测试和调试

作者：少林之巅

目录

自动化测试框架

1. testing

    A. testing包提供了自动化测试相关的框架

    B. 支持单元测试和压力测试

```
import (
    “testing”
)
```

测试规范

2. Go中的测试约定

   A. 用来测试的代码必须以_test.go结尾

   B. 单元测试的函数名必须以Test开头， 并且只有一个参数,类型是 *Testing.T

   C. 基准测试或压力测试必须以 Benchmark开头，并且只有参数，类型是*Testing.B

```
import (
    “testing”
)

func TestAdd(t *testing.T) {

}
```

```
import (
    “testing”
)

func BenchmarkAdd(t *testing.B) {

}
```

单元测试

3. 单元测试

   A. 对于各个分支进行测试，如果不符合预期则失败

   B. 使用testing.T这个对象进行单元测试控制

4. 基准测试

A. 主要用来做性能测试。

B. go test自动会执行所有的基准测试，并且打印执行耗时统计

```
package main

import (
      "fmt"
    “testing"
)

func BenchmarkHello(b *testing.B) {
    for i := 0; i < b.N; i++ {
        fmt.Sprintf("hello")
    }
}
```

Go test命令介绍

5. Go test命令介绍

A. go test加报名，执行这个包下面的所有测试用例

B. Go test加测试源文件，执行这个测试源文件里的所有测试用例

C. go test -run选项，执行只定的测试用例

Delve进行调试

6. delve工具介绍

   A. 追踪程序中的异常代码

   B. 通过打日志的方式，追查问题比较低效

   C. 能够有一种工具，直接分析程序执行的情况，就牛逼了

Delve进行调试

7. delve安装

   A. go get github.com/derekparker/delve/cmd/dlv

   B. 默认安装到GOPATH的bin目录下

   C. 把安装目录设置到PATH环境变量中

Delve进行调试

8. Delve使用

A. dlv命令， dlv debug 包的路径或源代码路径

B. Dlv会编译我们的程序，然后进入调试界面

C. 进入调试界面后，就可以一步一步的执行的我们代码了

D. 使用vscode进行调试，图形界面的使用

Delve进行调试

9. 调试正在运行的程序

   A. dlv命令， dlv attach 进程id

   B. 多线程调试