



FAKULTET INŽENJERSKIH NAUKA UNIVERZITET U KRAGUJEVCU

Tema: Programiranje aplikacija IOS sistema
Student: Nevena Stašić
Profesor: Nenad Grujović

Kragujevac, 2019. godine

Sadržaj:

- ◉ Funkcije
- ◉ Definisanje funkcije
- ◉ Enum tip podataka
- ◉ Strukture

Funkcije

- Funkcija je skup akcija koje su organizovane zajedno da bi se izvršio određeni zadatak. Swift 4 funkcija može biti jednostavna kao jednostavna C funkcija i kompleksna kao funkcija Objective C programskog jezika. To nam omogućava da prosledimo lokalne i globalne vrednosti parametara unutar poziva funkcija.
- Deklaracija funkcija: govori kompajleru ime funkcije, vrstu povratne informacije i parametrima.
- Definisanje funkcija: pruža stvarni sadržaj funkcije.

Definisanje funkcija

- U Swift 4 programskom jeziku, funkcija je definisana ključnom rečju “func”. Kada je funkcija definisana, možemo da unesemo jednu ili više vrednosti kao ulazne ‘parametre’ i parametre koje će funkcija vratiti kao izlazni ‘return types’.
- Svaka funkcija ima ime funkcije, koje opisuje zadatak koji funkcija obavlja. Da bismo koristili funkciju, ‘pozovemo’ tu funkciju sa svojim imenom i prosledimo ulazne vrednosti koje odgovaraju tipovima parametara funkcije.

Definisanje funkcije

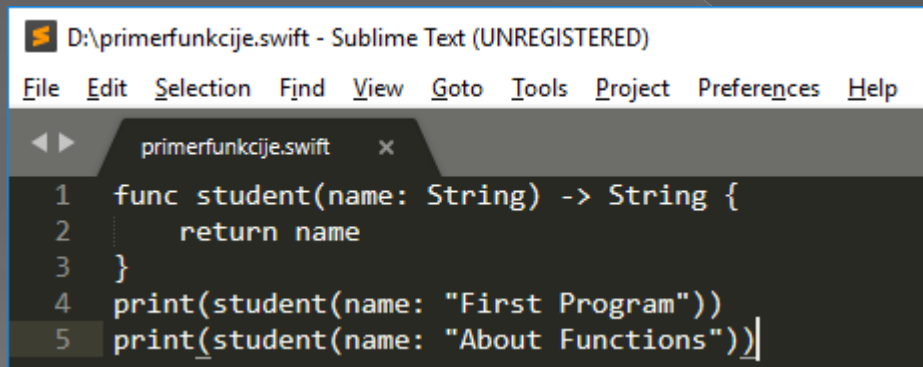
- Argumenti funkcije uvek moraju biti navedeni u istom redosledu kao i lista parametara te funkcije.
- Sintaksa za definisanje funkcija je sledeća:

```
func funcname(Parameters) -> returntype {  
    Statement1  
    Statement2  
    ---  
    Statement N  
    return parameters  
}
```

Slika 1 – Sintaksa za definisanje funkcija

Definisanje funkcije

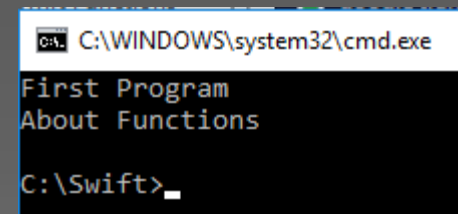
- Korišćenje sintakse za definisanje funkcija je najjednostavnije objasniti na primeru, koji će biti predstavljen na slici 2, ispod:



The screenshot shows a Sublime Text editor window titled "D:\primerfunkcije.swift - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The editor displays a Swift file named "primerfunkcije.swift" with the following code:

```
1 func student(name: String) -> String {  
2     return name  
3 }  
4 print(student(name: "First Program"))  
5 print(student(name: "About Functions"))
```

Slika 2 – Primer za definisanje funkcije



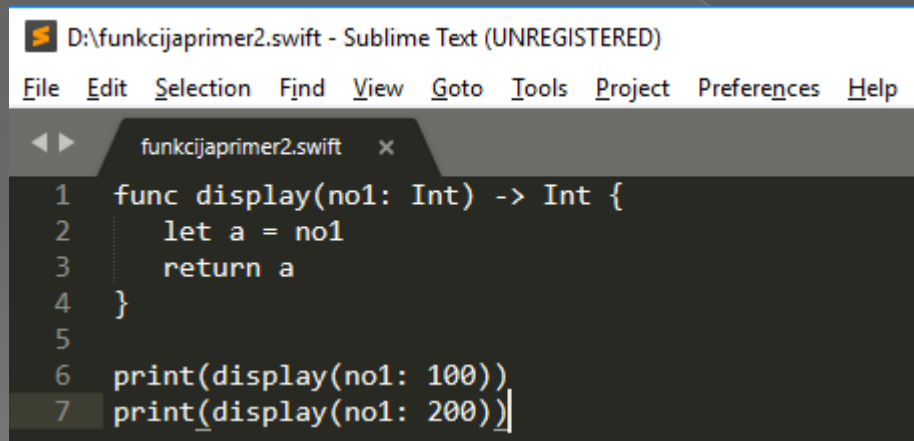
The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt is "C:\Swift>". The output of the program is displayed as:

```
First Program  
About Functions
```

