**Code Kata-Go**

**2019-04-17**

**Prerequisites**

1. Download and install Go (e.g., D:\Go)
2. Go: <https://golang.google.cn/dl/>
3. Set GOROOT=D:\Go; GOPATH=D:\workspace\github.com\rwang5688\go
4. Download and install LiteIDE (e.g., D:\liteide)
5. LiteIDE: <http://liteide.org/cn/>

**Bowling Game**

1. Unzip bowling\_game-Go-1stTest.zip under D:\workspace\github.com\$(UserId)\go
2. **执行任何操作前：export GOPATH=/d/workspace/github.com/$(UserId)/go/bowling\_game**
3. Start LiteIDE
4. Open directory: D:\workspace\github.com\$(UserId)\go\bowling\_game; 以下是目录结构：
   1. bin
   2. pkg
      1. windows\_amd64
         1. app
   3. src
      1. app
         1. bowling\_game: “bowling\_game” package; under “pkg/windows\_amd64/app”, install bowling\_game.a
         2. bowling\_game\_app: “main” package; under “bin”, install bowling\_game\_app.exe
      2. test
         1. unit
            1. bowling\_game: “bowling\_game\_unittest” package
5. Follow Bowling Game code kata with “package bowlingGame”
6. Golang idiosyncracies
   1. export GOPATH=$(Custom GOPATH)，e.g.：
   2. export GOPATH=/d/workspace/github.com/$(UserId)/go/bowling\_game
   3. package must be "main" to install $(directory\_name).exe to $(GOPATH)/bin
7. **Package - Clean All:**
8. **export GOPATH=/d/workspace/github.com/$(UserId)/go/bowling\_game**
9. **cd /d/workspace/github.com/rwang5688/go/bowling\_game/src/app/bowling\_game (or bowling\_game\_app)**
10. **go clean -i**
11. OR
12. Go to src/app/bowling\_game, select “game.go”; or src/app/bowling\_game\_app, select “bowling\_game\_app.go”
13. Click 编译-》编译配置
14. Select self-defined GOPATH, and add：
15. “D:\workspace\github.com\$(UserId)\go\bowling\_game” (export GOPATH=...)
16. Click open the big blue “B” button and click on the “C!” button to execute:
17. “D:/Go/bin/go.exe clean -i
18. [D:/workspace/github.com/rwang5688/go/bowling\_game/src/app/bowling\_game]”
19. **Package - Test:**
20. **export GOPATH=/d/workspace/github.com/$(UserId)/go/bowling\_game**
21. **cd /d/workspace/github.com/rwang5688/go/bowling\_game/src/test/unit/bowling\_game**
22. **go test -v**
23. OR
24. Go to src/test/unit/bowling\_game, select “game\_test.go”
25. Click 编译-》编译配置
26. Select self-defined GOPATH, and add：
27. “D:\workspace\github.com\$(UserId)\go\bowling\_game” (export GOPATH=...)
28. Click on the big blue “T” button to execute:
29. “D:/Go/bin/go.exe test -v [D:/workspace/github.com/rwang5688/go/bowling\_game]”
30. **Package - Install:**
31. **export GOPATH=/d/workspace/github.com/$(UserId)/go/bowling\_game**
32. **cd /d/workspace/github.com/rwang5688/go/bowling\_game/src/app/bowling\_game (or bowling\_game\_app)**
33. **go install -v (or go run main.o .... 这样不用install EXE)**
34. OR
35. Go to src/app/bowling\_game, select “game.go”; or src/app/bowling\_game\_app, select “bowling\_game\_app.go”
36. Click 编译-》编译配置
37. Select self-defined GOPATH, and add：
38. “D:\workspace\github.com\$(UserId)\go\bowling\_game” (export GOPATH=...)
39. Click open the big blue “B” button and click on the “I” button to execute:
40. “D:/Go/bin/go.exe install -v
41. [D:/workspace/github.com/rwang5688/go/bowling\_game/src/app/bowling\_game]”

**Golang Unit Test Framework**

Link: <https://studygolang.com/articles/12171?fr=sidebar>

这里编写测试函数，有几个需要注意的点：

1. 每个测试文件必须以 ****\_test.go**** 结尾，不然 go test 不能发现测试文件
2. 每个测试文件必须导入 ****testing**** 包
3. 功能测试函数必须以 ****Test**** 开头，然后一般接测试函数的名字，这个不强求

根据这些条件，我们可以写出一个测试文件：

