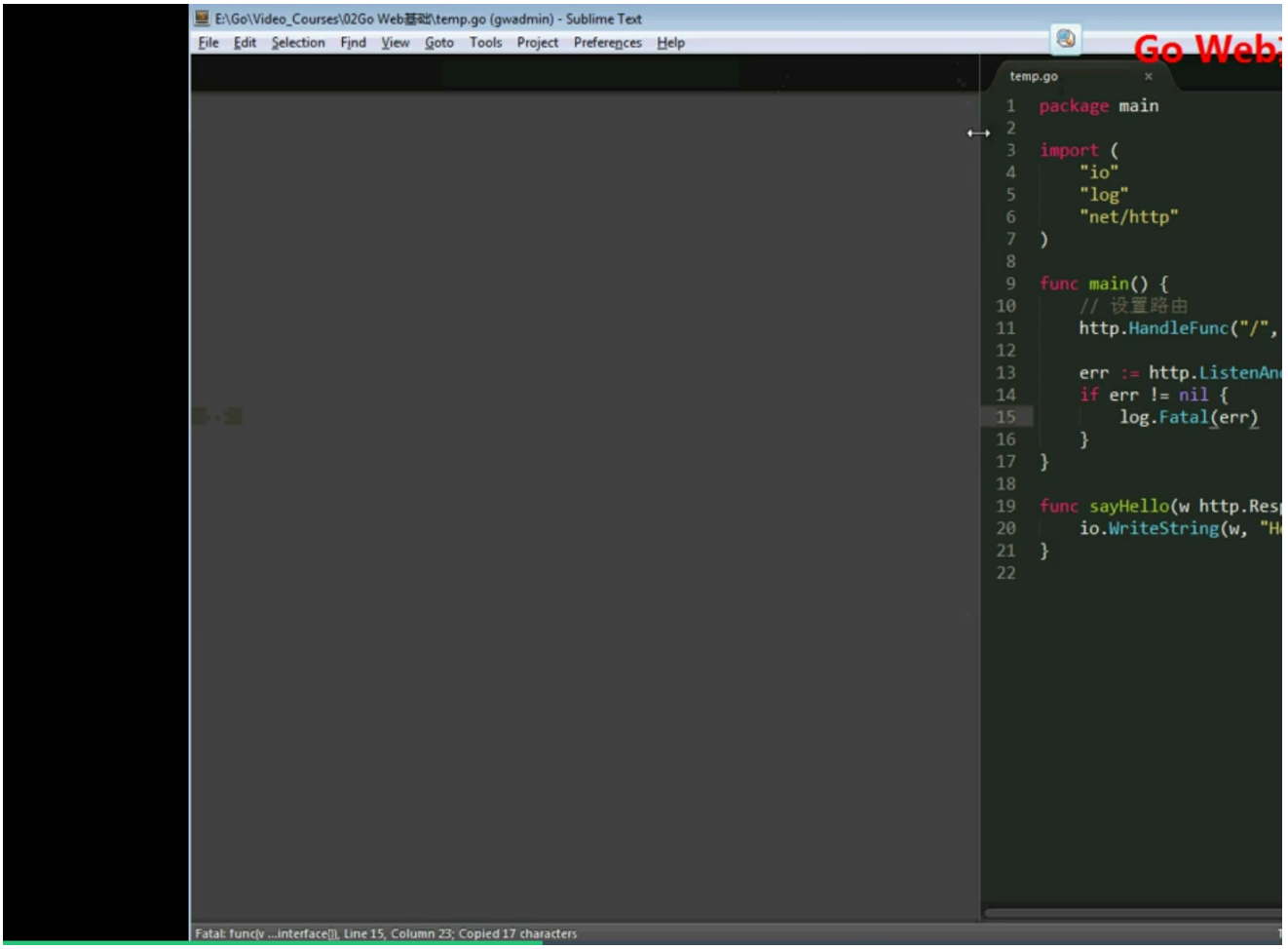
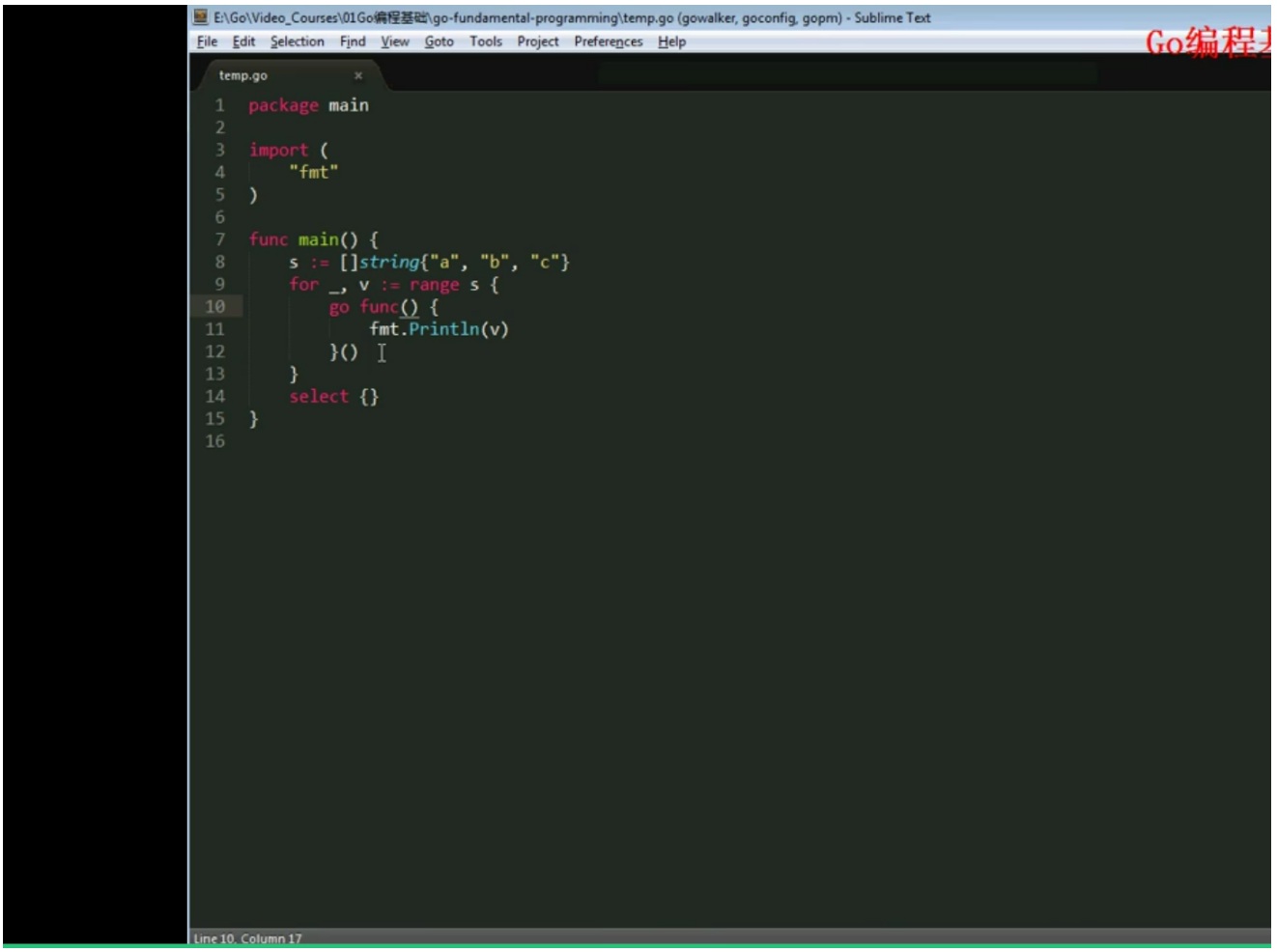
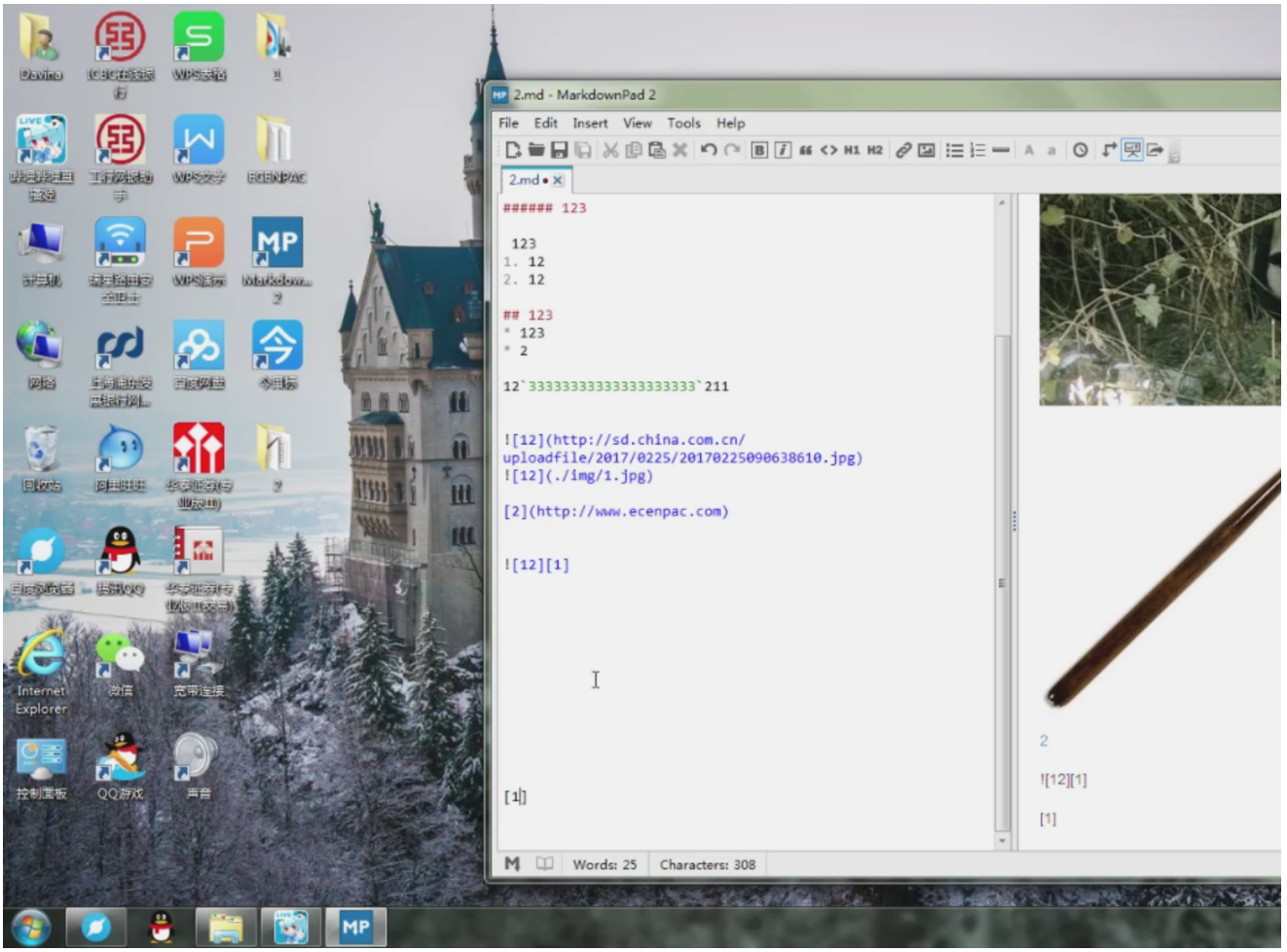
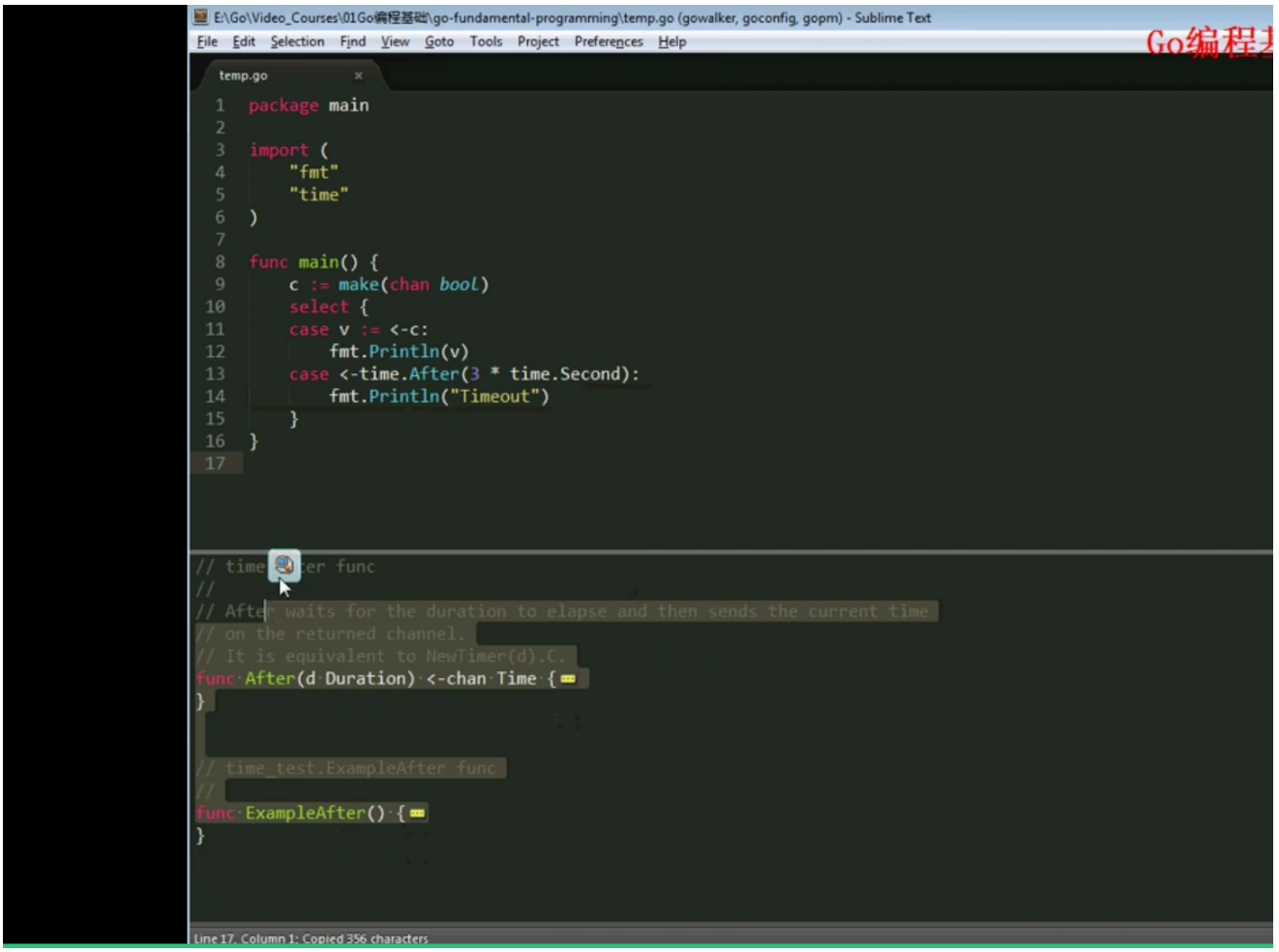
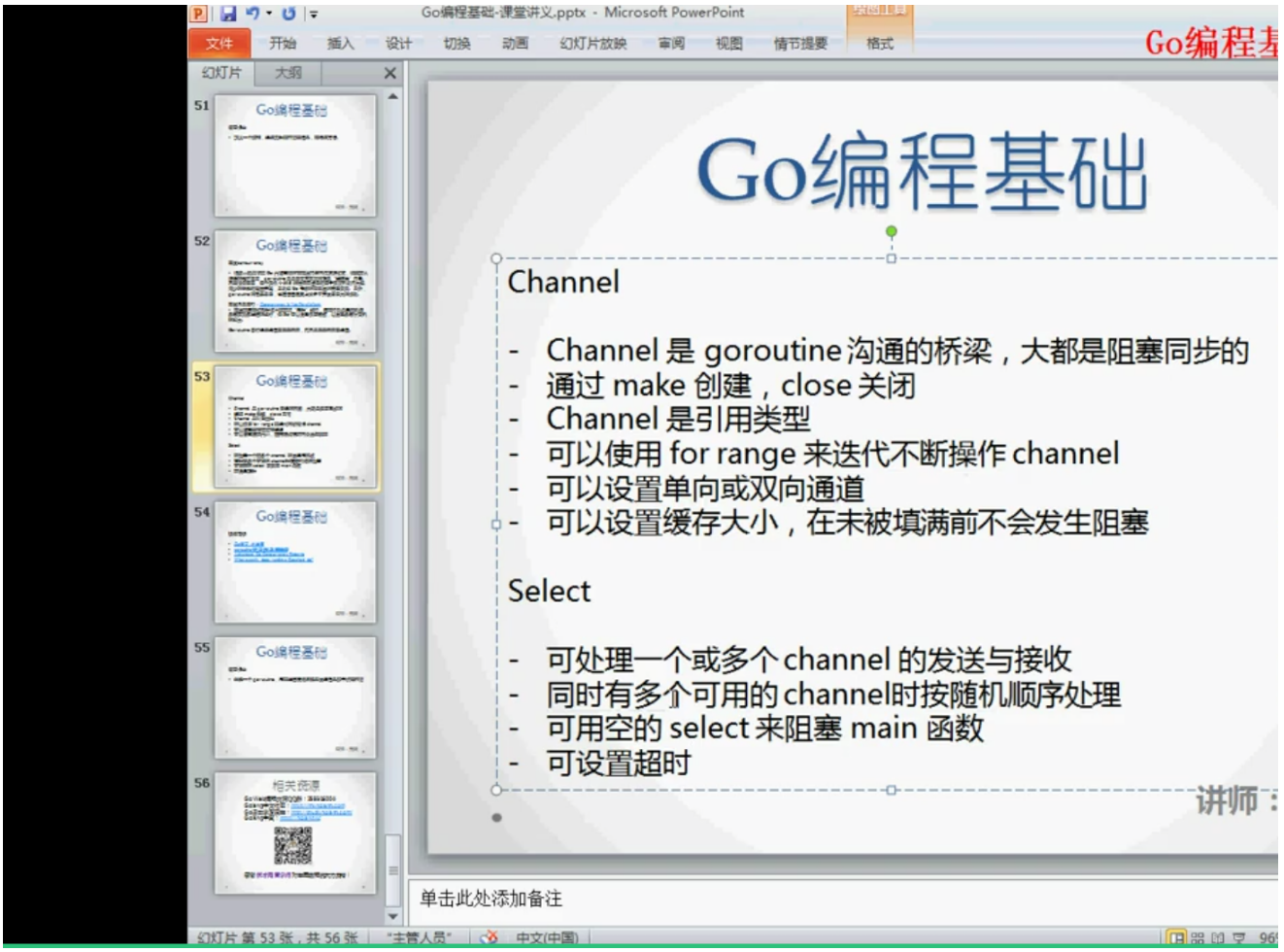
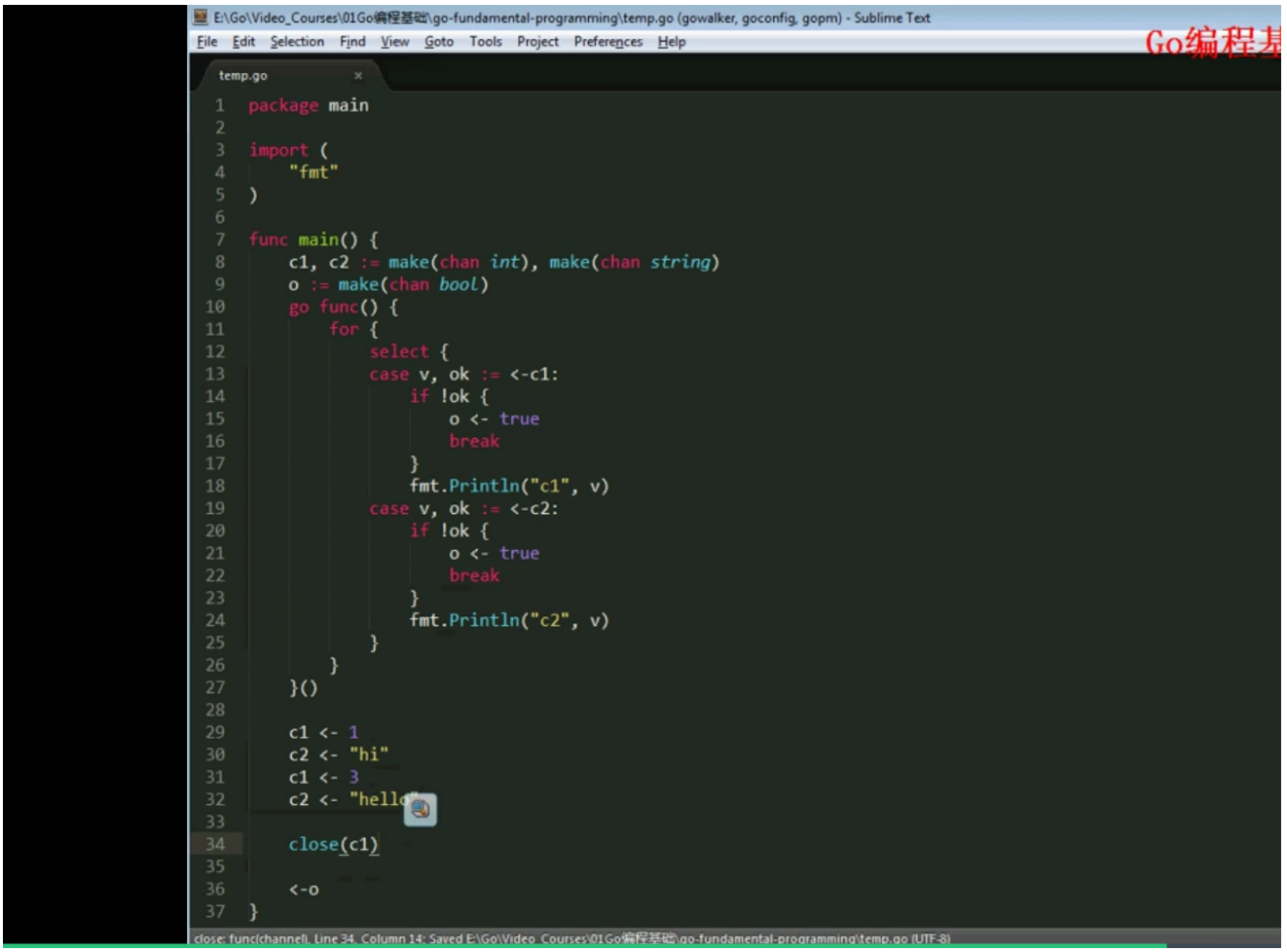

E:\Go\Video_Courses\02Go Web基础\temp.go (gwadmin) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go Web
temp.go
1 package main
2
3 import (
4 "io"
5 "log"
6 "net/http"
7)
8
9 func main() {
10 // 设置路由
11 http.HandleFunc("/",
12
13 err := http.ListenAn
14 if err != nil {
15 log.Fatal(err)
16 }
17 }
18
19 func sayHello(w http.Res
20 io.WriteString(w, "H
21 }
22
Fatal: func(v ...interface{}), Line 15, Column 23; Copied 17 characters

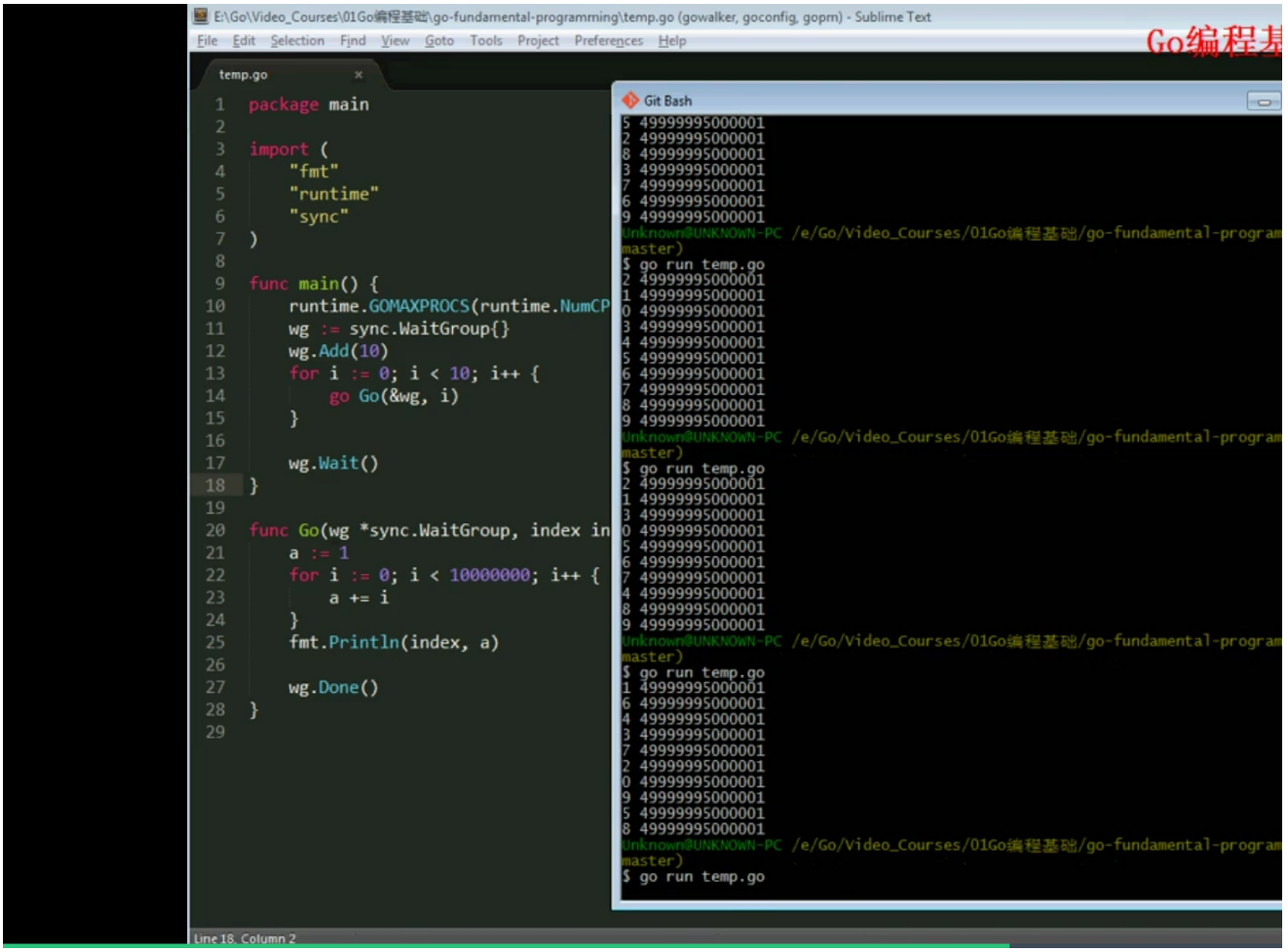

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 s := []string{"a", "b", "c"}
9 for _, v := range s {
10 go func() {
11 fmt.Println(v)
12 }()
13 }
14 select {}
15 }
16
Line 10, Column 17

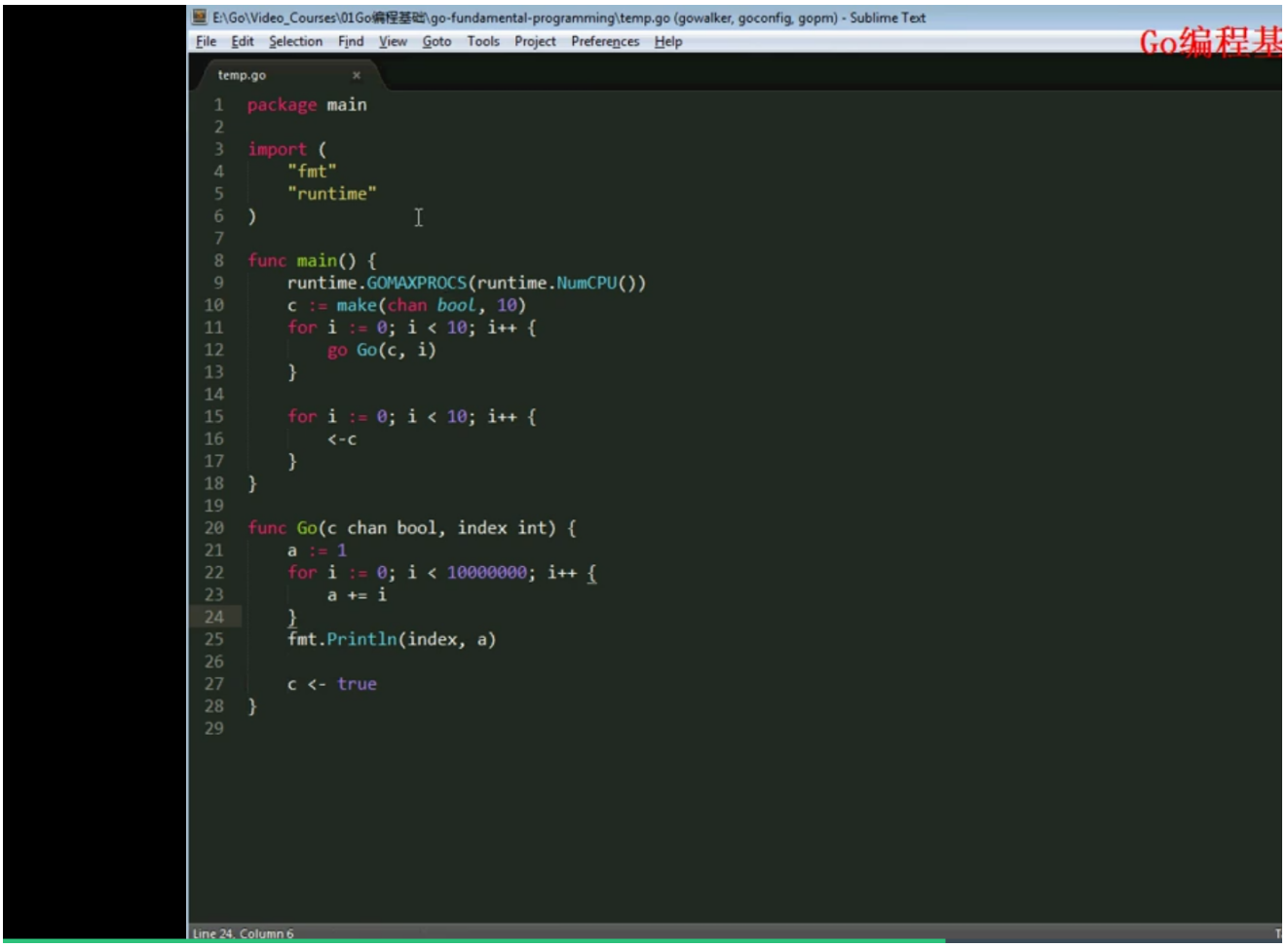

2.md - MarkdownPad 2
File Edit Insert View Tools Help
2.md
123
123
1. 12
2. 12
123
* 123
* 2
12`3333333333333333333`211
![12](http://sd.china.com.cn/uploadfile/2017/0225/20170225090638610.jpg)
![12](./img/1.jpg)
[2](http://www.ecenpac.com)
![12][1]
[1]
2
![12][1]
[1]
Words: 25
Characters: 308

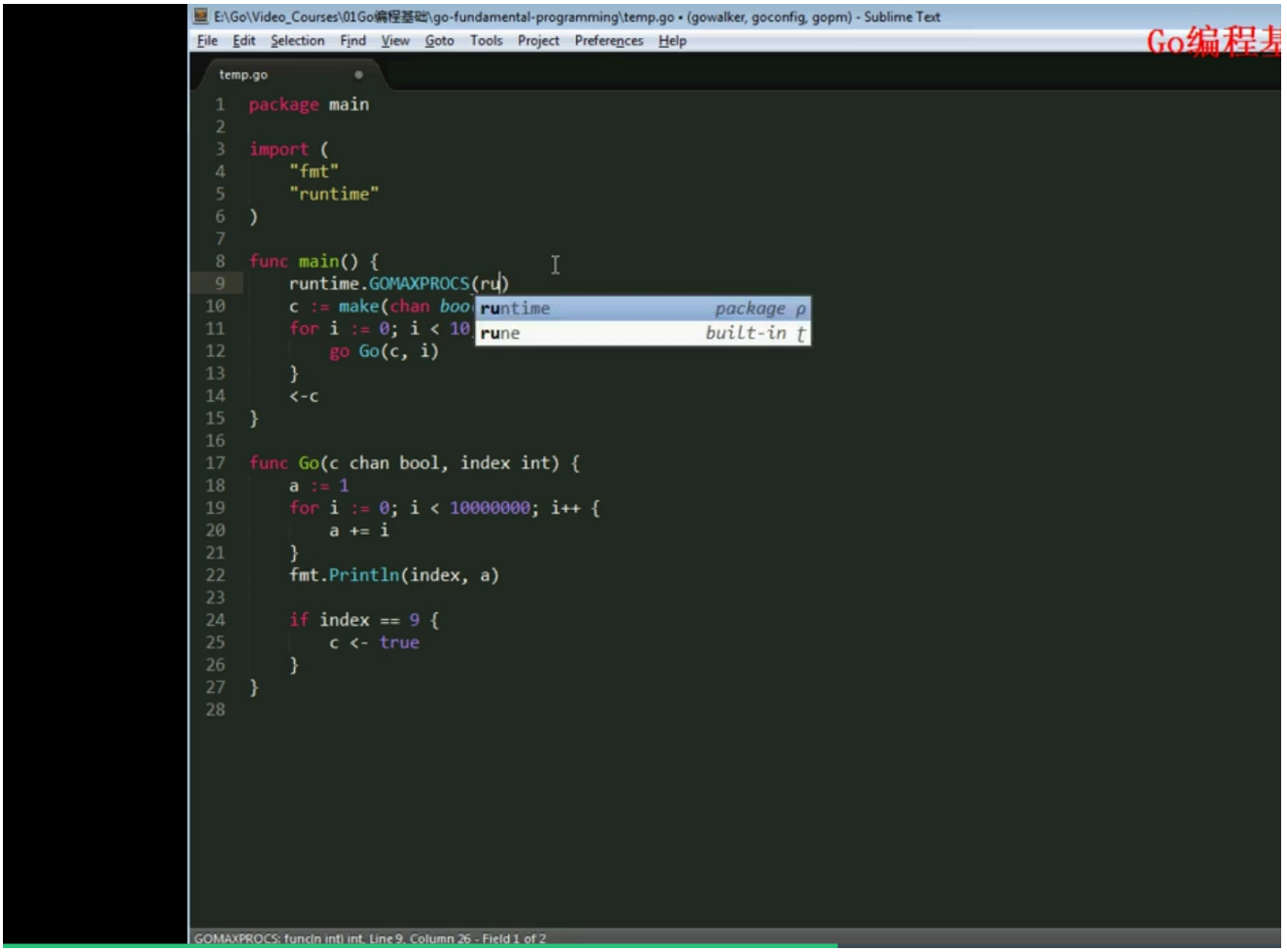

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "time"
6)
7
8 func main() {
9 c := make(chan bool)
10 select {
11 case v := <-c:
12 fmt.Println(v)
13 case <-time.After(3 * time.Second):
14 fmt.Println("Timeout")
15 }
16 }
17
// time.After func
//
// After waits for the duration to elapse and then sends the current time
// on the returned channel.
// It is equivalent to NewTimer(d).C.
func After(d Duration) <-chan Time {
}
// time_test.ExampleAfter func
//
func ExampleAfter() {
}
Line 17, Column 1; Copied 356 characters

