# 介绍go语言

今天看了一下go 纯纯c+py+协程的组合产品 有各种各样语法糖

首先go是经典的var定义变量 可以带类型也可以不带 应该是自己判断

里面有c语言经典结构体struct 还有接口interface 指针

结构体可以去继承接口，接口之间也是可以互相组合的 这个里面的所有类型的超类显示是Interface{}

里面的make关键字就是c语言的malloc

Go语言的并发是比较重要的

使用goroutines channel和调度器实现

goroutines就相当于是携程

Channel 用于作为携程之间传输数据的通道 可以直接传进你的协程里面作为参数

Channel的大小和是否缓冲 会改变生产者和消费者的阻塞动作表现

Range用到channel效果就很类似于从管道往出读数据

select 语句使得一个 goroutine 可以等待多个通信操作。select 会阻塞，直到其中的某个 case 可以继续执行

我觉得最大的特点是很多能力和编排，都通过关键字和传值、初始化来完成，不需要锁和繁杂的工具类来实现了，而且协程本来就是快。代码清晰，出问题很好排查。

说一下最重要的区别

线程上的处理 go很简单 java很繁琐很多类

Go可以自己控制内存那些更适合高性能 也有自动GC 鼓励使用官方库 java比较厚重第三方库众多

Go中没有重写 重载那些 最好每个方法都可以有不同的名字或者签名不同 java有重写重载

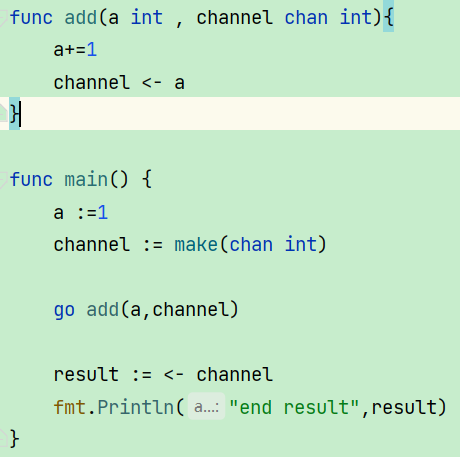
错误处理上go时根据函数返回错误码 必须定义才能返回 也可以使用Defer, Panic, and Recover完成各种错误处理 java则是异常

Go面向过程 java面向对象 面向过程编程更适合解决较为简单、线性的任务，而面向对象编程则更适用于复杂系统的设计，特别是那些需要模拟现实世界实体及其交互的应用。

Go也是根据机器编译 java是面向虚拟机

# 线程实践版本为1.24

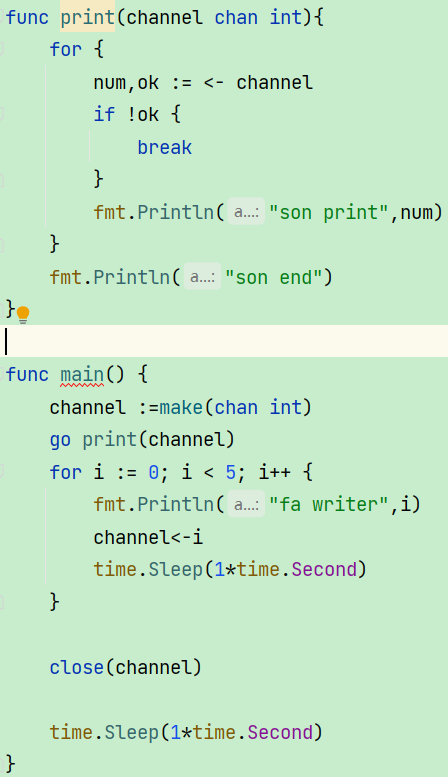
请用go语言，完成以下需求，主线程中定义一个变量A为1，开启子线程并把A传入，子线程做+1处理，再把变量返回回来，由主线程进行打印



