

初识Golang

📅 2018-07-29 | 👁 8

1. Golang概述

Golang以Ken Thompson为首开发设计，2009年由谷歌开源，是一种现代化编程语言。因此，Golang吸收和借鉴了众多编程语言的优点，崇尚简单的编程哲学，方便处理实际工程上的问题。例如提供更加方便灵活的类型系统、简单高效的并发编程模型、相对统一的编码规范。

主要特点：

1. 静态强类型语言
2. 支持自动垃圾回收，方便内存管理
3. 提供灵活丰富的类型系统，除了常用的类型之外，提供字符串、数组、数组切片slice、字典类型map等常用数据结构类型
4. 放弃了构造函数、析构函数、虚函数、继承等面向对象的组件，构建以“**结构体struct**”、“**接口**”，“**组合**”为基础的面向对象的编程方式。
5. Golang非侵入式的接口，值得深入学习
6. Golang从语言层面支持协程，以goroutine和channel为基础，构建高效便捷的并发模型。相比内存（数据）共享，它更加推崇消息传递的并发模型
7. Golang非常适合工程实践，在工程管理和编码规范有一些强制性的规定，例如以目录结构为基础的工程管理方式、**标准错误处理的规范**、支持函数多返回值等
8. 支持反射

2. Golang开发环境

“win10+VMware14+ubuntu16.04+Sublime3”

补充：sublime3代理的配置

3. 基本语法

3.1 变量与类型

- 变量（声明、初始化、声明和初始化、支持多重赋值和匿名变量），由于Golang存在右值类型推导，所以有点类似动态语言，但是Golang是非常严格的静态强类型语言

- 常量 (const、itoa)、枚举
- bool类型只支持true和false，不支持0和1
- 整数类型 (不同类型的整型数不同相互比较、运算)
- 浮点数 (float32、float64，小数会被自动推到成float64、浮点数的比较)
- 字符串string是内置的基本类型，一旦初始化后不允许修改，当字符串中包含非ANSI的字符时，注意将源码的编码格式设置为UTF-8
- 字符类型byte
- 数组类型 (固定长度；注意Golang中的数组是值类型，这意味着传参需要复制产生副本)
- 数组切片slice(动态变长数组，相当于vector，注意它的创建方式make)
- map类型 (注意Golang中map是未排序的，而C++中的map是排序的；创建、赋值、删除、查找)

3.2 逻辑控制

- 条件语句,if...else...
- 选择语句，在switch...case...结构中，不需要用break来明确退出一个case
- 循环语句，不支持while和do..while结构,和循环相关结构全部使用for结构代替，包括“无限循环”等
- Golang支持goto跳转结构

3.3 函数

- 函数的定义，“func 函数名 (参数) (返回值)”，相邻参数类型相同，可以合并
- 函数名称的大小写，Golang中大写字母开头的函数能被其它包调用，小写字母开头的函数只能在本包内可见
- Golang支持不定参数，同种类型的不定参数 (args ...int) ,任意类型的不定参数 (args ...interface{})
- 函数多返回值，返回值可以命名也可以不命名，命名返回值会使得代码更加清晰
- Golang支持随时在代码中定义匿名函数
- 支持闭包

3.4 错误处理规范

- error接口，nil，返回错误类型
- 自定义错误类型。实现error接口，error接口只包含一个Error方法
- defer错误，异常的延迟处理
- panic和recover函数

4. 面向对象编程

1. Golang中类型大多数基于值传递，通过type关键字自定义类型，任何类型都可以添加方法
2. 深入理解“值语义”和“引用语义”，注意Golang中数组属于值语义
3. Golang放弃class和继承等大量面向对象的特性，只保留了struct和组合等基础特性
4. Golang中的面向对象在语言层面表现为结构体类型struct和类型的成员方法
5. 没有构造函数，通常由全局创建函数NewXXX来完成对象的构建
6. 对象的构造和初始化,{}
7. Golang放弃了类的继承，支持类的组合方法

8. Golang中没有public、private等关键字，通过控制变量或者函数名称的大小写来控制可见性。Golang的可见性是包级别的，而不是类型级别的。

9. Golang的接口interface是类型系统的基石

10. 侵入式的接口和非侵入式的接口，Golang非侵入式接口的优势？先有类型，再定义接口？一个类只需要实现接口所要求的所有函数，我们就说这个类实现了该接口
11. 接口赋值（对象实例给接口赋值，接口给接口赋值）
12. 接口查询，判断某一对象是否实现某个接口
13. Golang支持类型查询，后面也可利用反射功能进行类型查询
14. Golang支持多种接口的组合
15. Any类型与空接口，interface{}

5. 并发编程

基于goroutine协程加channel的消息传递的并发模型

1. once类型，保证全局唯一——一次性操作，处理某些函数只需要执行一次的情况
2. 互斥锁和读写锁；结合defer的经典锁使用模式
3. channel用于goroutine之间进行消息传递和数据共享；
4. channel是类型相关的，channel在定义的时候必须指明类型，一个channel只能传递一种类型的值
5. 多个同类型的channel构成channel数组
6. channel类似管道，分为数据的读和写，没有数据时，读操作会被阻塞，反之有数据时，写操作会被阻塞
7. channel的声明、定义与基本类型相似，在类型之前增加chan
8. channel的缓冲机制
9. channel的超时机制，避免永久死锁，select加超时等待goroutine
10. channel关闭和关闭的判断
11. Golang并发编程原则：协程同步优先考虑使用基于channel的消息通信机制完成，之后再考虑锁机制

6. 实战

- **实战一：**根据播放器样例代码，熟悉Golang的基本语法和面向对象的特性
- **实战二：**棋牌游戏服务器样例代码，深入理解Golang的并发模型

◀ 基于Github双分支和Hexo搭建博客