

1 Instaliranje i pokretanje

Da bi se uspešno pokretao perl skript potrebno je imati kompajler. U ovom poglavlju biće objašnjeno njegovo instaliranje na Linux i Windows operativnim sistemima.

1.1 Instaliranje na Linux operativnom sistemu

Da bismo instalirali Perl kompajler na Linux operativnom sistemu potrebno je da u terminalu unesemo komandu `sudo apt-get install perl`. Od vas će se zatražiti da unesete lozinku. Kada se instalira Perl kompajler u terminalu pokrenite sledeću komandu `curl -L http://xrl.us/installperlnix | bash`. Kada sve bude gotovo, da biste se uverili da je instalacija uspešno izvršena ili ako želite da proverite koja je verzija Perl-a instalirana, to možete uraditi komandom `perl -v`.

1.2 Pokretanje na Linux operativnom sistemu

Pokretanje Perl skripta na Linux operativnom sistemu se radi na jednostavan način. Potrebno je otvoriti terminal, a zatim uneti komandu `perl <putanja_do_fajla>`. Ova komanda će pokrenuti izvršavanje skripta.

1.3 Instaliranje na Windows operativnom sistemu

Pre instalacije na Windows operativnom sistemu poželjno je prethodno proveriti da nijedna verzija Perla već nije instalirana. Ako želite da deinstalirate prethodnu verziju idite na Control Panel - Add/Remove programs. Ako i dalje imate C:\Strawberry folder, izbrišite ga ili ga preimenujte. Skinite i instalirajte Padre, the Perl IDE/editor. Strawberry Perl je deo instalacije, ali takođe dobijate i mnoge druge korisne CPAN module. Nakon instalacije, možda ćete morati da restartujete računar. Zatim idite na Start meni/All Programs i pronađite folder sa Perlom. U njemu kliknite na Perl komandnu liniju. Da biste instalirali module iz CPAN-a u komandnoj liniji ukucajte komandu `cpan App::cpanminus`. Ukoliko želite da potvrdite da je instalacija uspešna ukucajte `perl -v`, ovom komandom, takođe, možete proveriti koja je verzija perl-a instalirana.

1.4 Pokretanje na Windowsu

2 Primeri koda

U listingu 1 dat je primer ispisa Hello world programa. Komentare pišemo iza tarabe. Tekst možemo ispisati tako što ga navedemo iza navodnika ili koristeći funkciju `qq`.

```
1000 #Komentar
      print "Hello, world!\n";
1002 print qq =Did you say "Hello"? \n=;
```

Listing 1: Primer Hello world programa

Naredni primer demonstrira program koji računa zbir dva broja. Imena promenljivih počinju znakom `$`. U komentarima je dato šta koja print naredba ispisuje.

```

1000 $a = 5;
      $b = 4;
1002 print qq =a \= $a\n=; # a = 5
      print qq =b \= $b\n=; # b = 4
1004 $zbir = $a + $b;
      print qq=a + b \= $zbir\n=; #a + b = 9

```

Listing 2: Zbir dva broja

Nazivi nizova u Perlu su u formatu @naziv_niza. Niz se može inicijalizovati tako što elemete navedemo u zagradama, razdvojene zarezom. Ako želimo da ispišemo sve elemete niza, to možemo učiniti samo navođenjem imena niza. Određenom članu niza pristupamo tako što njegov indeks navedemo u uglastim zagradama iza naziva niza. U ovom slučaju ime niza može počinjati i znakom @ i znakom \$. Indeks poslednjeg člana niza dobijamo tako što ispre imena niza stavimo \$#.

```

1000 @dan = ("Danas", "je", "lep", "dan");
      $broj_reci = @dan; #Broje elementa niza
1002 print "broj reci u recenici \"@dan\" je $broj_reci.\n";
      print "@dan[2], $dan[3]\n";
1004 print "Indeks poslednjeg elementa je $#dan\n";

```

Listing 3: Niz

Grananje u programu možemo izvršiti koristeći if-else naredbu. Naredni program prvo učitava broj sa standardnog ulaza, za zatim proverava da li je pozitivan ili je negativan. Uslov se navodi u zagradi, a if i else blok se navode iznad vitičastih zagrada.

```

1000 $broj = <STDIN>;
1002 if($broj >= 0){
      print "Broj je pozitivan\n";
1004 }
      else{
1006     print "Broj je negativan\n";
      }

```

Listing 4: If-else naredba

Možemo zadavati i argumente komandne linije. Argumentima komandne linije pristupamo isto kao i elementima niza, samo što kao naziv niza navodimo ARGV. Broj argumenata komandne linije možemo dobiti tako što indeks poslednjeg elementa uvećamo za jedan.

```

1000 $br_arg = $#ARGV + 1;
      print "Broj argumenata komandne linije je $br_arg, a to su @ARGV\n";

```

Listing 5: Argumenti komandne linije

Sledećim kodom je prikazan primer korišćenja while petlje. Program prihvata argumente sa ulaza i ispisuje ih na ekran. Ključna reč my označava da je domen promenljive samo taj blok. Promenljiva \$in u while petlji nema uticaj na promenljivu sa istim nazivom van petlje.

```

1000 $in = 5;
1002 while(my $in = <>) {
      print $in; #Ispisuje element koji smo uneli
1004 }
      print $in; #Ispisuje broj 5

```

Listing 6: Unos i ispis elemenata u while petlji

Prethodnu petlju možemo zapisati i na sledeći način.

```
1000 while(<>) {  
1002     print;  
}
```

Listing 7: Unos i ispis elemenata u while petlji

U Perlu možemo koristiti i for petlju na sličan način kao i u programskom jeziku C.

```
1000 for(my $i = 0; $i <= $#ARGV; $i++) {  
1002     print "$ARGV[$i]\n";  
}
```

Listing 8: For petlja

Funkcije započinju ključnom rečju `sub` iza koje sledi naziv funkcije. Telo funkcije se piše između vitičastih zagrada. Argumenti se navode u telu funkcije na sledeći način `my(lista_argumenata) = @_`; Poziv funkcije se vrši na standardan način.

```
1000 sub obim_kvadrata {  
1002     my($a) = @_;  
1004     return 4*$a;  
1006 }  
1008 $obim = obim_kvadrata($aa);  
1010 print "$obim\n";  
1012 print obim_kvadrata(4);  
1014 print "\n";
```

Listing 9: Funkcija koja računa obim kvadrata

Primer 2.1 *Sledeći kod predstavlja jednostavan način da zamenimo vrednosti dve promenljive.*

```
1000 aa = 3;  
1002 $bb = 2;  
1004 ($aa, $bb) = ($bb, $aa);  
1006 print "$aa $bb\n";
```

Listing 10: Zamena vrednosti dve promenljive