**Code Kata-Go**

**2019-08-08**

**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\$(UserId)\go
4. Download and install LiteIDE (e.g., D:\liteide)
5. LiteIDE: <http://liteide.org/cn/>
6. Copy Code Kata-Go zipfiles to D:\workspace\github.com\$(UserId)\go\zipfiles

**Bowling Game**

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