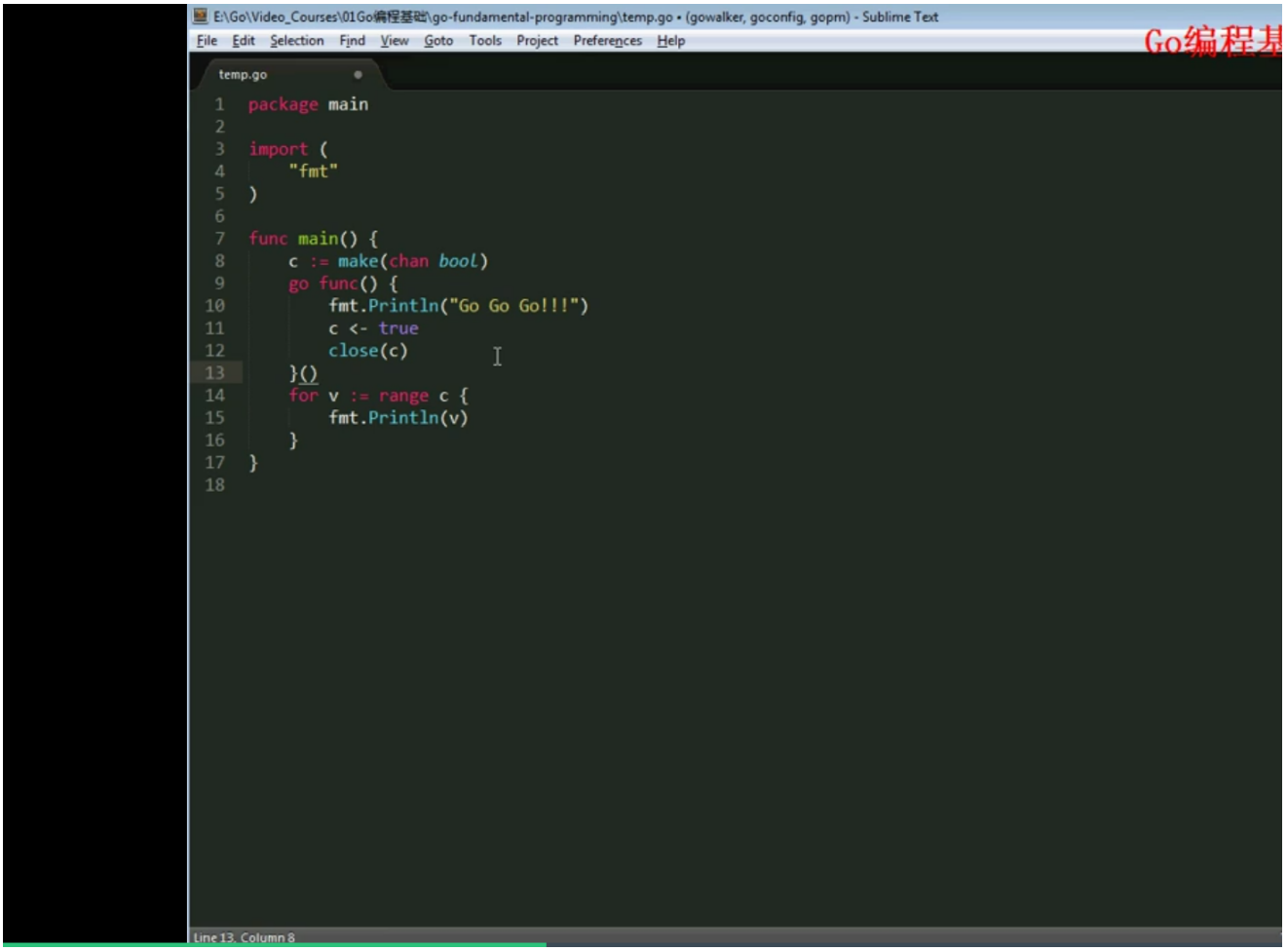

Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
文件 开始 插入 设计 切换 动画 幻灯片放映 审阅 视图 情节提要 格式
幻灯片 大纲
Go编程基础
Channel
- Channel 是 goroutine 沟通的桥梁，大都是阻塞同步的
- 通过 make 创建，close 关闭
- Channel 是引用类型
- 可以使用 for range 来迭代不断操作 channel
- 可以设置单向或双向通道
- 可以设置缓存大小，在未被填满前不会发生阻塞
Select
- 可处理一个或多个 channel 的发送与接收
- 同时有多个可用的 channel时按随机顺序处理
- 可用空的 select 来阻塞 main 函数
- 可设置超时
讲师：
单击此处添加备注
幻灯片 第 53 张，共 56 张
中文(中国)

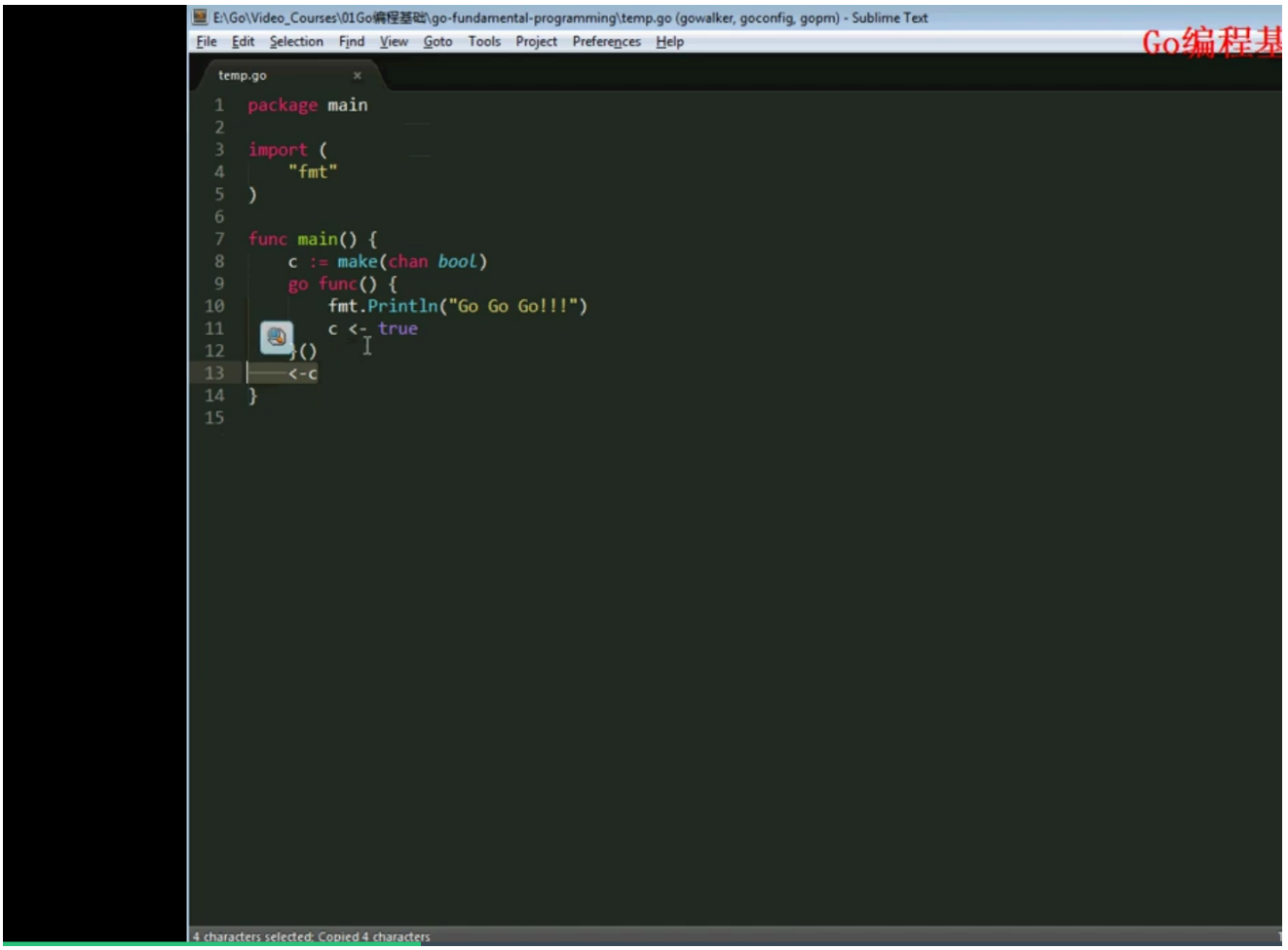

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 c1, c2 := make(chan int), make(chan string)
9 o := make(chan bool)
10 go func() {
11 for {
12 select {
13 case v, ok := <-c1:
14 if !ok {
15 o <- true
16 break
17 }
18 fmt.Println("c1", v)
19 case v, ok := <-c2:
20 if !ok {
21 o <- true
22 break
23 }
24 fmt.Println("c2", v)
25 }
26 }
27 }()
28
29 c1 <- 1
30 c2 <- "hi"
31 c1 <- 3
32 c2 <- "hello"
33
34 close(c1)
35
36 <-o
37 }
close: func(channel), Line 34, Column 14; Saved E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (UTF-8)

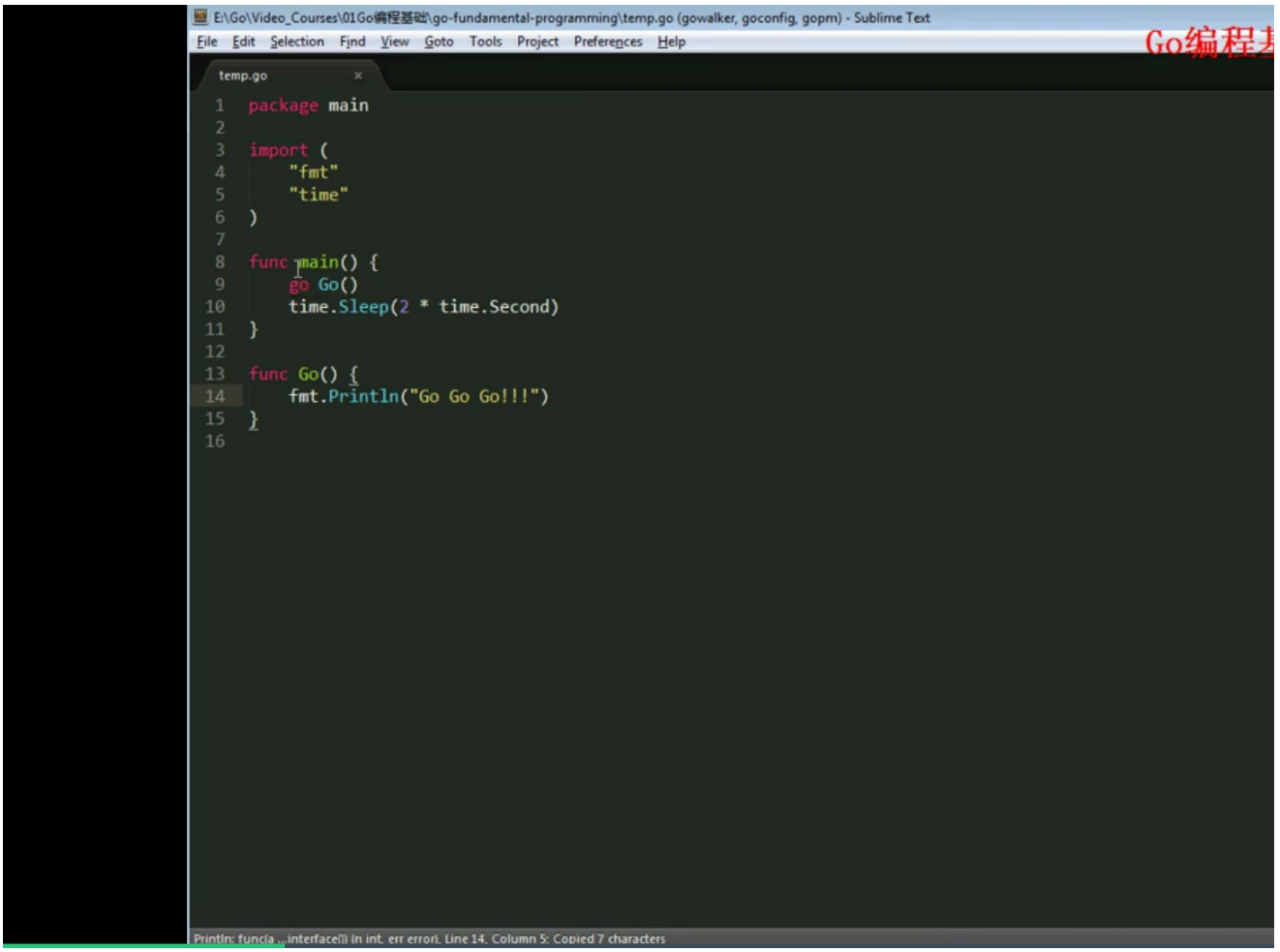

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "runtime"
6 "sync"
7)
8
9 func main() {
10 runtime.GOMAXPROCS(runtime.NumCP
11 wg := sync.WaitGroup{}
12 wg.Add(10)
13 for i := 0; i < 10; i++ {
14 go Go(&wg, i)
15 }
16
17 wg.Wait()
18 }
19
20 func Go(wg *sync.WaitGroup, index in
21 a := 1
22 for i := 0; i < 10000000; i++ {
23 a += i
24 }
25 fmt.Println(index, a)
26
27 wg.Done()
28 }
29
Line 18, Column 2
Git Bash
5 49999995000001
2 49999995000001
8 49999995000001
3 49999995000001
7 49999995000001
6 49999995000001
9 49999995000001
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go编程基础/go-fundamental-program
master)
$ go run temp.go
2 49999995000001
1 49999995000001
0 49999995000001
3 49999995000001
4 49999995000001
5 49999995000001
6 49999995000001
7 49999995000001
8 49999995000001
9 49999995000001
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go编程基础/go-fundamental-program
master)
$ go run temp.go
2 49999995000001
1 49999995000001
3 49999995000001
0 49999995000001
5 49999995000001
6 49999995000001
7 49999995000001
4 49999995000001
8 49999995000001
9 49999995000001
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go编程基础/go-fundamental-program
master)
$ go run temp.go
1 49999995000001
6 49999995000001
4 49999995000001
3 49999995000001
7 49999995000001
2 49999995000001
0 49999995000001
9 49999995000001
5 49999995000001
8 49999995000001
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go编程基础/go-fundamental-program
master)
$ go run temp.go

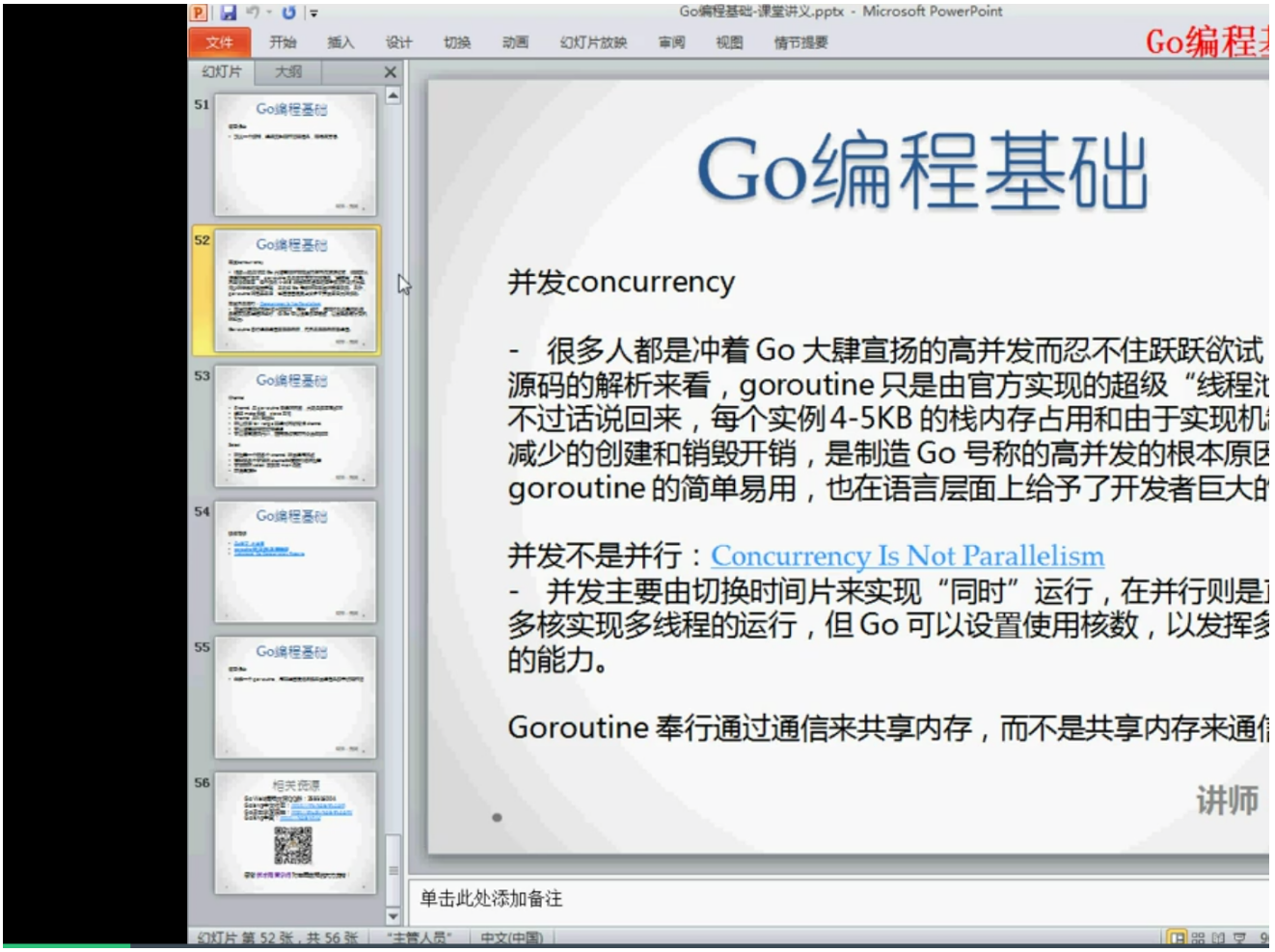

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "runtime"
6)
7
8 func main() {
9 runtime.GOMAXPROCS(runtime.NumCPU())
10 c := make(chan bool, 10)
11 for i := 0; i < 10; i++ {
12 go Go(c, i)
13 }
14
15 for i := 0; i < 10; i++ {
16 <-c
17 }
18 }
19
20 func Go(c chan bool, index int) {
21 a := 1
22 for i := 0; i < 10000000; i++ {
23 a += i
24 }
25 fmt.Println(index, a)
26
27 c <- true
28 }
29
Line 24, Column 6

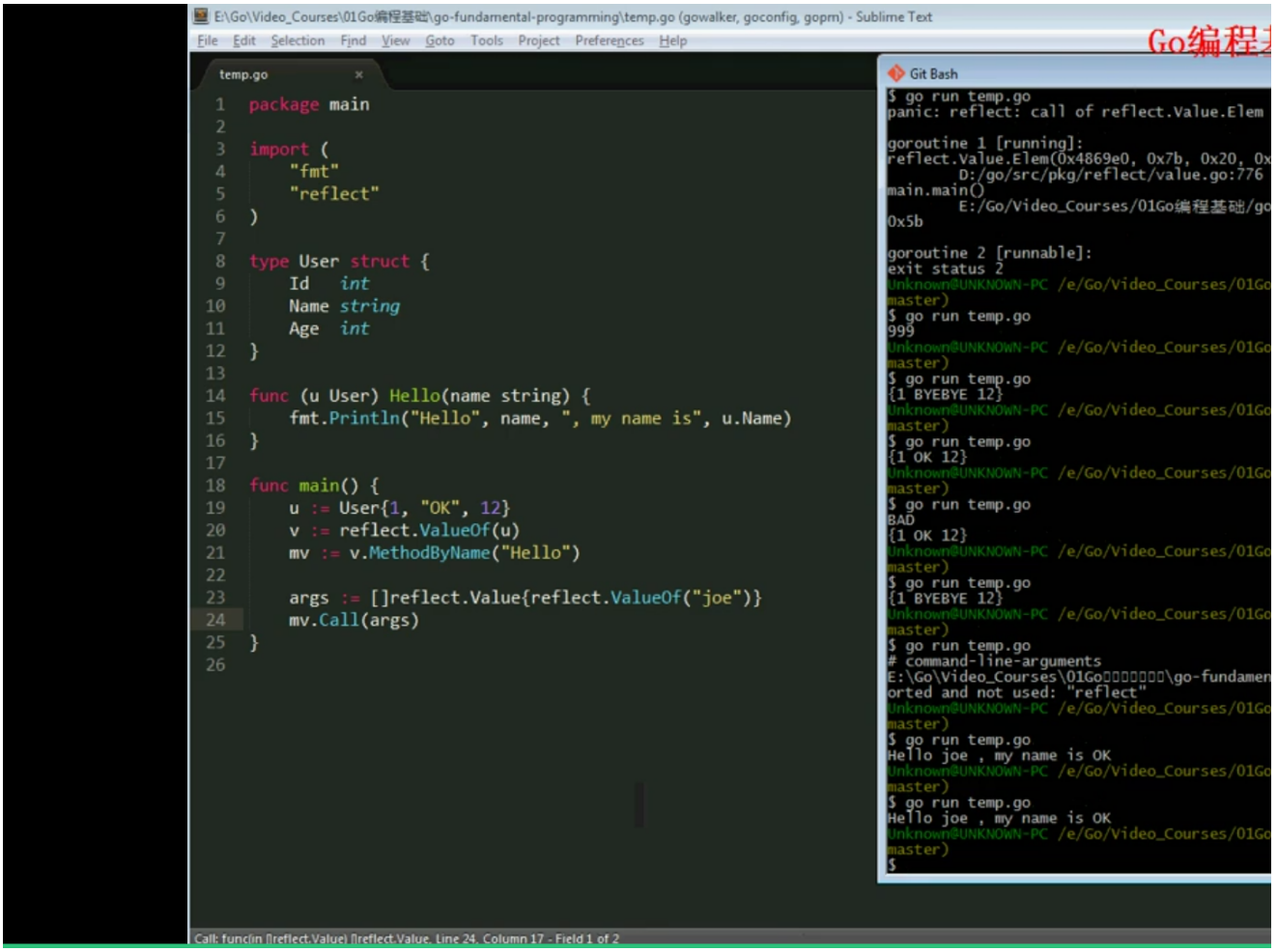

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "runtime"
6)
7
8 func main() {
9 runtime.GOMAXPROCS(ru)
10 c := make(chan boo
11 for i := 0; i < 10
12 go Go(c, i)
13 }
14 <-c
15 }
16
17 func Go(c chan bool, index int) {
18 a := 1
19 for i := 0; i < 10000000; i++ {
20 a += i
21 }
22 fmt.Println(index, a)
23
24 if index == 9 {
25 c <- true
26 }
27 }
28
runtime package p
rune built-in t
GOMAXPROCS: func(n int) int, Line 9, Column 26 - Field 1 of 2

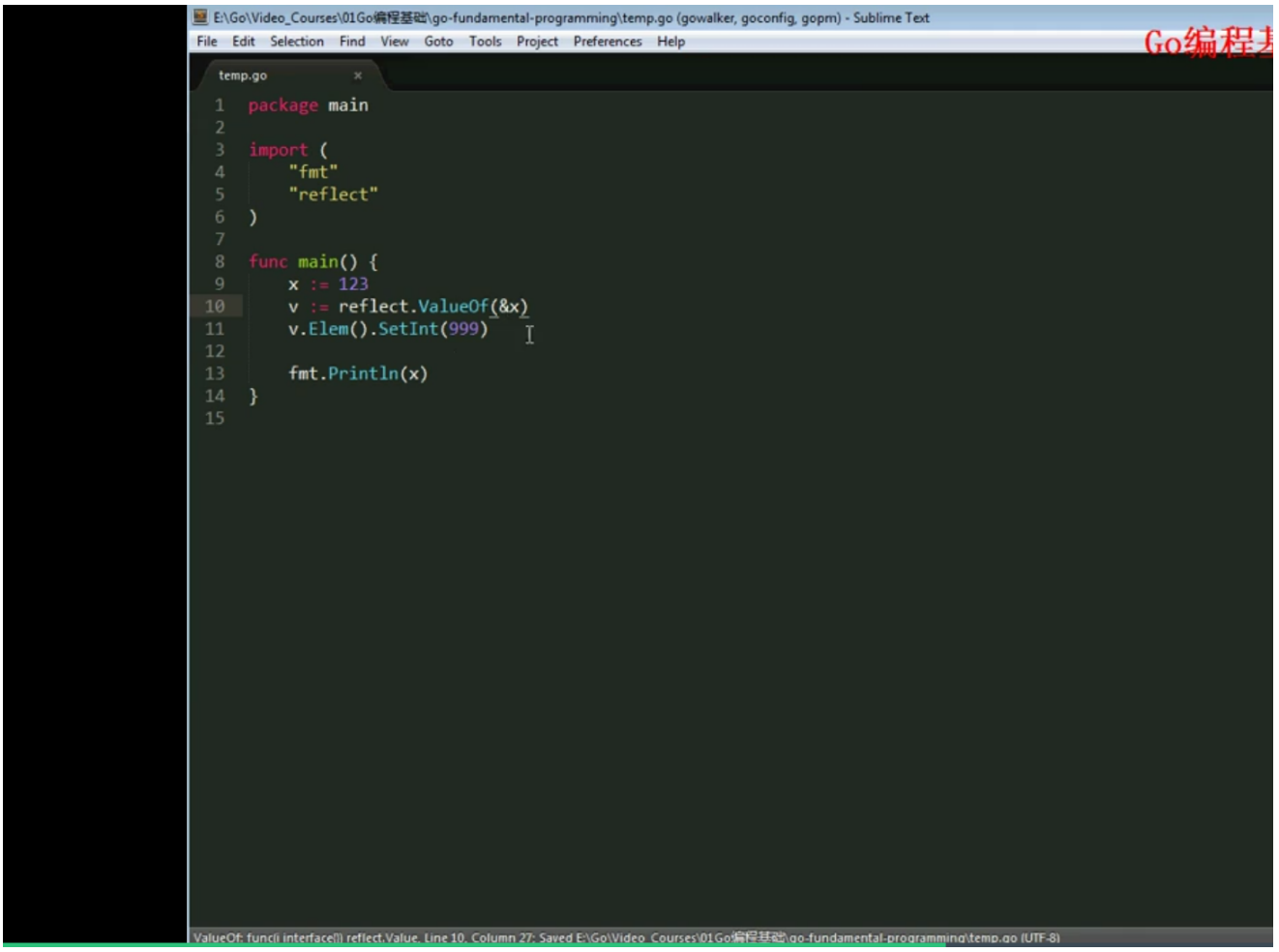

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 c := make(chan bool)
9 go func() {
10 fmt.Println("Go Go Go!!!")
11 c <- true
12 close(c)
13 }()
14 for v := range c {
15 fmt.Println(v)
16 }
17 }
18
Line 13, Column 8

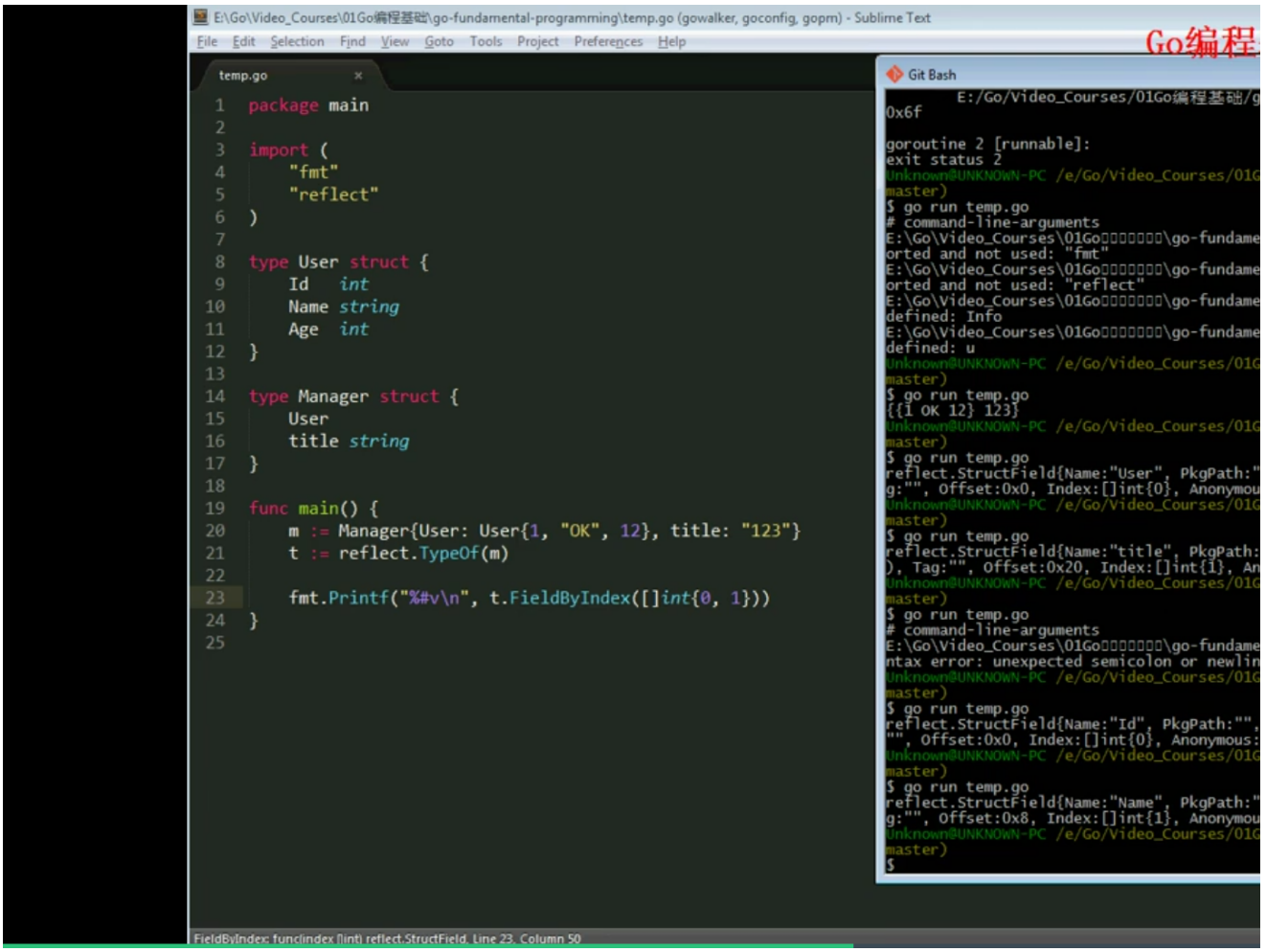

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 c := make(chan bool)
9 go func() {
10 fmt.Println("Go Go Go!!!")
11 c <- true
12 }()
13 <-c
14 }
15
4 characters selected: Copied 4 characters

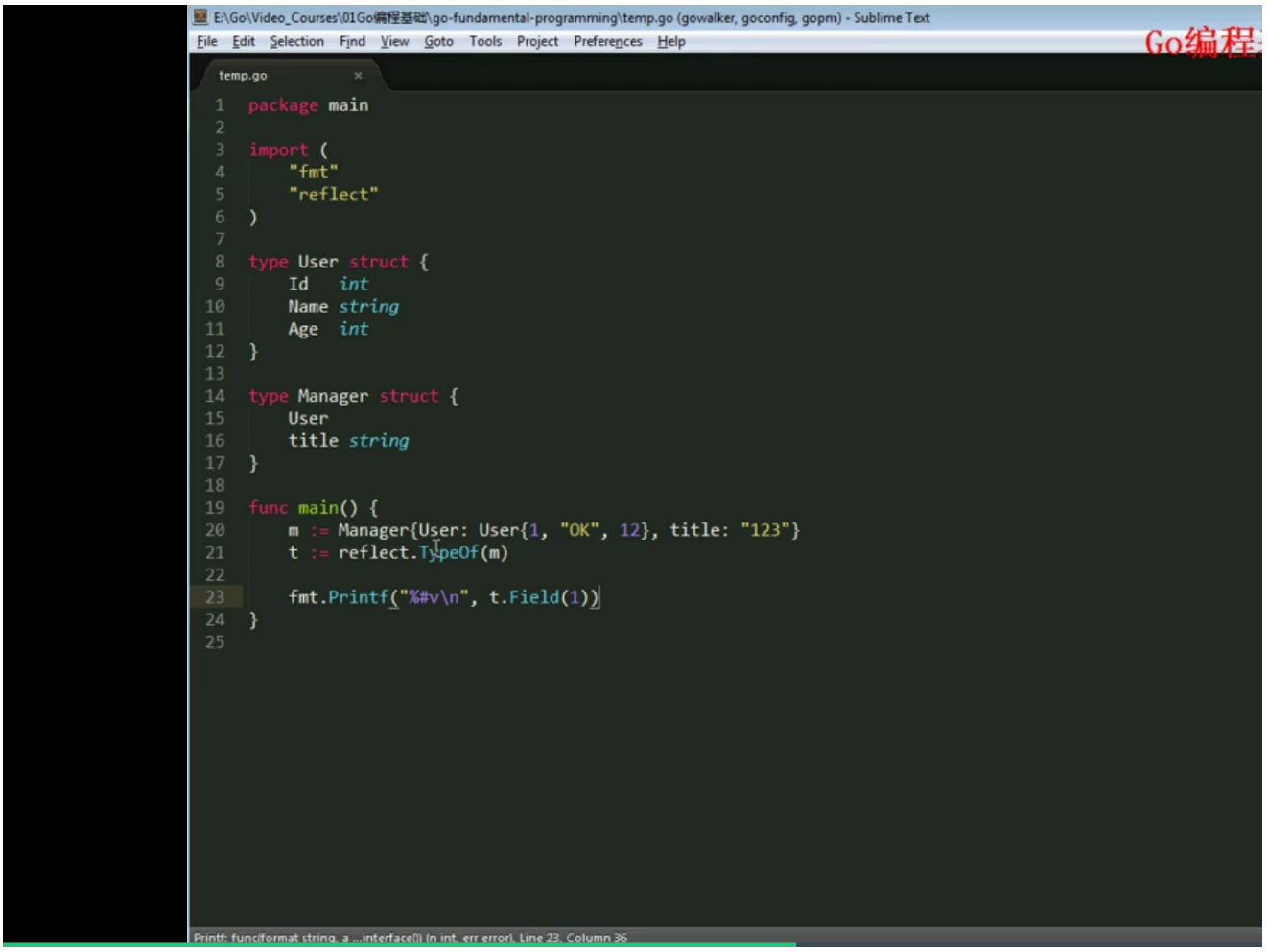

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "time"
6)
7
8 func main() {
9 go Go()
10 time.Sleep(2 * time.Second)
11 }
12
13 func Go() {
14 fmt.Println("Go Go Go!!!")
15 }
16
Println: func(a ...interface{}) (n int, err error), Line 14, Column 5; Copied 7 characters

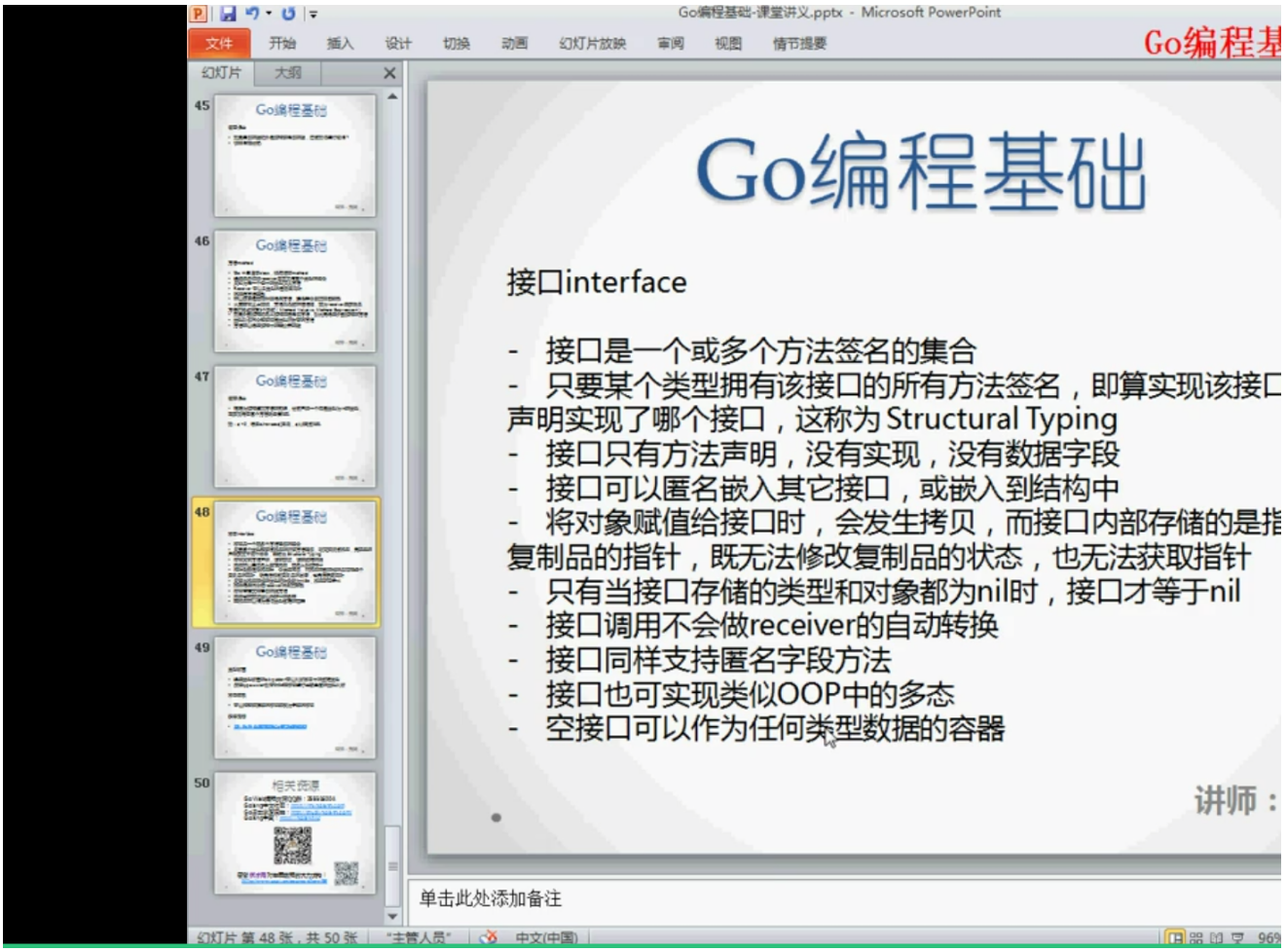

Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
文件
开始
插入
设计
切换
动画
幻灯片放映
审阅
视图
情节提要
幻灯片
大纲
51
52
53
54
55
56
Go编程基础
并发concurrency
- 很多人都是冲着 Go 大肆宣扬的高并发而忍不住跃跃欲试
源码的解析来看，goroutine 只是由官方实现的超级“线程池
不过话说回来，每个实例 4-5KB 的栈内存占用和由于实现机制
减少的创建和销毁开销，是制造 Go 号称的高并发的根本原因
goroutine 的简单易用，也在语言层面上给予了开发者巨大的
并发不是并行：Concurrency Is Not Parallelism
- 并发主要由切换时间片来实现“同时”运行，在并行则是直
多核实现多线程的运行，但 Go 可以设置使用核数，以发挥多
的能力。
Goroutine 奉行通过通信来共享内存，而不是共享内存来通信
讲师
单击此处添加备注
幻灯片 第 52 张，共 56 张
“主管人员”
中文(中国)

