

**Prolećni semestar, 2016/17**

##### Domaći zadatak: 02

­

**Ime:** Nemanja Andri’

**BrojIndeksa:** 2001

**Predmet:** Skripting jezici

**Šifra predmeta:** CS324

**Datum:** 09.07.2017.

# Cilj domaćeg zadatka:

Za domaći zadatak uraditi sledeće zadatke:

1. Koju funkciju ima interpreter?
2. Šta predstavlja prva linija koda (#! /usr/bin/perl)?
3. Koja su tri osnovna tipa Perl promenljivih?
4. Napiasati Perl potprogram show\_args koji štampa svoje argumente. Na primer, ako su argumenti “fred”, “barney”, i “betty”, potprogram se poziva na slijedeći način: show\_args( 'fred', 'barney', 'betty' );

# Rešenje:

1. Interpretiranje je način prevođenja izvornog koda u izvršni kod programa u toku njegovog izvršenja. Prevođenje se vrši pri svakom izvođenju programa. Pri ovom načinu prevođenja obrađuje se svaka naredba (tj. svaka linija koda) pojedinačno, što može u određenoj meri usporiti izvršavanje, ali sa druge strane omogućava trenutno otkrivanje eventualnih greški i njihovo interaktivno otklanjanje.

Što se tiče programa pisanih u jeziku Perl, svaki od njih mora proći kroz Perlovu izvršnu datoteku (interpreter) kako bi bio izvršen. Na računaru, on se obično nalazi na jednoj od ove 3 lokacije: ***/usr/bin*** ili ***/usr/local/bin*** ili ***/usr/share/man/man1.*** U suprotnom, jednostavnim ukucavanjem ***whereis perl*** komande u komandnu liniju, lako možemo saznati lokaciju izvršne datoteke.

Međutim, važno je napomenuti da iako Perl spada u grupu takvih, on ipak nije strogo interpretirani jezik, budući da interpreter najpre prevodi kod u *bytecode* format pre samog izvršenja, te se zbog toga često naziva *interpreter/prevodioc*.

1. Zbog svega ovoga gore spomenutog, jako je važno da svaka Perl skripta na svom početku sadrži obaveznu liniju koda koja predstavlja putanju do interpretera, u kombinaciji sa **#!** (hash-bang ili shebang) simbolima.

***#!/usr/bin/perl*** linija koda omogućava Perlovoj skripti da postane izvršna, ona zapravo traži izvršnu datoteku na datoj lokaciji i predaje joj ostatak koda na izvršenje.

1. U Perl-u razlikujemo tri tipa promenljivih (varijabli):

* Skalare
* Nizove
* Heševe

Svaki od ova tri tipa promenljivih obeležava se posebnim simbolom ispred samog imena varijable.

Za definisanje skalara koristimo simbol **$** ispred imena promenljive. Ovaj tip promenljive u suštini predstavlja jednostavne varijable (brojeve, stringove, reference).

Primeri skalara u Perl-u:

*my $name = "Mirko";*

*my $age = 30;*

Nije na odmet napomenuti da su sve promenljive u Perlu unapred definisane kao globalne. Dodavanjem reči ***my*** ispred same promenljive, menjamo njen opseg delovanja na jedan blok, i ona postaje lokalna, što predstavlja dobru programersku praksu.

Nizovima, kao tipu varijable, prethodi znak @, i oni predstavljaju uređene liste skalara, kojima se pristupa pomoću numeričkog indeksa (ne zaboraviti da indeksiranje u nizovima kreće od 0).

Primer niza u Perl-u:

@names = (“Milan”, “Aleksandra”, “Ivana”);

Niz može sadržati i praznu listu. Kao što je već gore spomenuto, elementima niza pristupa se pomoću njihovog indeksa, tj.lokacije u listi:

print $names[0]; # daće rezultat -> Milan

Trećem tipu promenljivih u Perlu prethodi znak procenta %. Ove promenljive nazivaju se heševi i uvek su definisane kao skup parova ključ/vrednost.

Primer heša:

%ID = ('age', 22, 'id\_number', 6);

Često se pri definisanju ovih parova ključ/vrednost, radi bolje preglednosti, koristi operator => koji je ništa više do zamene za zapetu:

%ID = (

age => 22,

id\_number => 6

);

Ukoliko želimo da izdvojimo vrednost koja je pridružena ključu ***age***, to radimo na sledeći način:

$ID {'age'};

1. Rešenje zadatka:

