بسم الله الرحمن الرحیم

تکنولوژی کامپیوتر

جلسه‌ی سوم
زبان گولنگ (۳)

جلسه‌ی گذشته

گولنگِ بیشتر

for حلقەی

- for i := 0; i < n; i++ {

  }
- for i < n {

  }
- for {

  }

if شرط

- if i < n {

  } else if i < 2 * n {

  } else {

  }
- if c, err := file.ReadFile(“main.go”); err == nil {

  } else {
  fmt.Println(err)
  }

swtich با شرط

- switch x {
  case 1:
  …
  case 2, 3:
  …
  default:
  …
  }

- switch {
  case condition 1:
  …
  case condition 2:
  …
  default:
  …
  }

defer

- f, _ := os.Open(“main.go”)
  defer f.Close()

pointers

```
var p *int
i := 42
p = &i
*p = 43
fmt.Println(i)
```

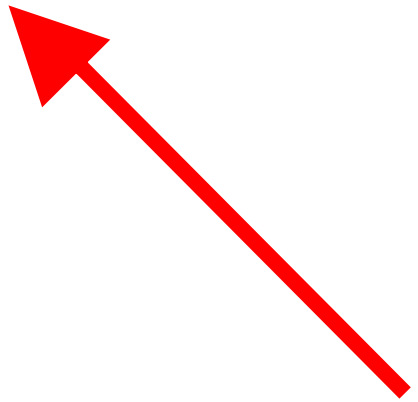
type

```
type MyInt int
var a MyInt = 5
fmt.Println(a)
```

type

```
type Weekday int

const (
    Sunday Weekday = iota
    Monday
    Tuesday
    Wednesday
    Thursday
    Friday
    Saturday
)
```



struct

```
struct {
    A int
    B int
    C string
}
```

Struct…

```
var p struct {
    A int
    B int
} = struct {
    A int
    B int
}{10, 20}
fmt.Printf("%+v", p)
```

```
Output:
{A:10 B:20}
```

Struct with type

```
type Pair struct {
    A int
    B int
}
var p Pair = Pair{10, 20}
fmt.Printf("%+v", p)
```

```
Output:
{A:10 B:20}
```

Struct initialization

```
type Vertex struct {
    X, Y int
}

var (
    v1 = Vertex{1, 2}  // has type Vertex
    v2 = Vertex{X: 1}  // Y:0 is implicit
    v3 = Vertex{}      // X:0 and Y:0
    p  = &Vertex{1, 2} // has type *Vertex
)
```

Constructor???

```
type Vertex struct {
    X, Y int
}

func NewVertex(x, y int) *Vertex {
    return &Vertex{
        X: x,
        Y: y
    }
}
```

جلسه‌ی جدید

ادامه‌ی گولنگ...

ادامه‌ی چیزهایی شبیه به شیء گرایی

حالا که کلاس نداریم، استراکت داریم، متود چطور؟؟

- function with receivers…

Choosing a value or pointer receiver

- There are two reasons to use a pointer receiver.
  - *The first is so that the method can modify the value that its receiver points to.*
  - *The second is to avoid copying the value on each method call. This can be more efficient if the receiver is a large struct, for example.*
- In general, all methods on a given type should have either value or pointer receivers, but not a mixture of both.

Struct embeding

Interface?

- از جاوا، اینترفیس توش یه سری تعریف توابع میومد و هرکی ایمپلمنتش می‌کرد، باید اون توابع رو می‌داشت.
- فرایندی که در جاوا داشتیم:
  - *اینترفیس رو تعریف کنیم*
  - *توی کلاسمون بگیم که اون اینترفیس رو داریم پیاده می‌کنیم.*
  - *متودهای اینترفیس، عینا در کلاسمون بیاد.*

Interface

- در گولنگ
  - *اینترفیس رو تعریف کنیم*
  - ~~*توی کلاسمون بگیم که اون اینترفیس رو داریم پیاده می‌کنیم.*~~
  - *متودهای اینترفیس، عینا در استراکتمون بیاد.*

Array

Slice