



FAKULTET INŽENJERSKIH NAUKA UNIVERZITET U KRAGUJEVCU

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

Kragujevac, 2019. godine

SADRŽAJ:

- ⦿ Promenljive u Swiftu
- ⦿ Definisanje alijasa
- ⦿ Type safety
- ⦿ Deklarisanje promenljivih
- ⦿ Type annotations
- ⦿ Štampanje promenljivih
- ⦿ Opcionalni tip podatka (Optionals)
- ⦿ Konstante

Promenljive u Swiftu

Dok programiramo u bilo kom programskom jeziku, koristimo različite tipove promenljivih za skladištenje podataka. Promenljive nisu ništa drugo nego rezervisana memorijska mesta za skladištenje podataka. To znači da kada kreiramo promenljivu, mi rezervišemo prostor u memoriji.

Možda želimo da sačuvamo informacije o različitim tipovima podataka kao što su String, character, integer, double, boolean i drugi. Na osnovu tipa podataka promenljive, operativni sistem dodeljuje memoriju i odlučuje šta se može uskladištiti u rezervisanoj memoriji.

Promenljive u Swiftu

Swift 4 nudi programeru bogat asortiman ugrađenih i korisnički definisanih tipova podataka.

Osnovni tipovi podataka koji se najčešće koriste prilikom deklarisanja promenljivih su:

- Int
- Float
- Double
- Bool
- String
- Character
- Optional – Ovo predstavlja promenljivu koja može i ne mora da sadrži vrednost.
- Tuples – Ovo se koristi za grupisanje višestrukih vrednosti u pojedinačnoj vrednosti.

Promenljive u Swiftu

Sledeća tabela prikazuje tip promenljive, koliko memorije je potrebno za skladištenje vrednosti u memoriji i koja je maksimalna i minimalna vrednost koja se može uskladištiti u takvom tipu promenljivih.

| Type | Typical Bit Width | Typical Range |
|--------|-------------------|---|
| Int8 | 1byte | -127 to 127 |
| UInt8 | 1byte | 0 to 255 |
| Int32 | 4bytes | -2147483648 to 2147483647 |
| UInt32 | 4bytes | 0 to 4294967295 |
| Int64 | 8bytes | -9223372036854775808 to 9223372036854775807 |
| UInt64 | 8bytes | 0 to 18446744073709551615 |
| Float | 4bytes | 1.2E-38 to 3.4E+38 (~6 digits) |
| Double | 8bytes | 2.3E-308 to 1.7E+308 (~15 digits) |

Slika 1 – Tabela informacija tipova promenljivih

Definisanje alijasa

Uz pomoć alijasa, možemo da kreiramo novo ime za postojeći tip promenljive.

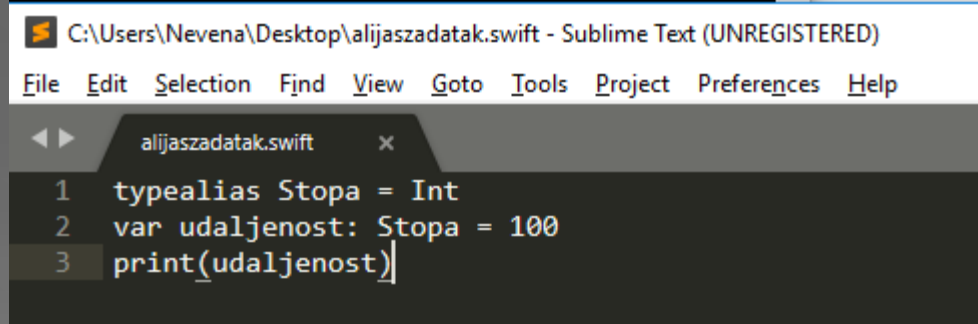
Jednostavna sintaksa za definisanje novog tipa sastoji se iz ključne reči 'typealias' i od tipa promenljive.

```
typealias newname = type
```

Slika 2 – Kreiranje alijasa

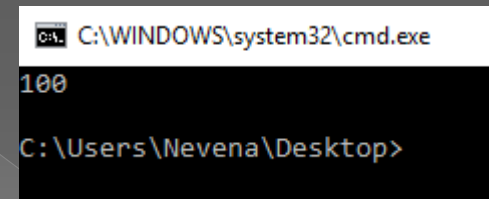
Definisanje alijasa

Primer: Definisanje alijasa pod nazivom 'stopa', tipa podatka integer, zatim definisanje nove promenljive udaljenost tipa 'stopa' i dodeljivanje neke vrednosti promenljivi pod nazivom udaljenost.



```
C:\Users\Nevena\Desktop\alijaszadatak.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
alijaszadatak.swift x
1 typealias Stopa = Int
2 var udaljenost: Stopa = 100
3 print(udaljenost)
```