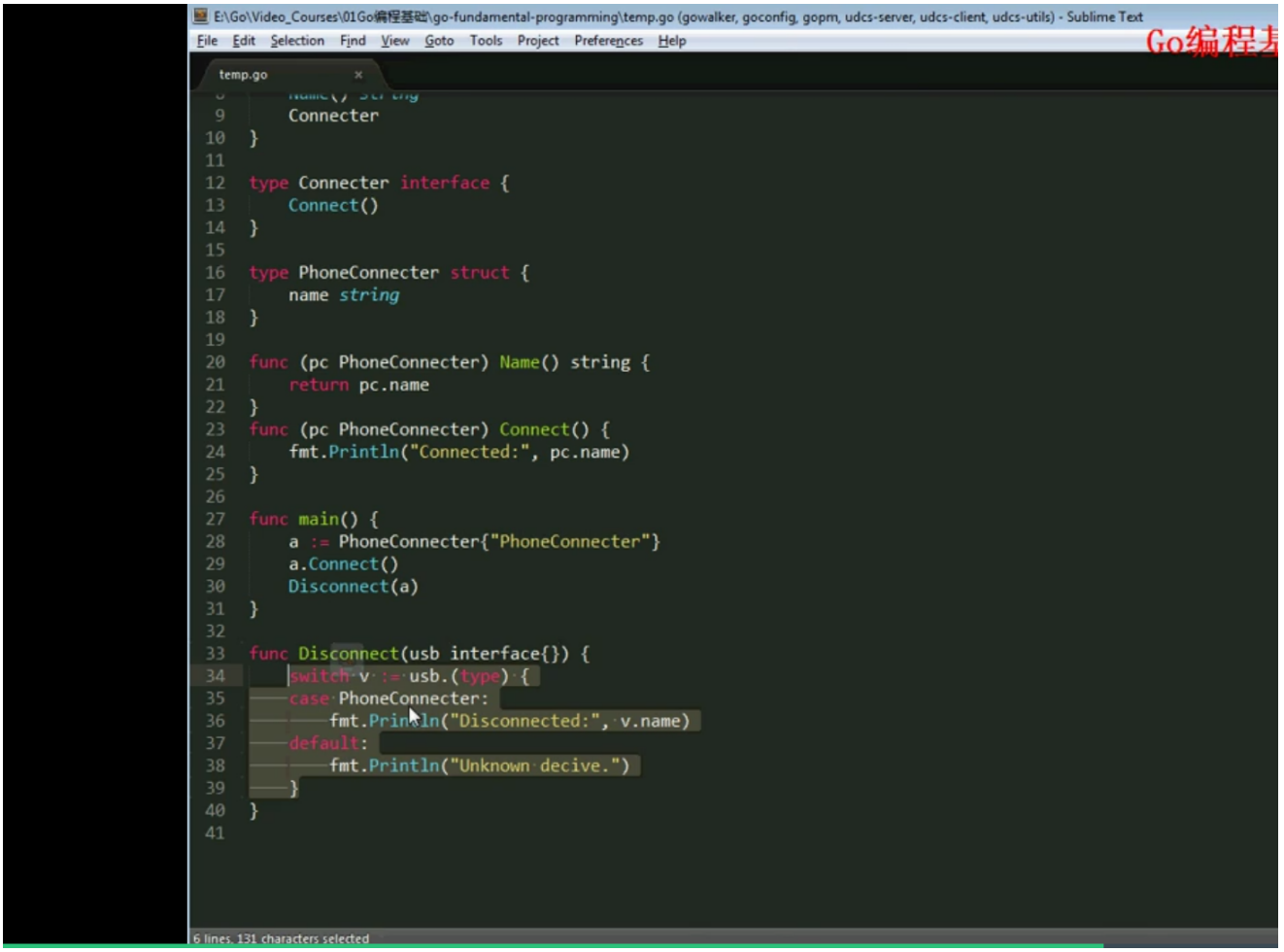

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5 "reflect"
6)
7
8 type User struct {
9 Id int
10 Name string
11 Age int
12 }
13
14 func (u User) Hello(name string) {
15 fmt.Println("Hello", name, ", my name is", u.Name)
16 }
17
18 func main() {
19 u := User{1, "OK", 12}
20 v := reflect.ValueOf(u)
21 mv := v.MethodByName("Hello")
22
23 args := []reflect.Value{reflect.ValueOf("joe")}
24 mv.Call(args)
25 }
26
Call: func(in []reflect.Value) []reflect.Value, Line 24, Column 17 - Field 1 of 2
Git Bash
$ go run temp.go
panic: reflect: call of reflect.Value.Elem
goroutine 1 [running]:
reflect.Value.Elem(0x4869e0, 0x7b, 0x20, 0x
D:/go/src/pkg/reflect/value.go:776
main.main()
E:/Go/Video_Courses/01Go编程基础/go
0x5b
goroutine 2 [runnable]:
exit status 2
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
999
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
{1 BYEBYE 12}
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
{1 OK 12}
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
BAD
{1 OK 12}
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
{1 BYEBYE 12}
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
# command-line-arguments
E:\Go\Video_Courses\01Go\go-fundamen
orted and not used: "reflect"
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
Hello joe , my name is OK
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$ go run temp.go
Hello joe , my name is OK
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01Go
master)
$

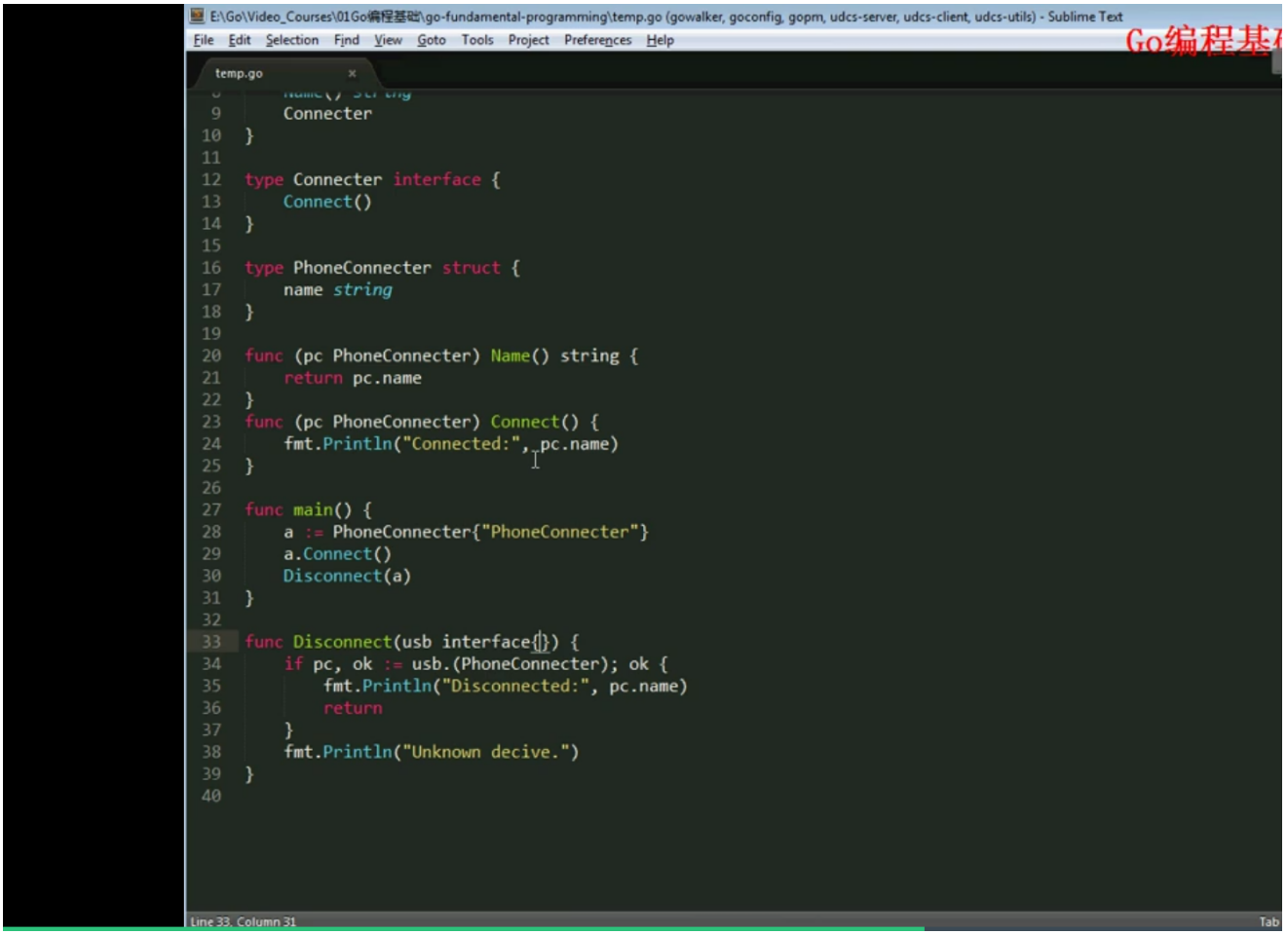

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5 "reflect"
6)
7
8 func main() {
9 x := 123
10 v := reflect.ValueOf(&x)
11 v.Elem().SetInt(999)
12
13 fmt.Println(x)
14 }
15
ValueOf: func(i interface{}) reflect.Value, Line 10, Column 27; Saved E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (UTF-8)

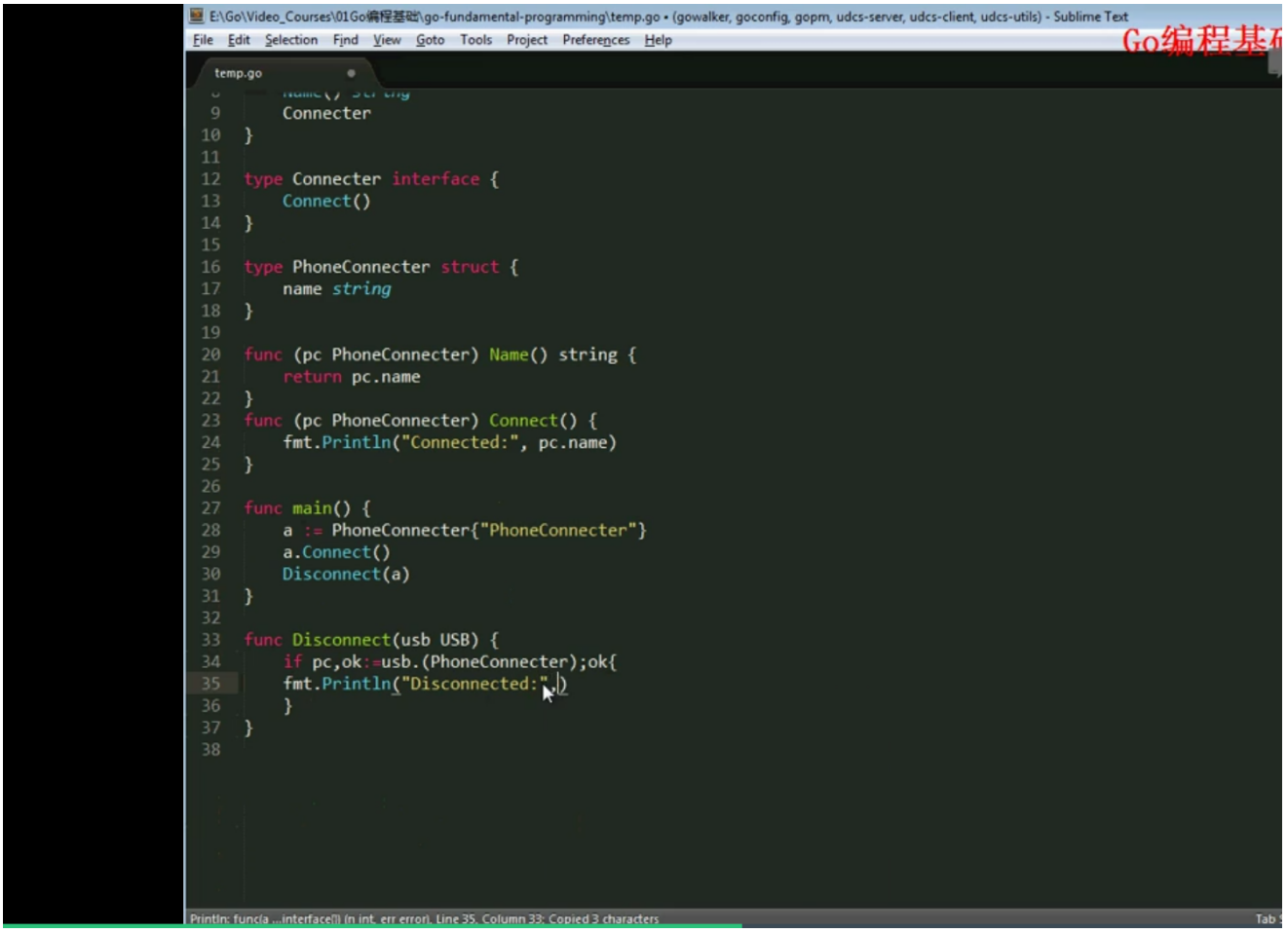

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程
temp.go
1 package main
2
3 import (
4 "fmt"
5 "reflect"
6)
7
8 type User struct {
9 Id int
10 Name string
11 Age int
12 }
13
14 type Manager struct {
15 User
16 title string
17 }
18
19 func main() {
20 m := Manager{User: User{1, "OK", 12}, title: "123"}
21 t := reflect.TypeOf(m)
22
23 fmt.Printf("%#v\n", t.FieldByIndex([]int{0, 1}))
24 }
25
Git Bash
E:/Go/Video_Courses/01Go编程基础/g
0x6f
goroutine 2 [runnable]:
exit status 2
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
# command-line-arguments
E:\Go\Video_Courses\01Go□□□□□□□□\go-fundame
orted and not used: "fmt"
E:\Go\Video_Courses\01Go□□□□□□□□\go-fundame
orted and not used: "reflect"
E:\Go\Video_Courses\01Go□□□□□□□□\go-fundame
defined: Info
E:\Go\Video_Courses\01Go□□□□□□□□\go-fundame
defined: u
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
{{1 OK 12} 123}
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
reflect.StructField{Name:"User", PkgPath:"
g:"", Offset:0x0, Index:[]int{0}, Anonymou
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
reflect.StructField{Name:"title", PkgPath:
), Tag:"", Offset:0x20, Index:[]int{1}, An
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
# command-line-arguments
E:\Go\Video_Courses\01Go□□□□□□□□\go-fundame
ntax error: unexpected semicolon or newlin
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
reflect.StructField{Name:"Id", PkgPath:"",
"", Offset:0x0, Index:[]int{0}, Anonymous:
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$ go run temp.go
reflect.StructField{Name:"Name", PkgPath:"
g:"", Offset:0x8, Index:[]int{1}, Anonymou
Unknown@UNKNOWN-PC /e/Go/Video_Courses/01G
master)
$
FieldByIndex func(index []int) reflect.StructField, Line 23, Column 50