Go的chan和在java中给线程函数中传递一个线程安全变量，是不是一样的效果呢？

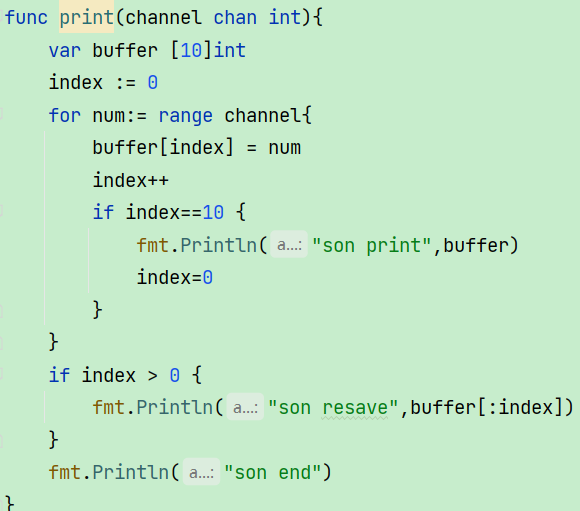
区别就在于我之前说的，go的线程同步就是简洁明了，不会有java中各种类的使用，也就是不会出现复杂问题，这是第一点。其次go遵循了“通过通信来共享内存”而不是“通过共享内存来通信”的设计哲学。

