

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 8. go中常见的错误

- 全部开发者教程
1. go最常用的设计模式 - 函数选项
2. 单例模式和懒加载
3. 测试金字塔
- 第23周 protoc插件开发、cobra命令行
1. protoc调试源码
2. protoc自定义gin插件
- 第24周 log日志包设计
- 日志源码
- 第25周 ast代码生成工具开发
- 错误码
- 第26周 三层代码结构
- 通用app项目启动



bobby · 更新于 2022-11-03

上一节 7. go代码的检测... 1. go最常用的设... 下一节

<https://github.com/teivah/100-go-mistakes#table-of-contents>

nil Map

map记得要make初始化， slice可以不用初始化！

<> 代码块

```
1 func main() {
2     var course map[string]string
3     course["name"] = "go体系课"
4 }
```

结构体空指针

空结构体和结构体空指针可不一样

<> 代码块

```
1 type Course struct {
2     Name string
3     Desc string
4 }
5
6 func (c *Course) String() float64 {
7     return c.Name + c.Desc
8 }
9
10 func main() {
11     var c *Course
12     fmt.Println(c.String())
13 }
```

使用对循环迭代器变量的引用 - 大坑！

在 Go 中，循环迭代器变量是一个单一的变量，在每个循环迭代中取不同的值。这如果使用不当，可能会导致非预期的行为。

<> 代码块

```
1 func main() {
2     var out []*int
3     for i := 0; i < 3; i++ {
4         out = append(out, &i)
5     }
6     fmt.Println("Values:", *out[0], *out[1], *out[2])
7     fmt.Println("Addresses:", out[0], out[1], out[2])
8 }
```

意见反馈

收藏教程

标记书签

解决办法

<> 代码块

```
1 func main() {
2     var out []*int
3     for i := 0; i < 3; i++ {
4         tmpi := i
5         out = append(out, &tmpi)
6     }
7     fmt.Println("Values:", *out[0], *out[1], *out[2])
8     fmt.Println("Addresses:", out[0], out[1], out[2])
9 }
```

原因是：在每次迭代中，我们将 i 的地址追加到 out 切片中，但由于它是同一个变量，我们实际上追加的是相同的地址，该地址最终包含分配给 i 的最后一个值。

所以只需要拷贝一份，让两者脱离关联就可以了。

同样的，如果这里是for循环然后启动多个goroutine，如下：

<> 代码块

```
1 package main
2
3 import (
4     "fmt"
5     "strconv"
6     "time"
7 )
8
9 func main() {
10     goodsID := []uint64{1, 2, 3, 4, 5}
11
12     for _, id := range goodsID {
13         go func() {
14             fmt.Println("正在查询商品: " + strconv.Itoa(int(id)))
15         }()
16     }
17
18     time.Sleep(time.Second * 5)
19 }
```

注意：这种bug在goland中一般会提醒，

常用的解决办法：

<> 代码块

```
1 package main
2
3 import (
4     "fmt"
5     "strconv"
6     "time"
7 )
8
9 func main() {
10     goodsID := []uint64{1, 2, 3, 4, 5}
11
12     for _, id := range goodsID {
13         go func(id uint64) {
14             fmt.Println("正在查询商品: " + strconv.Itoa(int(id)))
15         }(id)
16     }
17 }
```

[意见反馈](#)[收藏教程](#)[标记书签](#)

```
19      time.Sleep(time.Second * 5)
    }
```

7. go代码的检测工具 ◀ 上一节 下一节 ▶ 1. go最常用的设计模式 – 函数选项

 我要提出意见反馈

