

DD1339 Introduktion till datalogi

Namn: *Hampus Fristedt*

Uppgift: VT7

Grupp nummer: 1

Övningsledare: Peter Boström

Betyg: Datum: Rättad av:

1 Kod

../LoopsAndFunctions.go

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func Sqrt(x float64) float64 {
9     z := 1.0
10    for tmp := 0.0; math.Abs(tmp - z) > 0.00000001; {
11        tmp = z
12        z = z - (z * z - x) / (2 * z)
13    }
14    return z
15 }
16
17 func main() {
18     fmt.Println(Sqrt(2))
19 }

```

../Slice.go

```

1 package main
2
3 import (
4     "code.google.com/p/go-tour/pic"
5 )
6
7 func Pic(dx, dy int) [][]uint8 {
8     pic := make([][]uint8, dy)
9     for i := range pic {
10        pic[i] = make([]uint8, dx)
11
12        for j := range pic[i] {
13            pic[i][j] = uint8((i+j)/2)
14        }
15    }
16
17    return pic
18 }
19
20 func main() {
21     pic.Show(Pic)
22 }

```

../WC.go

```

1 package main
2
3 import (
4     "code.google.com/p/go-tour/wc"
5     "strings"
6 )
7

```

```

8 func WordCount(s string) map[string]int {
9     mappy := make(map[string]int)
10    words := strings.Fields(s)
11    for _, word := range words {
12        mappy[word] += 1
13    }
14    return mappy
15 }
16
17 func main() {
18     wc.Test(WordCount)
19 }

```

../FibonacciClosure.go

```

1 package main
2
3 import "fmt"
4
5 // fibonacci is a function that returns
6 // a function that returns an int.
7 func fibonacci() func() int {
8     i, j := 0, 1
9     return func() int {
10         fib := i + j
11         i = j
12         j = fib
13         return fib
14     }
15 }
16
17 func main() {
18     f := fibonacci()
19     for i := 0; i < 10; i++ {
20         fmt.Println(f())
21     }
22 }

```

../Clock.go

```

1 package main
2
3 import (
4     "fmt"
5     "time"
6 )
7
8 func Remind(text string, paus time.Duration) {
9     for {
10         fmt.Println("Klockan är " + time.Now().Format("15:04:05") + ":
11             " + text)
12         time.Sleep(paus)
13     }
14 }
15
16 func main() {
17     go Remind("Dags att äta", time.Hour*3)

```

1 KOD

```
17     go Remind("Dags att arbeta", time.Hour*8)
18     Remind("Dags att sova", time.Hour*24)
19 }
```

../Sum.go

```
1 package main
2
3 import "fmt"
4
5 // Add adds the numbers in a and sends the result on res.
6 func Add(a []int, res chan<- int) {
7     sum := 0
8     for _, value := range a {
9         sum += value
10    }
11    res <- sum
12 }
13
14 func main() {
15     a := []int{1, 2, 3, 4, 5, 6, 7}
16
17     n := len(a)
18     ch := make(chan int)
19     go Add(a[:n/2], ch)
20     go Add(a[n/2:], ch)
21
22     // TODO: Get the subtotals from the channel and print their sum.
23     sum1 := <-ch
24     sum2 := <-ch
25     fmt.Printf("%d %d %d \n", sum1, sum2, sum1 + sum2)
26 }
```