UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Paralelni i distribuirani algoritmi i strukture podataka

ms Nebojša Horvat

Zimski semestar 2019/2020.

Studijski program: Računarstvo i automatika

Modul: Računarstvo visokih performansi

Go (Golang)

Funkcije - nastavak

- Funkcija može da vrati proizvoljan broj vrednosti
- Import math/cmplx

```
func findArgAndModulusOfZ(z complex128) (float64, float64) {
    return cmplx.Phase(z), cmplx.Abs(z)
}

func main() {
    var z complex128 = 2 + 2i
    var arg, modul = findArgAndModulusOfZ(z)
    fmt.Printf("%f, %f", arg, modul)
}
```

Napomene

- Ne postoji null vrednost, umesto toga je nil
- Promenljive koje su deklarisane se moraju koristiti
  - Rezultovaće greškom ako se ne koriste
- Uvučeni (import-ovani) paketi se moraju koristiti
  - Rezultovaće greškom ako se ne koriste
- Ne postoji ; na kraju iskaza
  - Za razliku od C, C++, C#, Java
- GOPATH
- Vežbanje online
  - https://tour.golang.org/

Argumenti komandne linije

- Slično kao u drugim programskim jezicima

go run test.go fakultet tehnickih nauka

```
var args []string = os.Args
fmt.Println(len(args))
for _, arg := range args {
    fmt.Println(arg)
}
```

- Program će ispisati kao prvi argument putanju do go fajla, a zatim,
  - Fakultet
  - Tehnickih
  - Nauka

Rad sa fajlovima

- Primer čitanje

```
file, err = os.OpenFile(filePath, os.O_RDONLY, os.ModeExclusive)
scanner := bufio.NewScanner(file)
for scanner.Scan() {
    fmt.Println(scanner.Text())
}
if err := scanner.Err(); err != nil {
    log.Fatal(err)
}
defer file.Close()
return
```

Defer

- Ključna reč koja odlaže izvršavanje naredbe
  - Dok se ne završi funkcija u kojoj se nalazi Defer naredba

```
func main() {
    defer fmt.Println("world")

    fmt.Println("hello")
}
```

Rad sa fajlovima

- Primer pisanje

```
os.Create(filePath) // Može se otvoriti fajl u režimu koji ga kreira ako ne postoji
    file, err := os.OpenFile(filePath, os.O_RDWR, os.ModeAppend)
    if err != nil {
        log.Printf(“can’t open file”)
      }
    writer := bufio.NewWriter(file)
      charsWritten, err := writer.WriteString("1. row\n")
    if err != nil {
        log.Fatal(err)
    }
    err = writer.Flush()
    defer file.Close()  //what is defer?
```

Rad sa fajlovima

- Načini otvaranja fajla za pisanje
  - // os.O_WRONLY tells the computer you are only going to writo to the file, not read
  - // os.O_CREATE tells the computer to create the file if it doesn't exist
  - // os.O_APPEND tells the computer to append to the end of the file instead of overwritting or truncating it

- Primer upotrebe:
  - f, err := os.OpenFile(“studenti.txt", os.O_WRONLY|os.O_CREATE| os.O_APPEND, 0644)
- Postoje i drugi modovi za čitanje fajla, pogledati dokumentaciju

Strukture

- Slično kao u jeziku C
- Koriste se za modelovanje stanja objekta

```
type Person struct {
    firstName string
    lastName string
    balance float64
    personID string
}
```

Zadatak - Strukture

- Napisati funkciju koja prima parametar instancu strukture Trougao
- Funkcija vraća float64 vrednost koja predstavlja površinu datog trougla
- Za računanje površine trougla koristiti Heronov obrazac
  - s = (a+b+c)/2
  - P = Sqrt(s*(s-a)*(s-b)*(s-c))
- Paket math -> Sqrt

Pokazivači

- Tip podatka koji kao vrednost drži adresu
- Upotreba kao u jeziku C

```
func pointersExample() {
    i, j := 42, 32.4
    p := &i // point to i
    fmt.Println(*p) // read i through the pointer
    *p = 21 // set value of the i
    fmt.Println(*p) // write value of the variable on which the p is pointing
    fmt.Println(i) // write value of the variable i - > same as above
    fmt.Println(j) // is this line necessary ?
}
```

Složeni tipovi podataka

- Mape
- Nizovi
- Slajsevi (slices)
- Drugi tipovi kolekcija

Statički nizovi

- Kao u drugim jezicima
- Fiksne veličine

```
var names [2]string
    names[0] = "Marc"
    names[1] = "John "

    fmt.Println(names[0], names[1])
    fmt.Println(names)
```

Dinamički nizovi

- Statički nizovi nisu uvek pogodni jer se veličina mora znati unapred
  - Uvodi se pojam slajsa, koji je dinamički alociran niz
    - Slično kao dinamičke strukture podataka npr. liste u drugim programskim jezicima
  - numberSlice:= make([]int, 5)
    - Alocirano 5 elemenata, sa mogućnošću proširivanja

