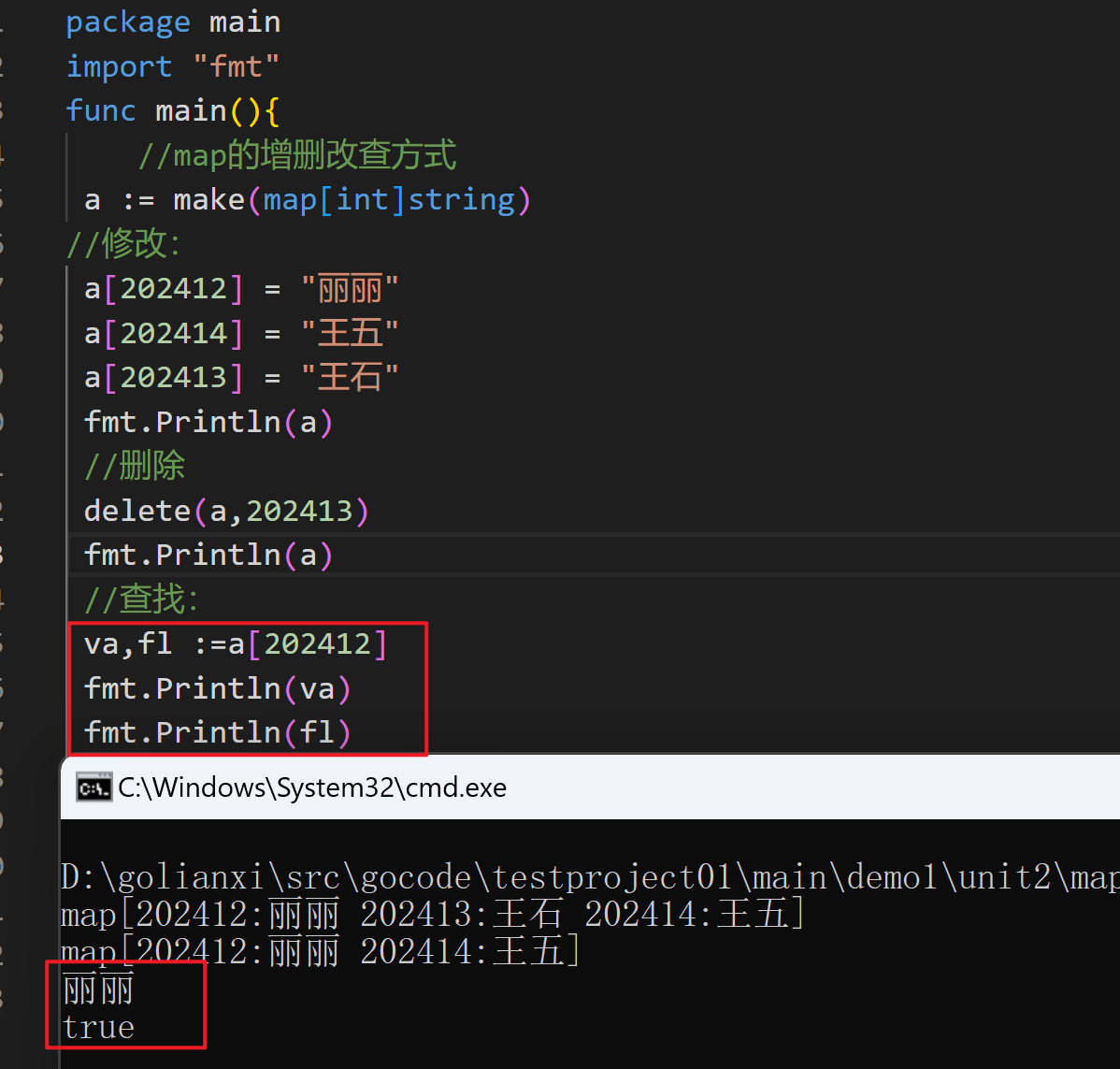
删除：delete是一个内置函数，如果key存在，就删除该key-value，如果key和value不存在，不操作，也不会报错

1. 清空操作
2. ,如果我们要删除map的所有key，没有专门的方法一次删除，需要通过遍历key的放式，逐个删除
3. ,或者map=make一个新的，让原来的称为垃圾，被bc回收
4. 查找操作：

Value， bool= map[key]

Value为返回的value，bool为是否返回，要么true，要么false

代码如下：



Map的遍历：

Map的遍历只能通过for-range实现