Slika 3 – Startovan program

Definisanje funkcije

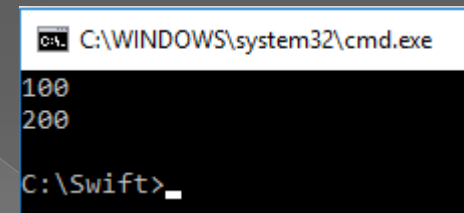
- U nastavku ispod, biće prikazan još jedan primer definisanja funkcije, koji će predstavljati funkciju display, u okviru koje će se izvršavati neka operacija.



```
D:\funkcijaprimer2.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

funkcijaprimer2.swift x
1 func display(no1: Int) -> Int {
2     let a = no1
3     return a
4 }
5
6 print(display(no1: 100))
7 print(display(no1: 200))
```

Slika 4 – Primer definisanja funkcije



```
C:\WINDOWS\system32\cmd.exe
100
200
C:\Swift>
```

Slika 5 – Startovan program

Enum tip podataka

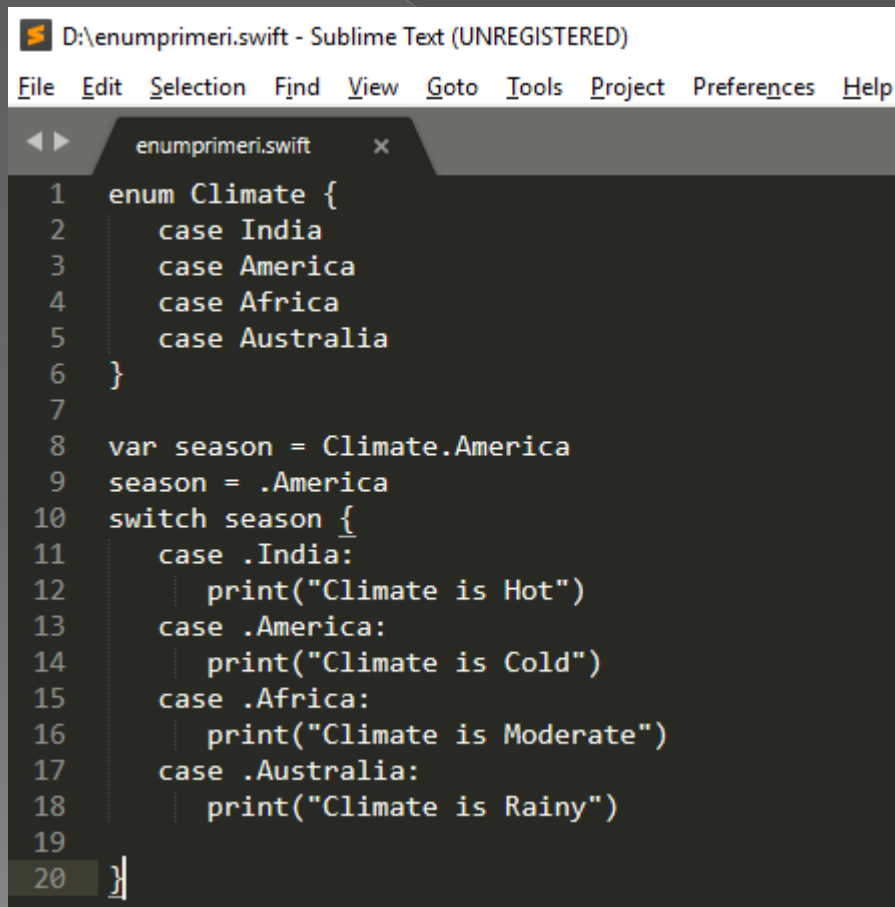
- Enum je korisnički definisani tip podataka koji se sastoji od skupa povezanih vrednosti. Ključna reč enum se koristi za definisanje ovog tipa podataka.
- Enumi u Swift 4 programskom jeziku podsećaju na strukture u C i Objective C programskom jeziku.
- Deklariše se u klasi i njegovim vrednostima se pristupa kroz instancu te klase.
- Sintaksa za definisanje enum tipa podataka je prikazana na slici ispod:

```
enum enumname {  
    // enumeration values are described here  
}
```