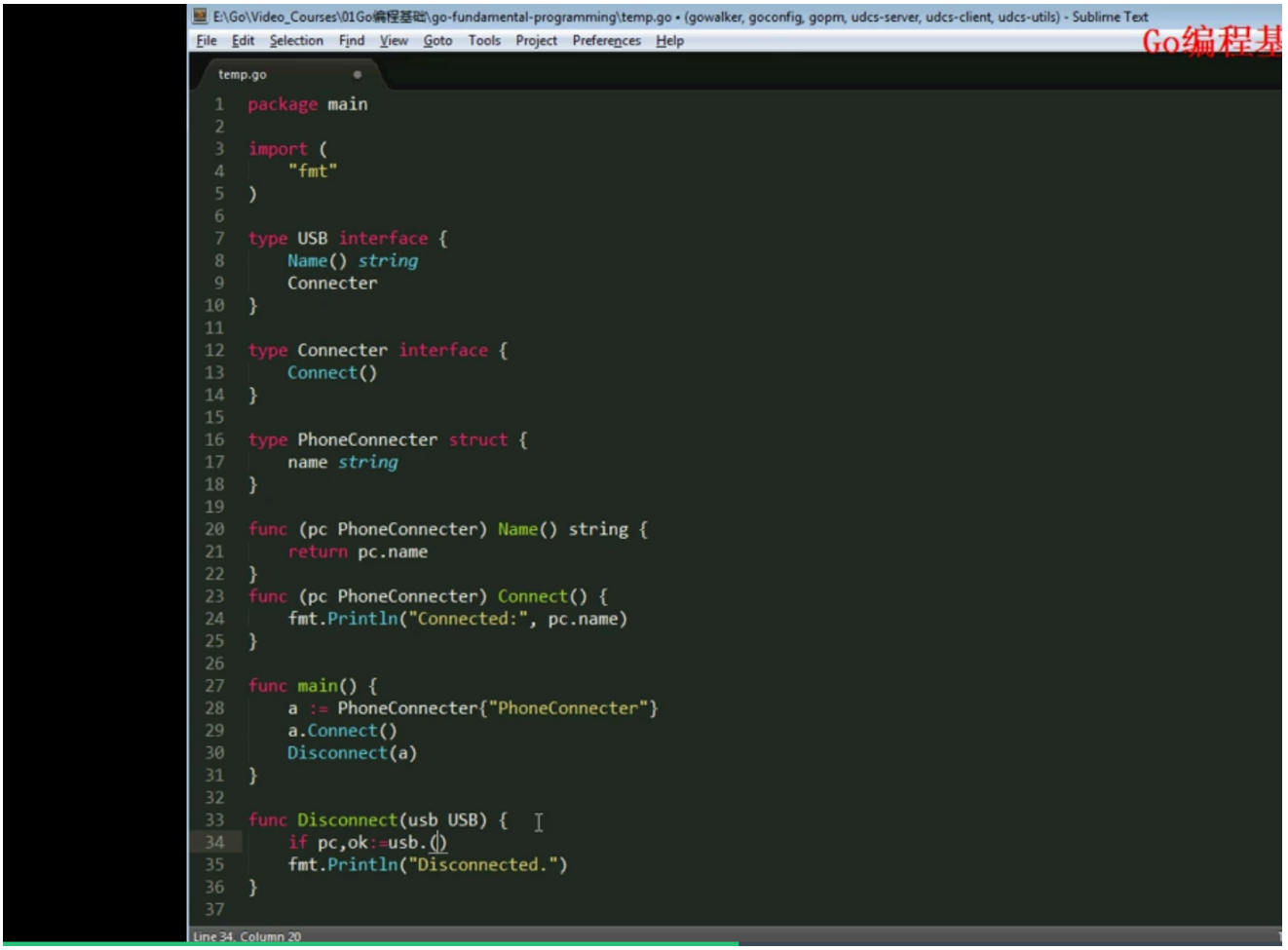

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程
temp.go
1 package main
2
3 import (
4 "fmt"
5 "reflect"
6)
7
8 type User struct {
9 Id int
10 Name string
11 Age int
12 }
13
14 type Manager struct {
15 User
16 title string
17 }
18
19 func main() {
20 m := Manager{User: User{1, "OK", 12}, title: "123"}
21 t := reflect.TypeOf(m)
22
23 fmt.Printf("%#v\n", t.Field(1))
24 }
25
Printf: func(format string, a ...interface{}) (n int, err error), Line 23, Column 36

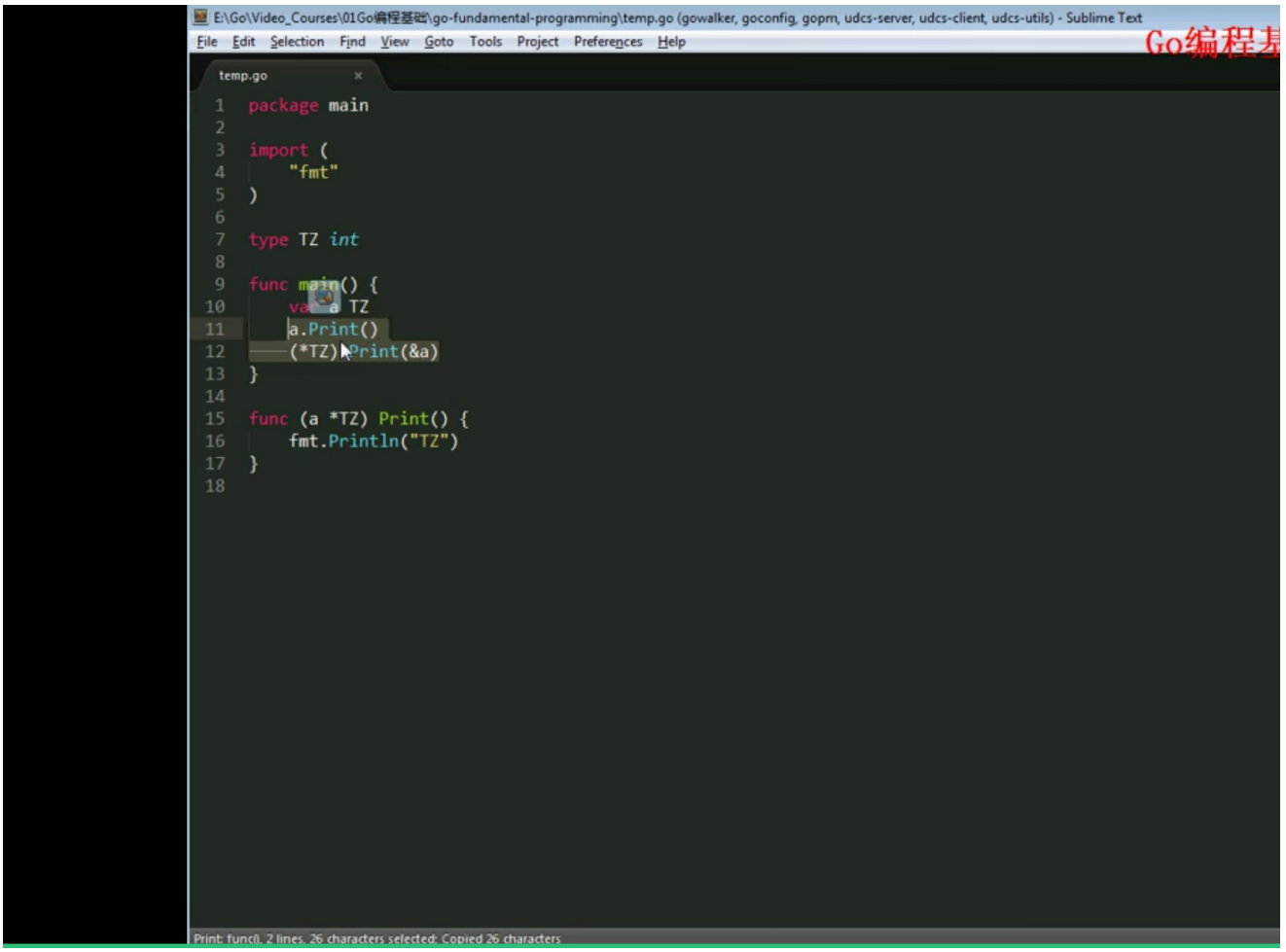

Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
文件 开始 插入 设计 切换 动画 幻灯片放映 审阅 视图 情节提要
幻灯片 大纲
Go编程基础
接口interface
- 接口是一个或多个方法签名的集合
- 只要某个类型拥有该接口的所有方法签名，即算实现该接口
声明实现了哪个接口，这称为 Structural Typing
- 接口只有方法声明，没有实现，没有数据字段
- 接口可以匿名嵌入其它接口，或嵌入到结构中
- 将对象赋值给接口时，会发生拷贝，而接口内部存储的是指
复制品的指针，既无法修改复制品的状态，也无法获取指针
- 只有当接口存储的类型和对象都为nil时，接口才等于nil
- 接口调用不会做receiver的自动转换
- 接口同样支持匿名字段方法
- 接口也可实现类似OOP中的多态
- 空接口可以作为任何类型数据的容器
讲师：
单击此处添加备注
幻灯片 第 48 张，共 50 张
中文(中国)

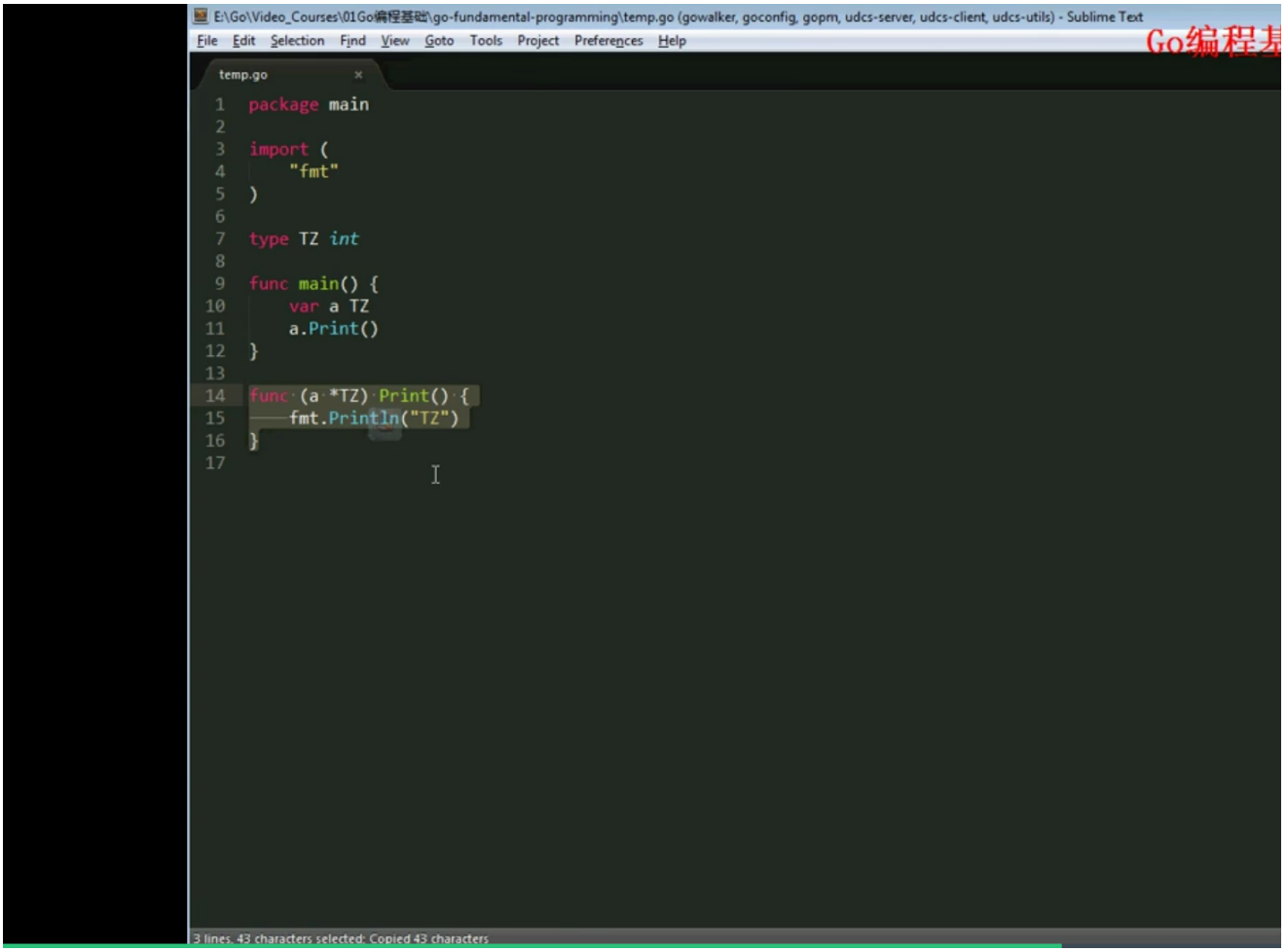

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
Connecter
}
type Connecter interface {
Connect()
}
type PhoneConnecter struct {
name string
}
func (pc PhoneConnecter) Name() string {
return pc.name
}
func (pc PhoneConnecter) Connect() {
fmt.Println("Connected:", pc.name)
}
func main() {
a := PhoneConnecter{"PhoneConnecter"}
a.Connect()
Disconnect(a)
}
func Disconnect(usb interface{}) {
switch v := usb.(type) {
case PhoneConnecter:
fmt.Println("Disconnected:", v.name)
default:
fmt.Println("Unknown decive.")
}
}
6 lines, 131 characters selected

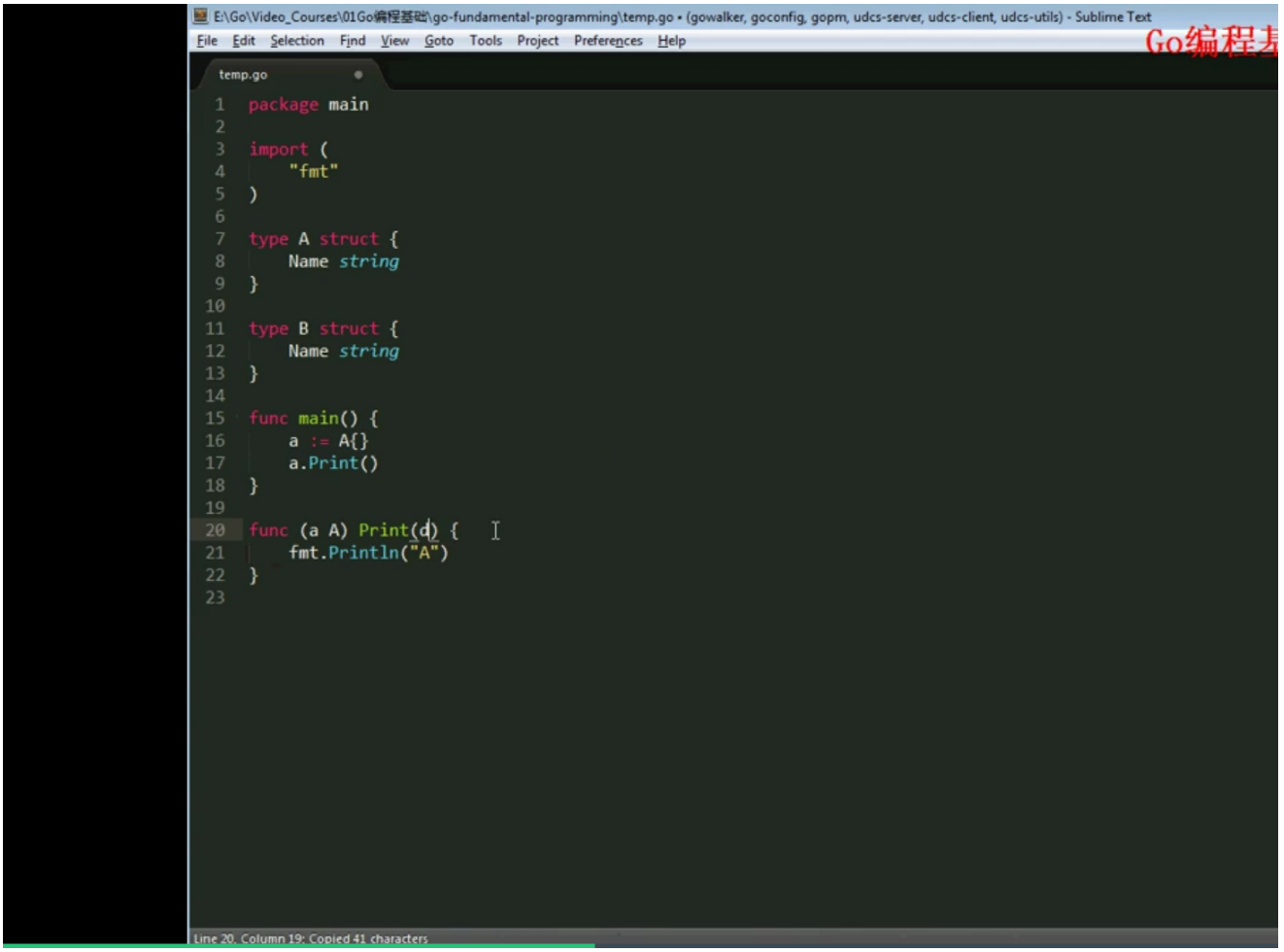

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
Connecter
}
type Connecter interface {
Connect()
}
type PhoneConnecter struct {
name string
}
func (pc PhoneConnecter) Name() string {
return pc.name
}
func (pc PhoneConnecter) Connect() {
fmt.Println("Connected:", pc.name)
}
func main() {
a := PhoneConnecter{"PhoneConnecter"}
a.Connect()
Disconnect(a)
}
func Disconnect(usb interface{}) {
if pc, ok := usb.(PhoneConnecter); ok {
fmt.Println("Disconnected:", pc.name)
return
}
fmt.Println("Unknown decive.")
}
Line 33, Column 31
Tab