Dinamički nizovi – funkcije len, cap

- The length of a slice is the number of elements it contains.
  - len(s)
- The capacity of a slice is the number of elements in the underlying array, counting from the first element in the slice.
  - cap(s)

```
s := make([]int, 0, 5) // len(s)=0, cap(s)=5
s = s[:cap(s)] // len(s)=5, cap(s)=5
s = s[1:] // len(s)=4, cap(s)=4
```

Slice - primer

```
primes := [8]int{2, 3, 5, 7, 11, 13, 17, 19}
primesLessThan10 := primes[0:4] //first is includes, last one is excluded

fmt.Println(primesLessThan10)

primesLessThan10 = primes[0:8] //slice is dynamically-sized

fmt.Println(primesLessThan10)

primesLessThan10[0] = 13     //primes is also changed, because slice is acutally reference to the array

fmt.Println(primes)
newSlice := make([]int, 5) // dinamically create slice

fmt.Println(newSlice)

newSlice = append(newSlice, 3)

fmt.Println(newSlice)
```

Slice - primer

[3]bool{true, true, false} //array

[]bool{true, true, false} //slice

Internally builds an array and references it

```
var myArray [5]
//the following expressions give the same result
myArray[0:5]
myArray[:5]
myArray[0:]
myArray[:]
```

Zadatak - liste

- Dat je niz (lista) celih pozitivnih brojeva
- Funkcija prima niz preko reference
- Povratna vrednost funkcije je nova lista koja sadrži sortirane brojeva iz prosleđene liste
- Sortiranje implementirati ili Bubble sort ili Selection sort ili Insertion sort algoritmom

Mape

- Struktura koja je slična rečnicima i mapama iz drugih programskih jezika
- Parovi ključ vrednost

```
var m map[string]int
m = make(map[string]int)
m["S1"] = 11
m["S2"] = 12
m["M3"] = 15

for key, value := range m {
    fmt.Printf("Key is : %s, Value is : %d\n", key, value)
}
```

Zadatak - Mape

- Kreirati mapu koja za ključ ima broj indeksa studenta, a za vrednost instancu strukture tipa Student (ime, prezime, godina upisa, prosek)
- Kroz meni omogućiti
  - unos novog studenta
  - brisanje postojećeg studenta
- Obraditi potencijalne greške

Greške

- Nema try-catch-finally blokova
- Funkcije vraćaju grešku kao jednu od povratnih vrednosti
  - Interfejs Error

```
f, err := os.Open("filename.ext")
if err != nil {
    log.Fatal(err)
}
```

Zadatak

- Implementirati funkciju koja koristi Quick sort algoritam za sortiranje niza celih brojeva
- Generisati niz od milion brojeva
  - Koristiti random
    - Paket - math/rand
  - Meriti vreme potrebno za sortiranje
  - Uporediti vremena dobijena sa vremenima koja se dobiju sortiranjem istog niza korišćenjem Bubble Sort-a

Zadatak

- Implementirati strukturu steka (sekvencijalno i spregnuto)
  - Funkcije:
    - pop
    - push
    - top
    - isEmpty

Zadatak

- Napisati program koji evidentira studente

- Omogućiti interakciju sa korisnikom, tako da korisnik može da bira jednu od tri opcije:
  - Unos novog studenta u evidenciju (fajl)
    - Pri čemu program neće dozvoliti unos studenta ako postoji u evidenciji student sa istim brojem indeksa (Vratiće opis greške korisniku)
    - Podaci o studentu su (broj indeksa, ime, prezime, prosek) Za svaki unos instancirati strukturu tipa Student
    - Upisati podatke o studentu u fajl
  - Ispis svih studenata sortiranih po proseku pri čemu se može birati opadajući ili rastući redosled
  - Ukupan prosek svih studenata
  - Obraditi potencijalne greške

Konekcija na bazu - primer

```
import "database/sql“
import _ "github.com/lib/pq"
//...
 db, err := sql.Open(“mysql", "user=root dbname=db1 sslmode=verify-full")
if err != nil {
       panic(err)
 }
id := 1
var col string sqlStatement := `SELECT A FROM testTable WHERE id=$1`
row := db.QueryRow(sqlStatement, id)
err := row.Scan(&col)
if err != nil {
      if err == sql.ErrNoRows {
            fmt.Println(“No rows found")
     } else {
          panic(err)
     }
 }
```

Zadatak za rad kod kuće

- Implementirati funkciju koja kreira binarno stablo na osnovu statičkog niza koji sadrži cele brojeve
- Omogućiti obilazak stabla metodom DFS
  - Pretraga prvi u dubinu
- Omogućiti obilazak stabla metodom BFS
  - Pretraga prvi u šitinu

Web programiranje u Golang-u

?