Slika 6 – Sintaksa za definisanje enum tipa podataka

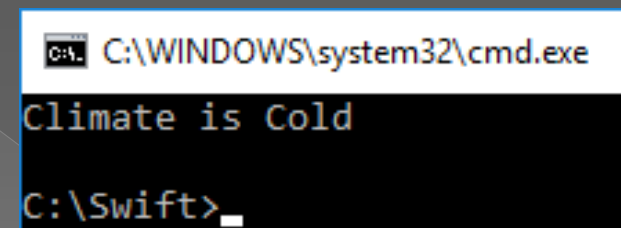
Enum tip podataka

- Korišćenje i rad sa enum tipom podataka je najjednostavnije prikazati kroz primer:



```
D:\enumprimeri.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

enumprimeri.swift x
1 enum Climate {
2     case India
3     case America
4     case Africa
5     case Australia
6 }
7
8 var season = Climate.America
9 season = .America
10 switch season {
11     case .India:
12         print("Climate is Hot")
13     case .America:
14         print("Climate is Cold")
15     case .Africa:
16         print("Climate is Moderate")
17     case .Australia:
18         print("Climate is Rainy")
19
20 }
```



```
C:\WINDOWS\system32\cmd.exe
Climate is Cold
C:\Swift>
```

Slika 8 – Startovan program

Slika 7 – Primer rada sa enum tipom podataka

Strukture

- ◉ Swift 4 programski jezik obezbeđuje fleksibilan blok za korišćenje konstruktora kao struktura. Korišćenjem ovih struktura nekada se mogu definisati metode i svojstva konstruktora.
- ◉ Struktura nam omogućava da kreiramo jedan fajl i proširimo njegov interfejs automatski na druge blokove.
- ◉ U strukturi se vrednosti promenljivih kopiraju i prosleđuju u narednim kodovima tako što se vraća kopija starih vrednosti i tada se vrednosti ne mogu menjati.

Strukture

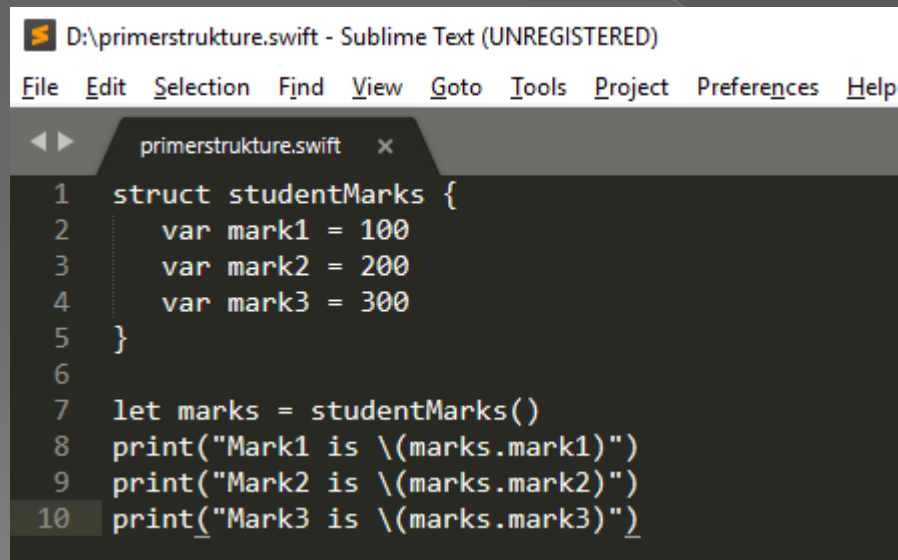
- Na slici ispod je prikazana sintaksa kreiranja strukture.

```
Structures are defined with a 'Struct' Keyword.  
struct nameStruct {  
    Definition 1  
    Definition 2  
    ---  
    Definition N  
}
```

Slika 9 – Sintaksa kreiranja struktura

Strukture

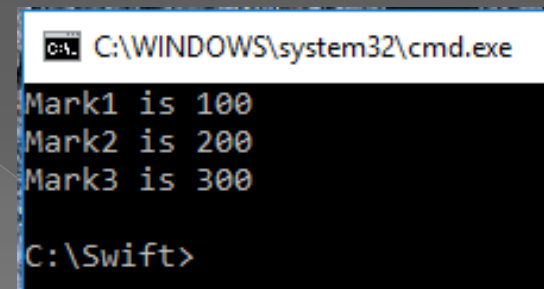
- Članovima strukture se pristupa nazivom strukture. Primeri strukture inicijalizuju se pomoću ključne reči “let”, kao što je prikazano na slici ispod:



```
D:\primerstrukture.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

primerstrukture.swift x
1 struct studentMarks {
2     var mark1 = 100
3     var mark2 = 200
4     var mark3 = 300
5 }
6
7 let marks = studentMarks()
8 print("Mark1 is \(marks.mark1)")
9 print("Mark2 is \(marks.mark2)")
10 print("Mark3 is \(marks.mark3)")
```

Slika 10 – Primer kreiranja strukture



```
C:\WINDOWS\system32\cmd.exe
Mark1 is 100
Mark2 is 200
Mark3 is 300
C:\Swift>
```

Slika 11 – Startovan program