Slika 3 – Kod primera za alijas



```
C:\WINDOWS\system32\cmd.exe
100
C:\Users\Nevena\Desktop>
```

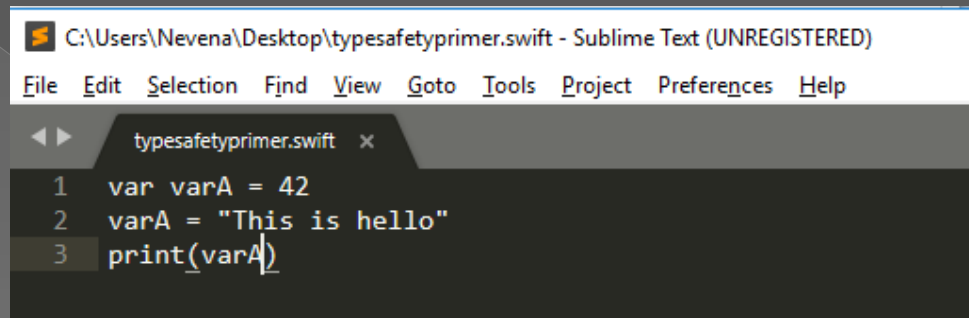
Slika 4 – Startovanje programa

Type safety

Swift 4 je type-safe jezik, što znači da, ako deo našeg koda očekuje String, ne možemo greškom da mu prosledimo int.

Budući da je Swift 4 bezbedan po tom pitanju, on izvršava proveru tipa podatka, prilikom kompajliranja koda i označava sve neusaglašene tipove kao greške.

Primer:



```
C:\Users\Nevena\Desktop\typesafetyprimer.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

typesafetyprimer.swift x
1 var varA = 42
2 varA = "This is hello"
3 print(varA)
```

Slika 5 - Kod

Logs

```
"C:/Swift/usr/bin/swiftc.exe" -swift-version 4 "C:\Users\Nevena\Desktop\typesafetyprimer.swift" -o "C:\Users\Nevena\Desktop\typesafetyprimer.exe"
```

```
C:\Users\Nevena\Desktop\typesafetyprimer.swift:2:8: error: cannot assign value of type 'String' to type 'Int'
varA = "This is hello"
      ^~~~~~
```

Compilation Failed

Slika 6 – Greška

Deklarisanje promenljivih

Deklaracija promenljivih govori kompajleru gde i koliko da kreira skladište za promenljivu. Pre nego što upotrebimo promenljive, moramo ih deklaristati pomoću 'var' ključne reči na sledeći način:

```
var variableName = <initial value>
```

Slika 7 – Deklarisanje promenljive

Type annotations

Anotacija tipa podataka se vrši kada želimo da deklariramo promenljivu, da bismo bili jasni o vrsti vrednosti koje promenljiva može da uskladišti.

Ako ne koristimo anotaciju tipa podataka, onda postaje obavezno dati inicijalnu vrednost za promenljivu.

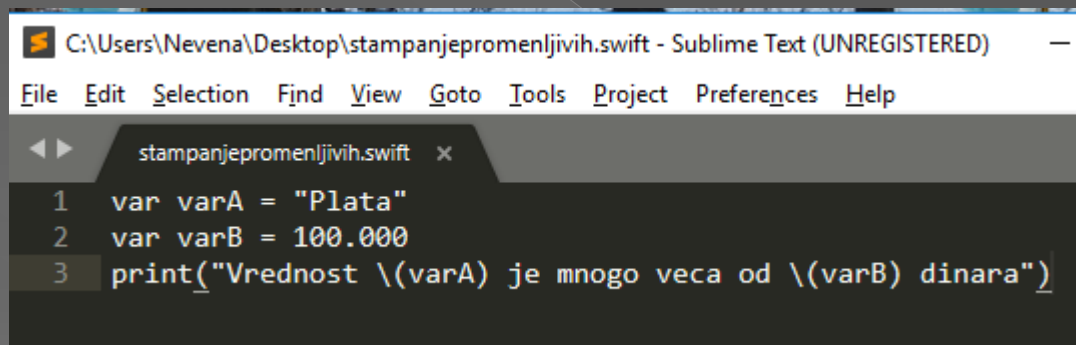
Primer anotacije tipa podataka:

```
var variableName:<data type> = <optional initial value>
```