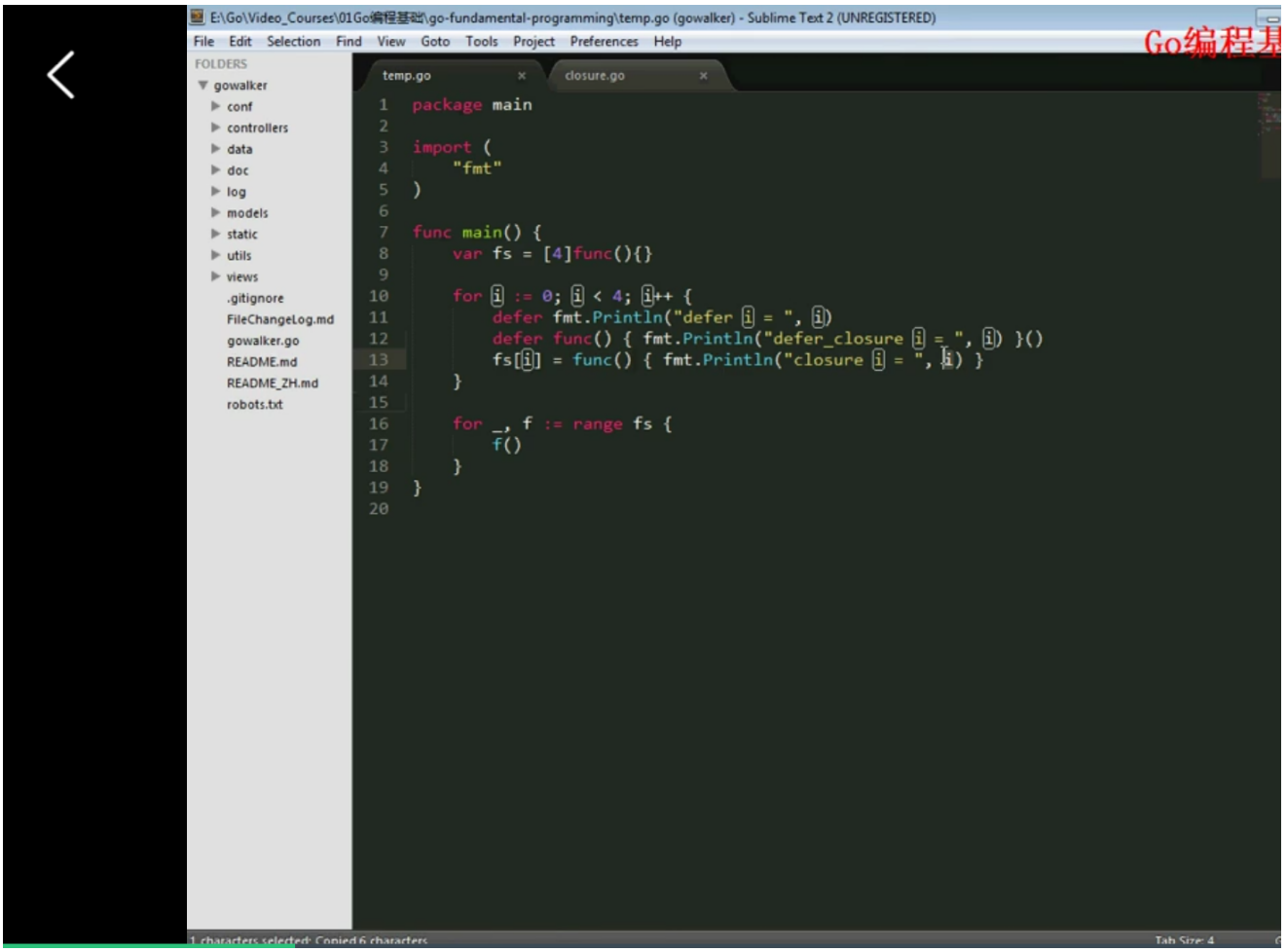

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
Connecter
}
type Connecter interface {
Connect()
}
type PhoneConnecter struct {
name string
}
func (pc PhoneConnecter) Name() string {
return pc.name
}
func (pc PhoneConnecter) Connect() {
fmt.Println("Connected:", pc.name)
}
func main() {
a := PhoneConnecter{"PhoneConnecter"}
a.Connect()
Disconnect(a)
}
func Disconnect(usb USB) {
if pc,ok:=usb.(PhoneConnecter);ok{
fmt.Println("Disconnected:")
}
}
Println: func(a ...interface{}) (n int, err error), Line 35, Column 33; Copied 3 characters
Tab

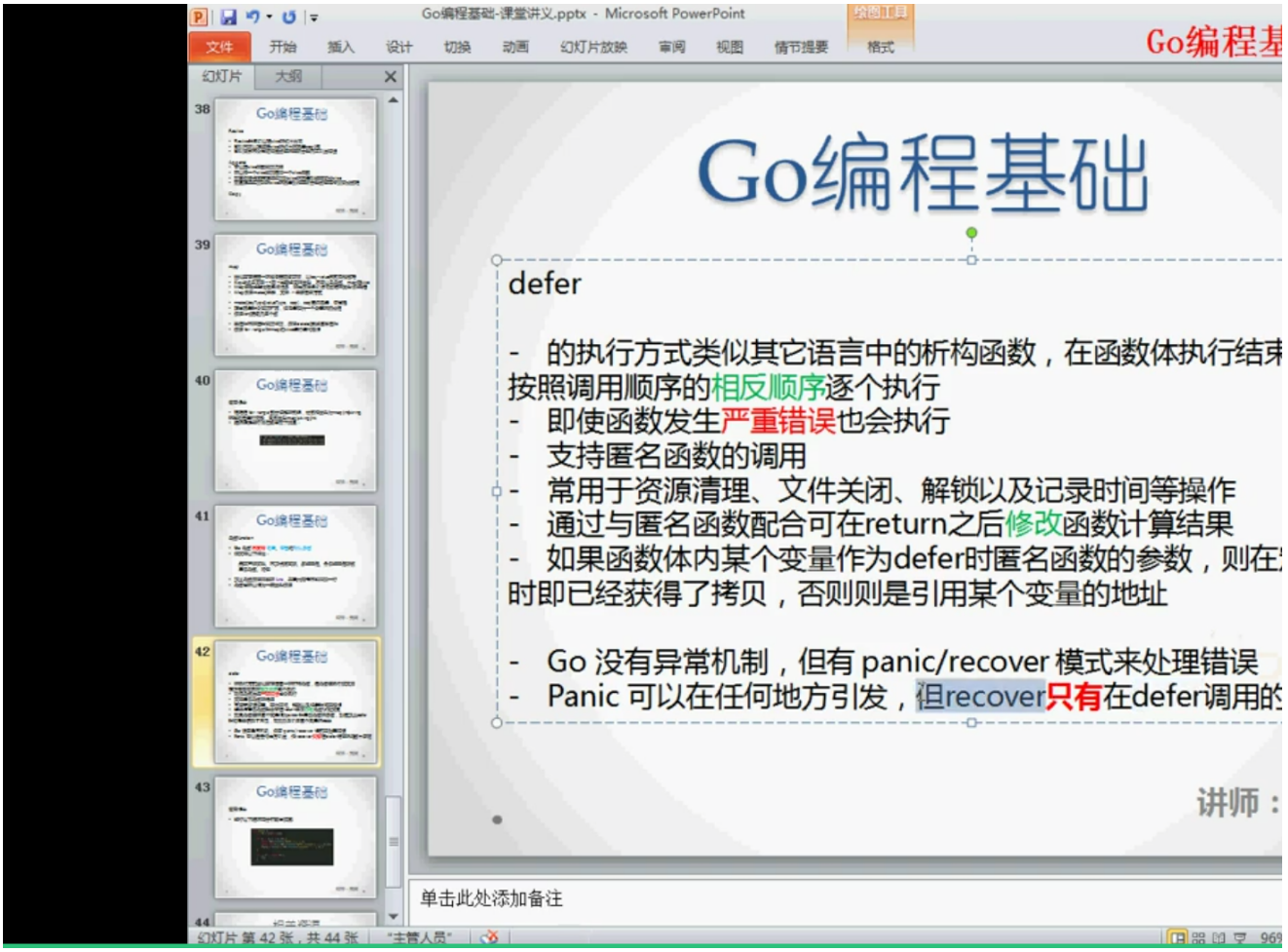

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 type USB interface {
8 Name() string
9 Connecter
10 }
11
12 type Connecter interface {
13 Connect()
14 }
15
16 type PhoneConnecter struct {
17 name string
18 }
19
20 func (pc PhoneConnecter) Name() string {
21 return pc.name
22 }
23 func (pc PhoneConnecter) Connect() {
24 fmt.Println("Connected:", pc.name)
25 }
26
27 func main() {
28 a := PhoneConnecter{"PhoneConnecter"}
29 a.Connect()
30 Disconnect(a)
31 }
32
33 func Disconnect(usb USB) {
34 if pc,ok:=usb.()
35 fmt.Println("Disconnected.")
36 }
37
Line 34, Column 20

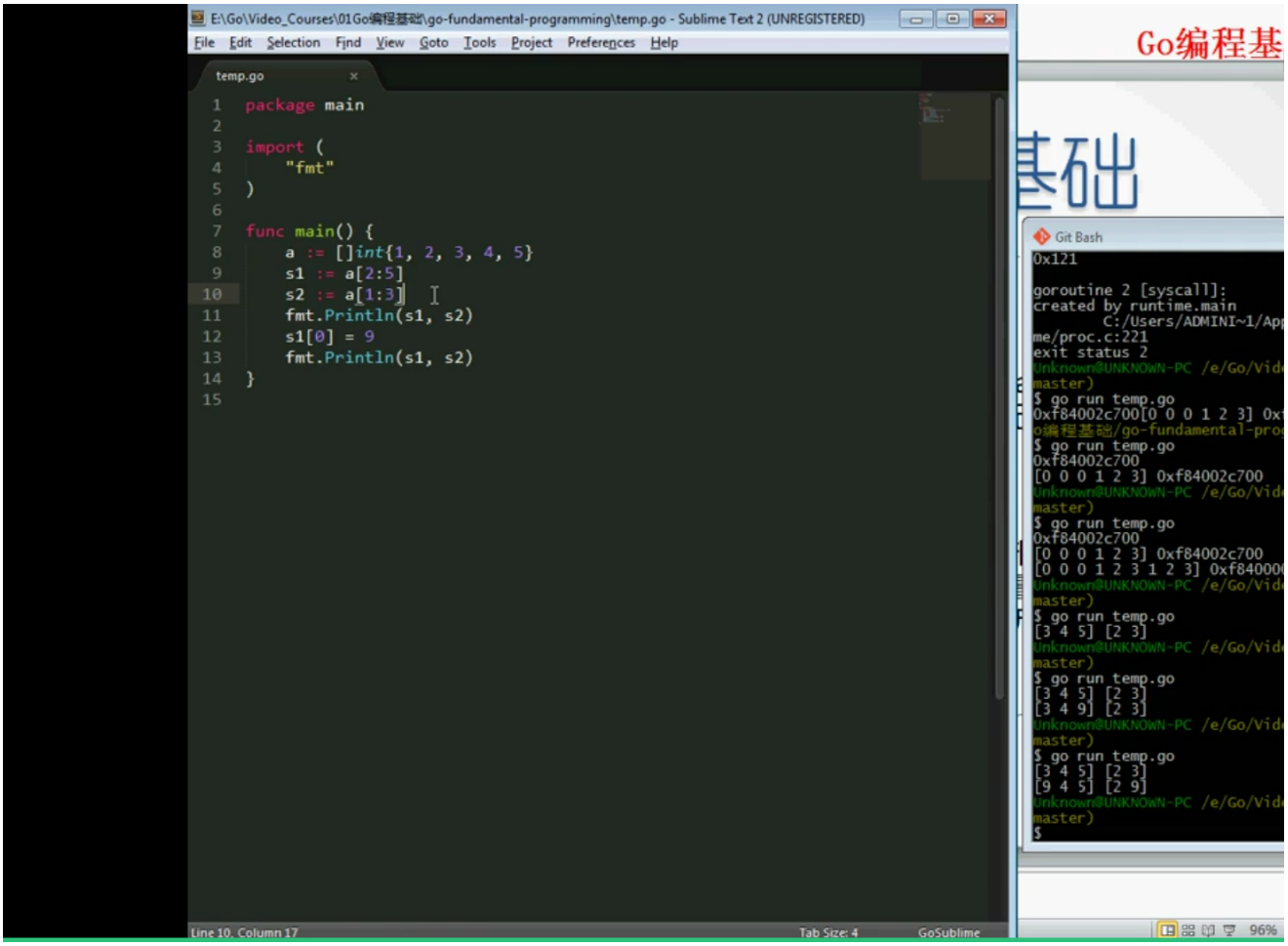

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 type TZ int
8
9 func main() {
10 var a TZ
11 a.Print()
12 (*TZ).Print(&a)
13 }
14
15 func (a *TZ) Print() {
16 fmt.Println("TZ")
17 }
18
Print: func(), 2 lines, 26 characters selected: Copied 26 characters

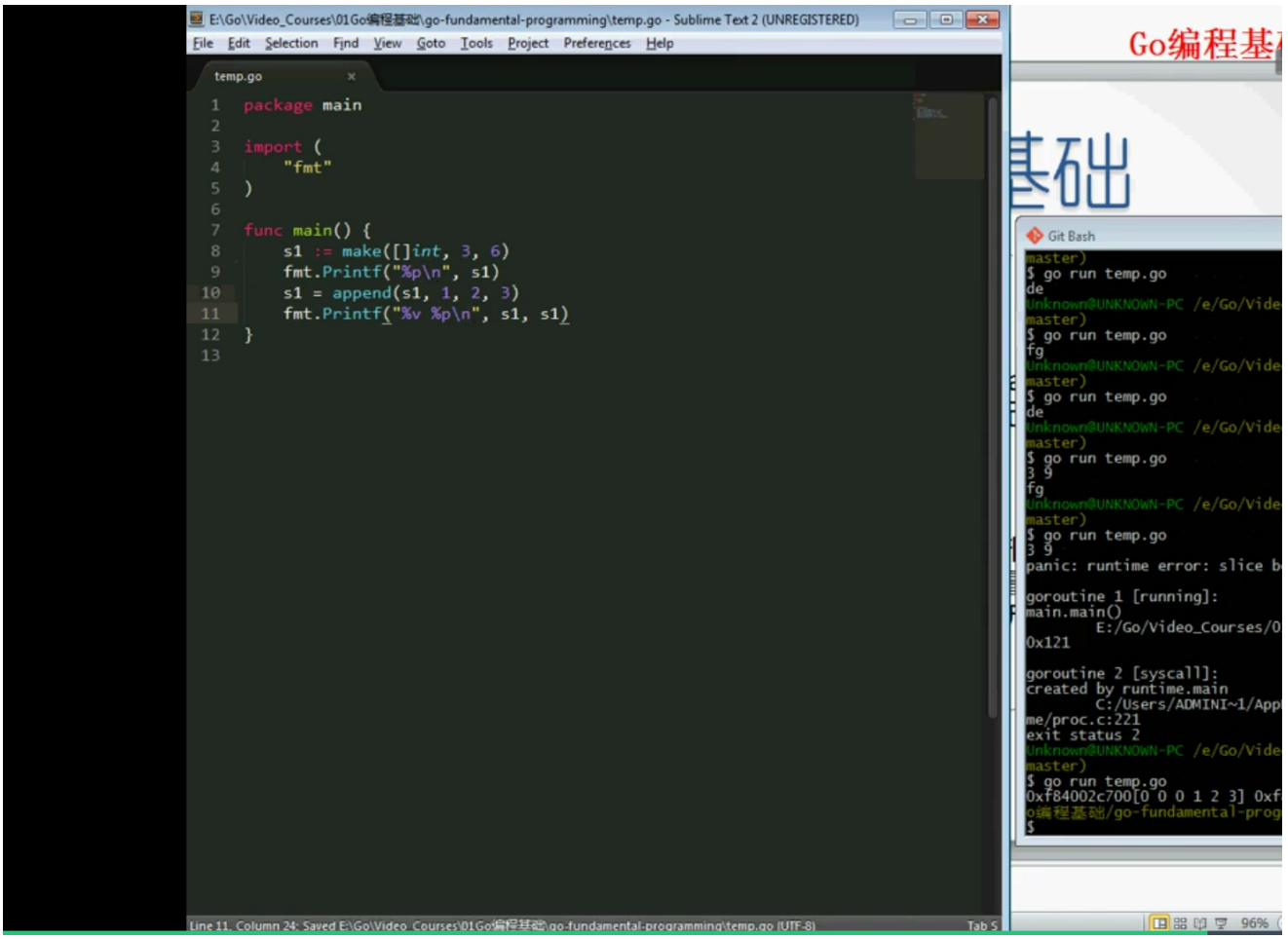

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 type TZ int
8
9 func main() {
10 var a TZ
11 a.Print()
12 }
13
14 func (a *TZ) Print() {
15 fmt.Println("TZ")
16 }
17
3 lines, 43 characters selected: Copied 43 characters

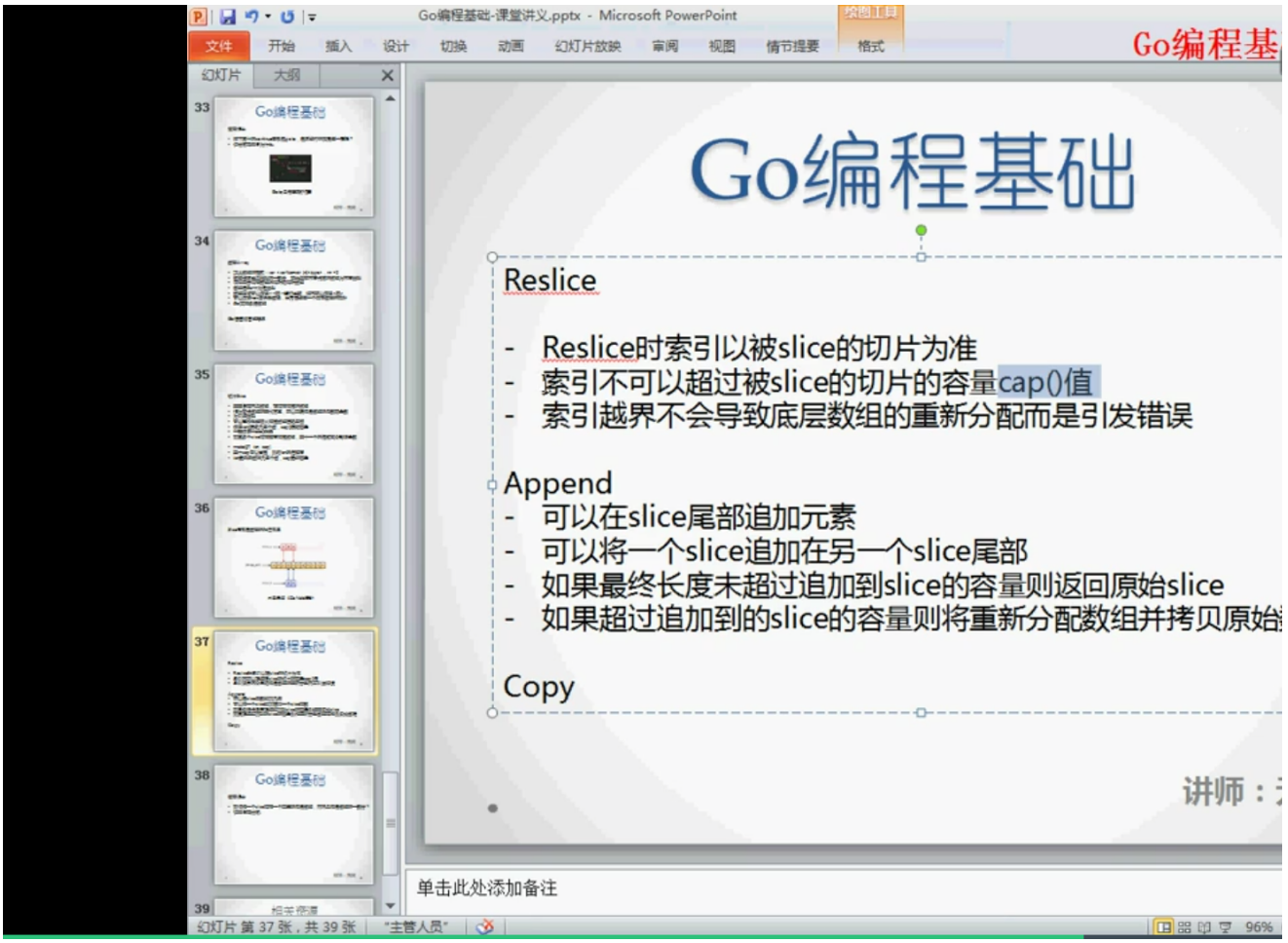

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • (gowalker, goconfig, gopm, udcs-server, udcs-client, udcs-utils) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 type A struct {
8 Name string
9 }
10
11 type B struct {
12 Name string
13 }
14
15 func main() {
16 a := A{}
17 a.Print()
18 }
19
20 func (a A) Print(d) {
21 fmt.Println("A")
22 }
23
Line 20, Column 19; Copied 41 characters