请用go语言，完成以下需求，主线程和子线程进行通信，主线程每写入一个数字，子线程就打印，主线程没有动作时，子线程就阻塞等待

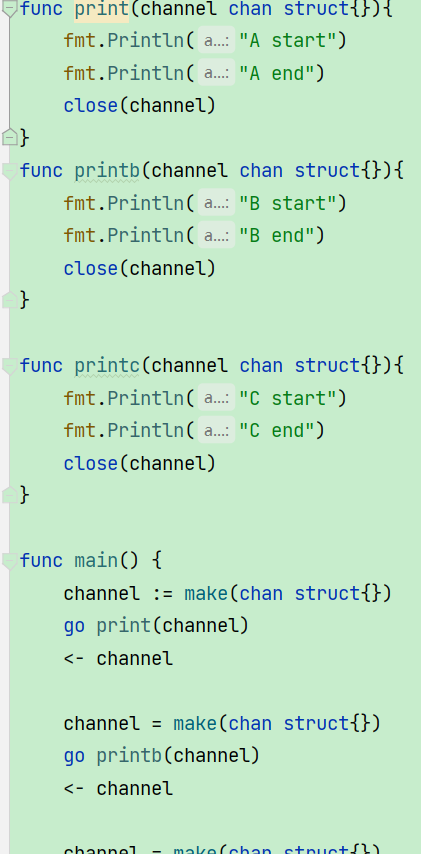


请用go语言，完成以下需求，主线程和子线程进行通信，主线程每写入10个数字，子线程就打印，没写够10个字时，子线程就阻塞等待

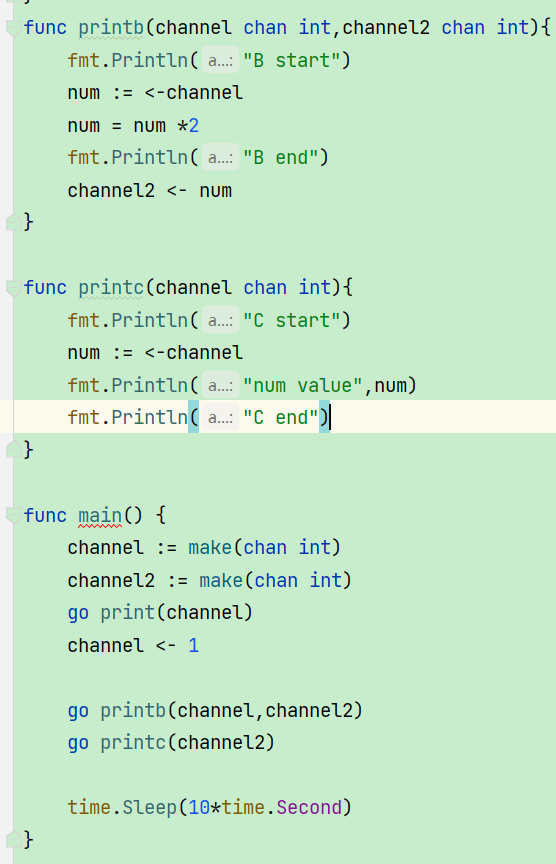
关键就是用range来完成流读取



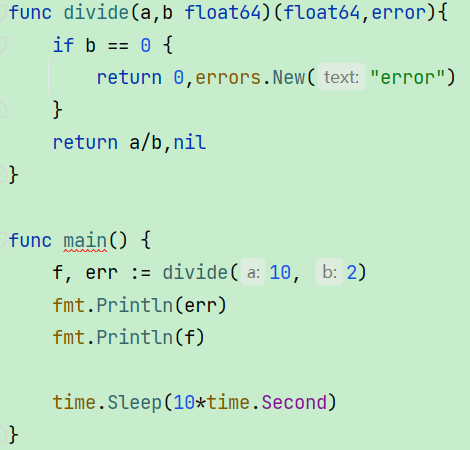
请用go语言，完成以下需求，主线程中启动A线程，A线程结束后，启动B线程，B线程结束后启动C线程



还可以互相之间进行传值



实践一下错误处理



不定义error返回值能返回error吗

那如果需要根据运行时才知道是否是错误的，这种如何处理呢？

内存相关提问和实践

Go语言中自己控制内存的方法有哪些？如果不主动释放，这块内存就会永久占用吗？Go中有内存自动回收机制吗？

Go中是有自动GC的，如果你的变量被全局变量持有，那么很有可能就不会释放类似于java永久代一样。

Go的GC收集原理是？

居然也是cms，简单来说和java差不多，只是它主要是堆，而java还有方法区

Go中除了make还有那些涉及内存资源的操作呢？

多了，各种new 分片那些都是

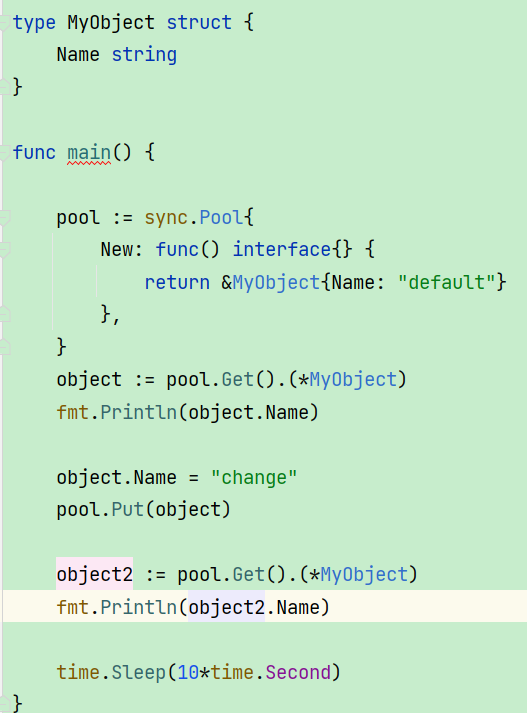
Go中make申请的资源，想要主动回收的话，需要怎么做呢？

Close关键字只是用来关闭通道的 其他资源关闭则是需要类中方法调用

Defer的作用是什么？

在Go语言中，defer语句用于安排一个函数在其所在的函数返回后执行。换句话说，当你在一个函数中使用defer调用另一个函数时，无论当前函数通过何种方式（正常返回、异常等）结束，被延迟的函数都会在当前函数即将返回之前被执行。这种机制非常有用，可以确保某些重要的清理工作得以执行，比如关闭文件、释放锁或解锁互斥量等。感觉和final一样

sync.Pool如何使用？



为什么new的时候不能直接=赋值而必须使用\*或者&呢？

因为其返回就是指针类型，会类型不匹配。

Go中有集合、hashmap这些结构吗？

有map，set可以模拟出来，go中是有泛型的，slice就是list

在go中Int\* a = 10 a = &10是等价的吗？

和c语言一样 是等价的 a中都是地址

Go中interface{}和struct{}都代表什么意思？都是如何使用的？

interface{} 在 Go 中表示一个空接口。由于 Go 的接口机制要求实现接口的所有方法，而空接口没有声明任何方法，因此所有类型都实现了空接口。这意味着你可以将任何类型的值赋给一个 interface{} 类型的变量。类似于我们的Object