Slika 8 – Type annotations

Štampanje promeljivih

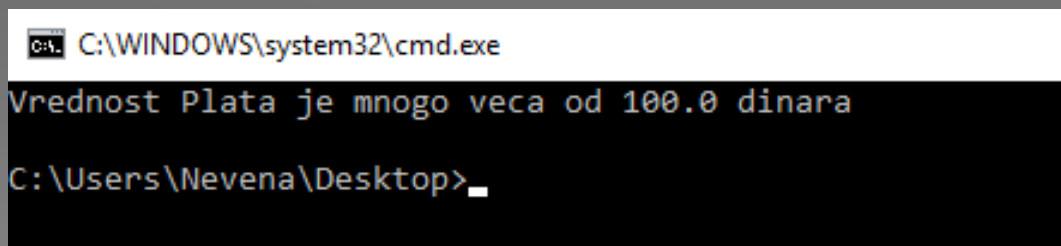
Primer: Deklarisanje dve promenljive sa određenim vrednostima i štampanje vrednosti te dve promenljive u okviru nekog teksta.



```
C:\Users\Nevena\Desktop\stampanjepromenjivih.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

stampanjepromenjivih.swift x
1 var varA = "Plata"
2 var varB = 100.000
3 print("Vrednost \ (varA) je mnogo veka od \ (varB) dinara")
```

Slika 9 - Kod



```
C:\WINDOWS\system32\cmd.exe
Vrednost Plata je mnogo veka od 100.0 dinara
C:\Users\Nevena\Desktop>
```

Slika 10 – Startovanje programa

Opcionalni tip podatka (Optionals)

Ovo je tip podataka koji ima dve moguće vrednosti: None i Some(T), gde je T, pridružena vrednost ispravnog tipa podataka dostupnog u Swiftu.

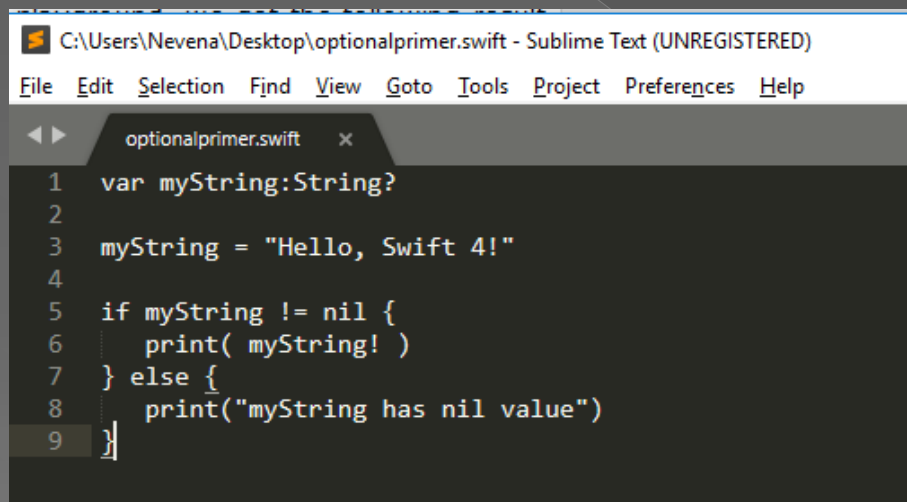
Primer deklaracije optional integer-a:

```
1 var vrednostInt: Int?
```

Slika 11 – Optional Integer deklaracija

Opcionalni tip podatka (Optionals)

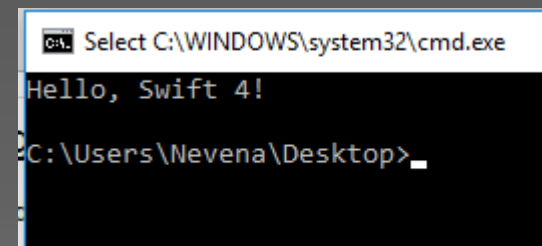
Primer: Deklaracija opcionalnog tipa podatka i rad sa takvim podatkom u okviru uslova.



```
C:\Users\Nevena\Desktop\optionalprimer.swift - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

optionalprimer.swift x
1 var myString:String?
2
3 myString = "Hello, Swift 4!"
4
5 if myString != nil {
6     print( myString! )
7 } else {
8     print("myString has nil value")
9 }
```

Slika 12 - Kod



```
C:\ Select C:\WINDOWS\system32\cmd.exe
Hello, Swift 4!
C:\Users\Nevena\Desktop>
```

Slika 13 – Startovan program

Konstante

Konstante se odnose na fiksne vrednosti koje program ne može da promeni tokom izvršavanja. Konstante mogu biti od bilo kog osnovnog tipa podataka kao što je integer constant, floating constant, character constant ili string constant. Postoje i enum konstante nabiranja.

Konstante se tretiraju kao regularne promenljive samo što se njihove vrednosti ne mogu menjati nakon njihove definicije.

Pre nego što upotrebimo konstante, moramo ih deklarirati pomoću ključne reči 'let' na sledeći način:

```
1 let imekonstante = inicijalnavrednost
```

Slika 14 – Deklaracija konstante