E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (gowalker) - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Go编程基
FOLDERS
gowalker
conf
controllers
data
doc
log
models
static
utils
views
.gitignore
FileChangeLog.md
gowalker.go
README.md
README_ZH.md
robots.txt
temp.go
closure.go
package main
import (
"fmt"
)
func main() {
var fs = [4]func(){}
for i := 0; i < 4; i++ {
defer fmt.Println("defer i = ", i)
defer func() { fmt.Println("defer_closure i = ", i) }()
fs[i] = func() { fmt.Println("closure i = ", i) }
}
for _, f := range fs {
f()
}
}

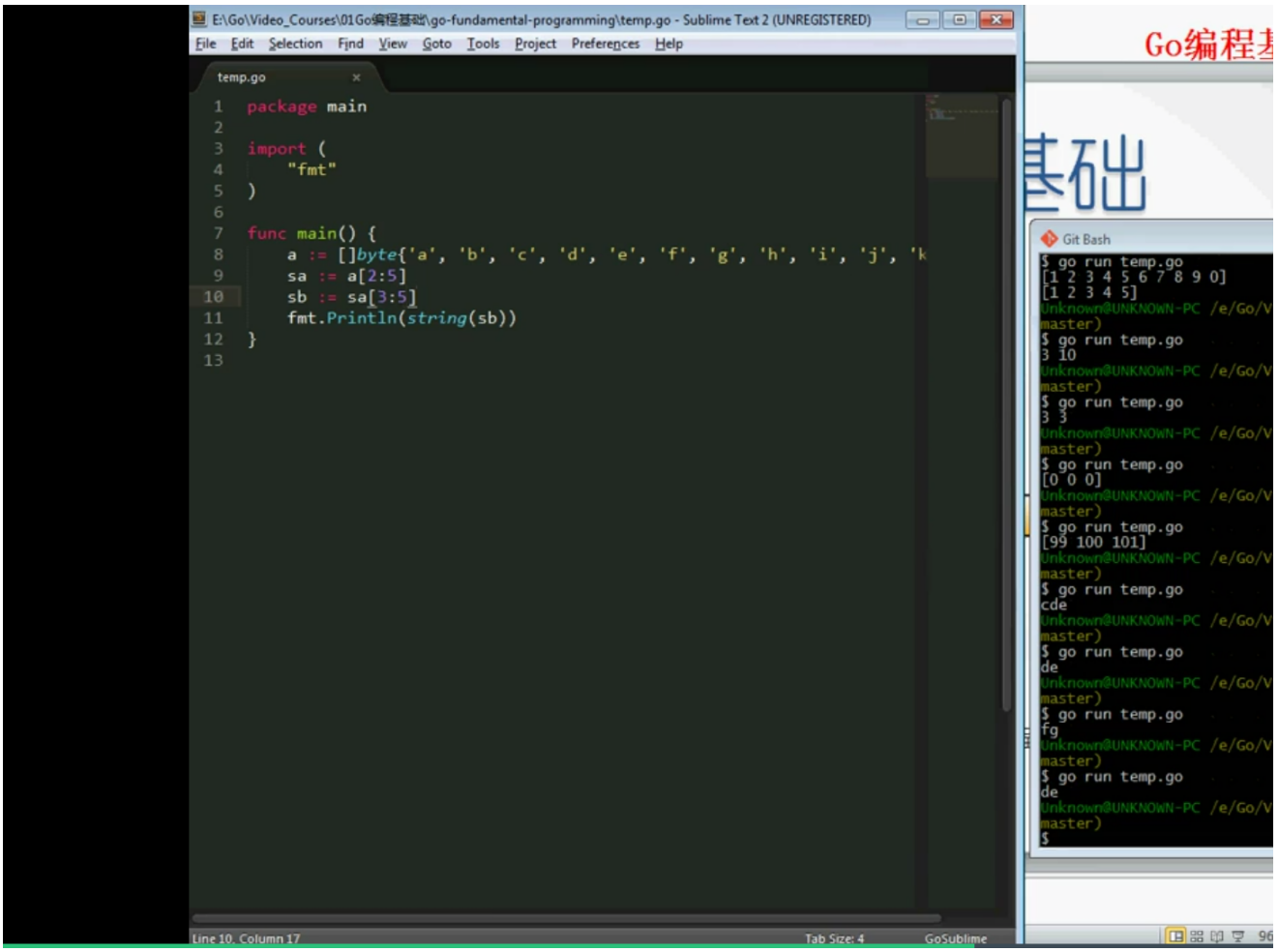

Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
Go编程基
文件 开始 插入 设计 切换 动画 幻灯片放映 审阅 视图 情节提要 格式
幻灯片 大纲
Go编程基础
defer
- 的执行方式类似其它语言中的析构函数，在函数体执行结束
按照调用顺序的相反顺序逐个执行
- 即使函数发生严重错误也会执行
- 支持匿名函数的调用
- 常用于资源清理、文件关闭、解锁以及记录时间等操作
- 通过与匿名函数配合可在return之后修改函数计算结果
- 如果函数体内某个变量作为defer时匿名函数的参数，则在
时即已经获得了拷贝，否则则是引用某个变量的地址
- Go 没有异常机制，但有 panic/recover 模式来处理错误
- Panic 可以在任何地方引发，但recover只有在defer调用的
讲师：
单击此处添加备注

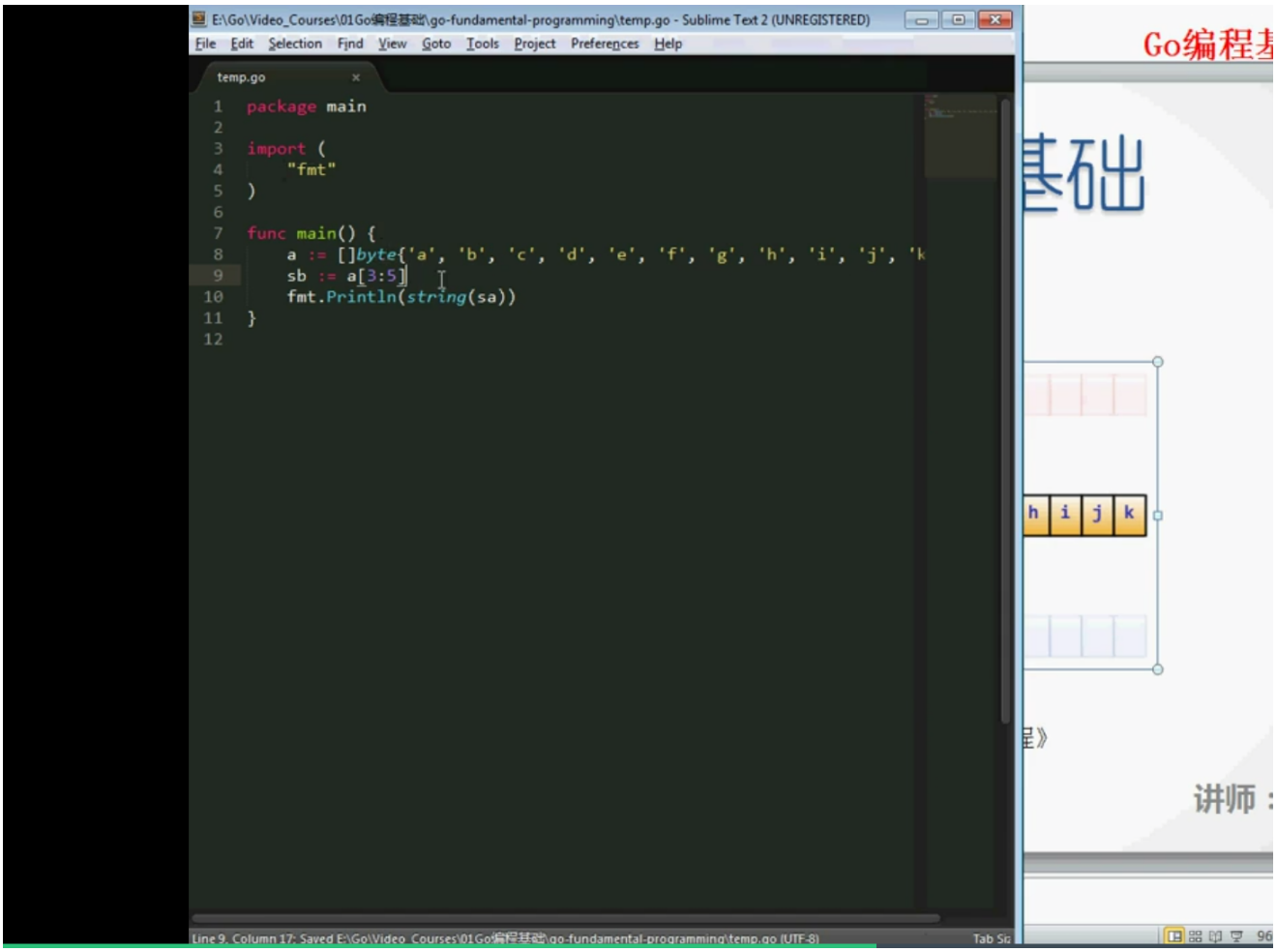

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 a := []int{1, 2, 3, 4, 5}
9 s1 := a[2:5]
10 s2 := a[1:3]
11 fmt.Println(s1, s2)
12 s1[0] = 9
13 fmt.Println(s1, s2)
14 }
15
Line 10, Column 17
Tab Size: 4
GoSublime
Go编程基
基础
Git Bash
0x121
goroutine 2 [syscall]:
created by runtime.main
C:/Users/ADMINI~1/App
me/proc.c:221
exit status 2
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
0xf84002c700[0 0 0 1 2 3] 0x
o编程基础/go-fundamental-proc
$ go run temp.go
0xf84002c700
[0 0 0 1 2 3] 0xf84002c700
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
0xf84002c700
[0 0 0 1 2 3] 0xf84002c700
[0 0 0 1 2 3 1 2 3] 0xf84000
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
[3 4 5] [2 3]
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
[3 4 5] [2 3]
[3 4 9] [2 3]
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
[3 4 5] [2 3]
[9 4 5] [2 9]
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$
96%

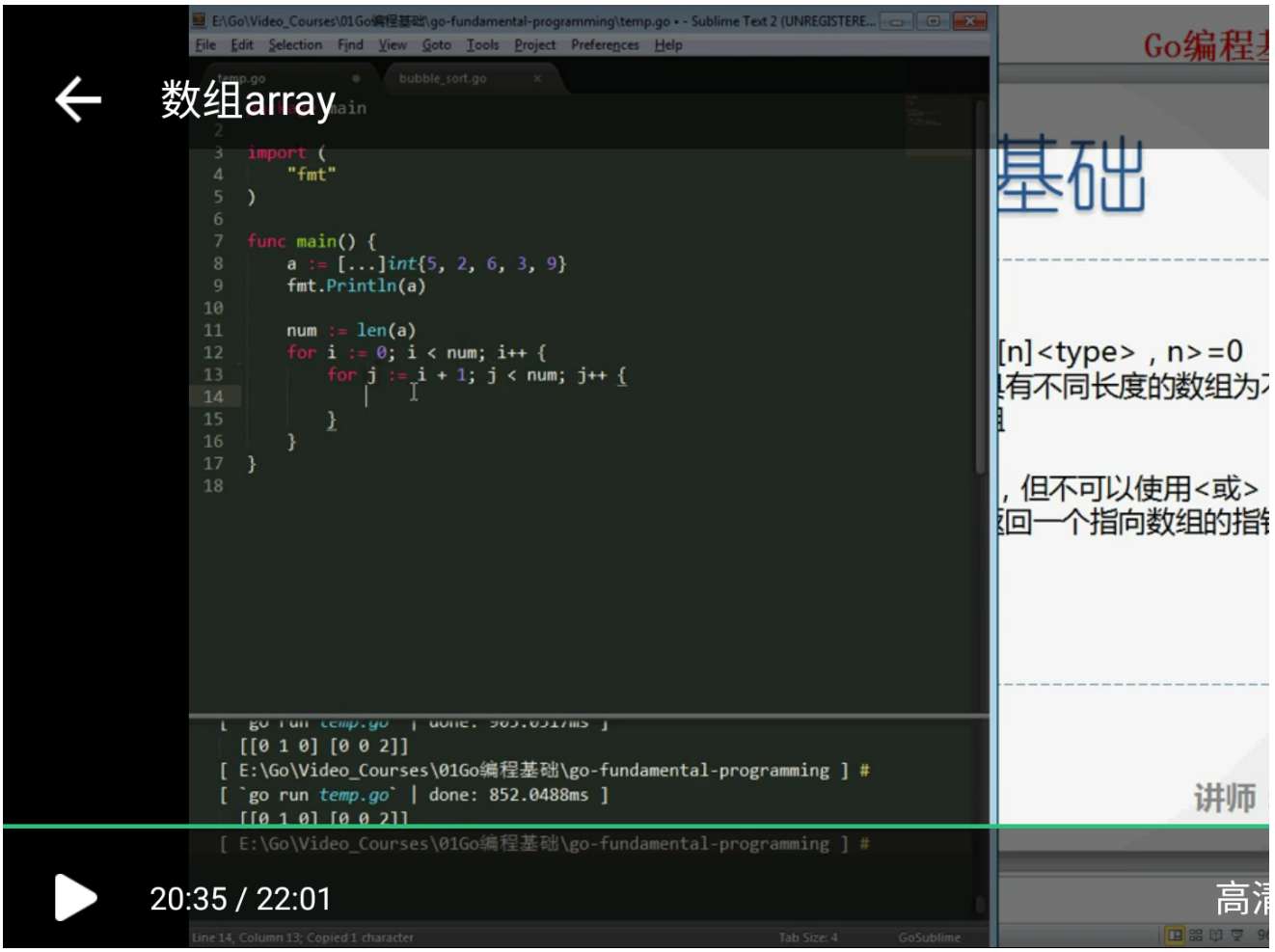

E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 s1 := make([]int, 3, 6)
9 fmt.Printf("%p\n", s1)
10 s1 = append(s1, 1, 2, 3)
11 fmt.Printf("%v %p\n", s1, s1)
12 }
13
Line 11, Column 24; Saved E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (UTF-8)
Tab Size: 4
Go编程基
基础
Git Bash
master)
$ go run temp.go
de
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
fg
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
de
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
3 9
fg
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
3 9
panic: runtime error: slice b
goroutine 1 [running]:
main.main()
E:/Go/Video_Courses/0
0x121
goroutine 2 [syscall]:
created by runtime.main
C:/Users/ADMINI~1/App
me/proc.c:221
exit status 2
Unknown@UNKNOWN-PC /e/Go/Vide
master)
$ go run temp.go
0xf84002c700[0 0 0 1 2 3] 0xf
o编程基础/go-fundamental-prog
$
96%


Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
文件 开始 插入 设计 切换 动画 幻灯片放映 审阅 视图 情节提要 格式
绘图工具
Go编程基
幻灯片 大纲
33
34
35
36
37
38
39
Go编程基础
Reslice
- Reslice时索引以被slice的切片为准
- 索引不可以超过被slice的切片的容量cap()值
- 索引越界不会导致底层数组的重新分配而是引发错误
Append
- 可以在slice尾部追加元素
- 可以将一个slice追加在另一个slice尾部
- 如果最终长度未超过追加到slice的容量则返回原始slice
- 如果超过追加到的slice的容量则将重新分配数组并拷贝原始
Copy
讲师：
单击此处添加备注
幻灯片 第 37 张，共 39 张
"主管人员"
96%


Go编程基础-课堂讲义.pptx - Microsoft PowerPoint
文件
开始
插入
设计
切换
动画
幻灯片放映
审阅
视图
情节提要
Go编程基
幻灯片
大纲
33
34
35
36
37
38
39
Go编程基础
Slice与底层数组的对应关系
Slice_a
c
d
e
Array_ori
a
b
c
d
e
f
g
h
i
j
k
Slice_b
d
e
本图来源《Go Web编程》
讲师：
单击此处添加备注
幻灯片 第 36 张，共 39 张
"主管人员"
中文(中国)


E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 a := []byte{'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k
9 sa := a[2:5]
10 sb := sa[3:5]
11 fmt.Println(string(sb))
12 }
13
Line 10, Column 17
Tab Size: 4
GoSublime
Go编程基
基础
Git Bash
$ go run temp.go
[1 2 3 4 5 6 7 8 9 0]
[1 2 3 4 5]
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
3 10
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
3 3
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
[0 0 0]
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
[99 100 101]
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
cde
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
de
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
fg
Unknown@UNKNOWN-PC /e/Go/V
master)
$ go run temp.go
de
Unknown@UNKNOWN-PC /e/Go/V
master)
$


E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
1 package main
2
3 import (
4 "fmt"
5)
6
7 func main() {
8 a := []byte{'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k
9 sb := a[3:5]
10 fmt.Println(string(sa))
11 }
12
Line 9, Column 17; Saved E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go (UTF-8)
Tab Siz
Go编程基
基础
h i j k
讲师：


数组array
E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming\temp.go • - Sublime Text 2 (UNREGISTERE...
File Edit Selection Find View Goto Tools Project Preferences Help
temp.go
bubble_sort.go
main
import (
"fmt"
)
func main() {
a := [...]int{5, 2, 6, 3, 9}
fmt.Println(a)
num := len(a)
for i := 0; i < num; i++ {
for j := i + 1; j < num; j++ {
}
}
}
[[0 1 0] [0 0 2]]
[E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming] #
[`go run temp.go` | done: 852.0488ms]
[[0 1 0] [0 0 2]]
[E:\Go\Video_Courses\01Go编程基础\go-fundamental-programming] #
Line 14, Column 13; Copied 1 character
Tab Size: 4
GoSublime
Go编程基
基础
[n]<type> , n>=0
，但不可以使用<或>
讲师
20:35 / 22:01
高清