

课程安排，请关注微信公众平台或者官方微博

编程语言：Golang 与 html5

编程工具：Goland 和 HBuilder

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔。

大家好，

欢迎来到 字节教育 课程的学习

字节教育官网：www.ByteEdu.Com

腾讯课堂地址：Gopher.ke.qq.Com

技术交流群：221 273 219

微信公众号：Golang 语言社区

微信服务号：Golang 技术社区

目录：

第一季 Golang 语言社区-设计模式.....	2
第六节 对象池模式.....	2
一、公众账号：.....	2
二、对象池模式讲解：.....	2

第一季 Golang 语言社区-设计模式

第六节 对象池模式

一、公众账号：



回复关键字：客服

获取课程助教的微信，申请加入课程微信群

二、对象池模式讲解：

Object Pool Pattern

对象池模式是一种创建型模式，根据需求来预测将要使用的对象，提前创建并保存在内存中。
实现对象池定义

```
1. package pool
```

```
2.
3.  import (
4.      "fmt"
5.      "strconv"
6.  )
7.
8.  type Object struct{
9.
10. }
11.
12. func (Object)Do(index int){
13.     fmt.Println("Object Do:"+strconv.Itoa(index))
14. }
15.
16.
17. type Pool chan *Object
18.
19. func NewPool(total int)*Pool{
20.     p := make(Pool,total)
21.     for i := 0;i<total;i++){
22.         p <- new(Object)
23.     }
24.     return &p
25. }
26.
```

复制代码

使用

```
1.  func main(){
2.
3.     p := pool.NewPool(5)
4.
5.     wait := sync.WaitGroup{}
6.
7.     for i:=0;i<100;i++ {
8.
9.         index := i
10.
11.         wait.Add(1)
```

```
7.         go func(pool pool.Pool, ind int) {
8.             select {
9.                 case Obj := <-pool:
10.                    Obj.Do(ind)
11.                    pool <- Obj
12.                 default:
13.                    fmt.Println("No Object:"+strconv.Itoa(ind))
14.            }
15.            wait.Done()
16.        }(*p, index)
17.    }
18.    wait.Wait()
19. }
20.
```

复制代码

这里使用 goroutines 来并发地读取 pool 中的对象。

注意

- 当创建对象的代价比维护代价更高的时候，使用对象池模式是极好的。
- 如果需求相对固定，那么维护对象的代价可能得不偿失
- 提前初始化对象对性能有积极的影响