interface{} 主要用于实现多态性和类型安全的泛型编程，而 struct{} 则更多地被用于特定的场景下，比如并发编程中的信号传递等。

# 实践以下框架 框架很多 大同小异 这里选择适用性大一些的gin

初始化你的模块

go mod init goo

修改数据源

go env -w GOPROXY=https://mirrors.aliyun.com/goproxy/,direct

下载

go get -u github.com/gin-gonic/gin

检查依赖是否完整

go mod tidy

gin依靠一个文件即可

创建engine

如果是default就是带了默认中间件 比如log recover

New就是干净的

中间件其实就类似于netty的Handler 一般都是日志打印 权限校验 监控这些能力

然后是创建路由、路由组

子分组会继承父分组的中间件，同时也可以添加自己的中间件

类似这样



验证使用validator库

Gin可以运行多个服务器，java也能只要创建多个Server即可

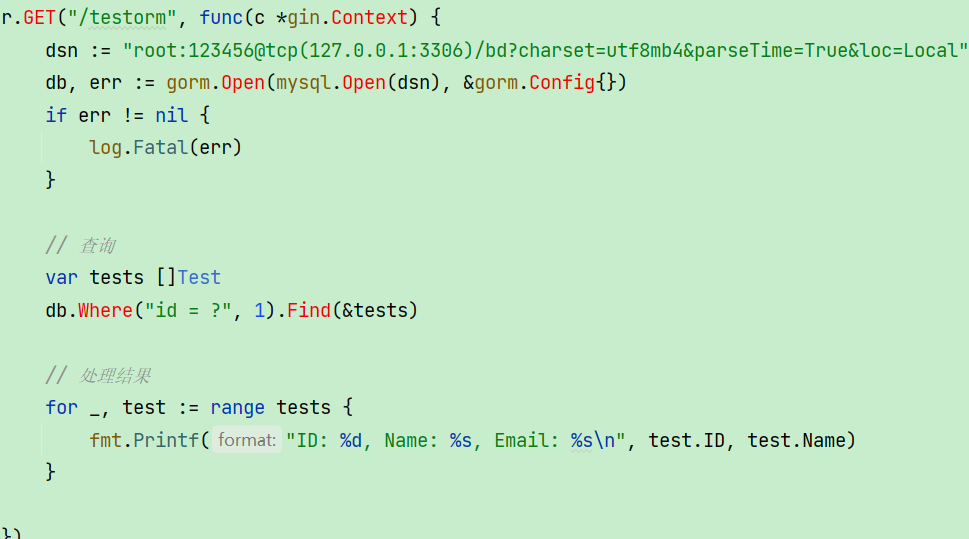
如果你懂netty的话，从java转go可以说没有任何难度，只是语法不同没有特别大的区别。

数据库操作 和python一样



用orm操作一下





数据的CURD、关联数据查询、数据库事务不展示可以自己查阅

Go的orm有预加载能力，即根据你的结构体对应关系，把相应的一次性取回来，怎么说呢底层也是执行了多个sql，但是更方便了

结构



基本上所有web都是这样 handler就是编排层 router是api层

http连接然后升级的，Websocket是数据帧格式

一般go实现的通讯都包括了用户、消息类型、聊天室等

一般业务都是发送消息到go然后go再决定发送到哪里如私聊、群聊等

Go主要负责各个实体类的创建和之间关系的维护，还有就是消息的处理也就是业务动作。

还可以实现优雅关闭和热重启

优雅关闭就是不再接受请求、处理完所有现有请求、关闭资源

热重启最重要的是，集群中不要同时重启，还有就是需要是最新的配置

两者都需要严格注意数据一致性问题

最后给大家附带一个教程网址 很多详细内容大家可以看这里

https://blog.csdn.net/GopherTribe/article/details/146546615