

PERL

DODATNI SLAJDOVI

Sadržaj

- Reference
- Paketi
- Objekti i klase
- CPAN

Pojam referenci u Perlu

- Motivacija
 - Nemogućnost kreiranja složenih podatkovnih struktura
 - Npr. liste u listama, ili asocijativna polja koja sadrže liste
 - Korištenje listi i asocijativnih polja pri pozivu funkcija
- Reference *pokazuju (referiraju)* na *nešto*
 - Slično kao pokazivači u C-u
- Dvije vrste referenci
 - *hard*
 - simboličke (*samo ćemo kratko o njima*)
- Reference mogu "pokazivati" na (skoro) sve tipove Perl podataka
 - problematične su datoteke...

Korištenje simboličkih referenci

- Simboličke reference pristupaju vrijednostima putem imena
 - Vrijednost skalara se uzima kao ime varijable
 - *Varijabla u varijabli*

- **Primjeri**

```
$name = "foo";  
$$name = 1;           # Sets $foo  
${$name} = 2;        # Sets $foo  
${$name x 2} = 3;    # Sets $foofoo  
$name->[0] = 4;       # Sets $foo[0]  
@$name = ();         # Clears @foo  
&$name();            # Calls &foo()
```

- Upozorenje 1: Ne radi za varijable deklarirane s `my`
- Upozorenje 2: Ne koristiti ako nije nužno

Stvaranje (čvrstih) referenci

- U Perlu postoji 6 tipova referenci
 - nama je bitno 5

Dodjela reference

`$refScalar = \ $scalar`

`$refArray = \@array`

`$refHash = \%hash`

`$refFunction = \&function`

`$refRef = \ $refScalar`

Pristup referenci (dereferenciranje)

`${$refScalar}`

`@{$refScalar}`

`%{$refScalar}`

`&{$refScalar}`

`$$refScalar`

- Alternativni način kreiranja referenci
 - Anonimne liste ili asocijativna polja

```
$lref = [ elem1, elem2, elem3, ... ]
```

```
$aref = { key1 => val1, key2 => val2, ... }
```

Primjer korištenja referenci (1)

```
$ cat primjer.pl
#!/usr/bin/perl
$scalar = 5;
$refScalar = \ $scalar;

print "$refScalar\n";
print "${$refScalar}\n";

$ ./primjer.pl
SCALAR(0x238ed90)
5
$
```

Primjer korištenja referenci (2)

- Stvaranje dvodimenzionalnog polja

```
@a = ([1, 2, 3],  
      [4, 5, 6],  
      [7, 8, 9]);
```

- Pristup elementima dvodimenzionalnog polja

- "Standardni" način

`$$a[0][0]` daje 1

- Pokrata 1

`$a[0]->[0]` daje 1 (`$a[ROW]->[COL]`)

- Pokrata 2

`$a[0][0]` daje 1 (`$a[ROW][COL]`)

Primjer korištenja referenci (3)

- Prijenos parametara potprograma
 - Vrlo je teško prenijeti dvije liste kao parametre u funkciju
 - Elementi lista se spajaju
 - Korištenjem referenci to je moguće

```
&funkcija( \@lista1, \%lista2 )
```
 - Analogno vrijedi za bilo koju kombinaciju lista i asocijativnih polja

Funkcija ref()

- Omogućava provjeru rukujemo li s referencom

```
$ cat primjer2.pl
#!/usr/bin/perl
$refScalar = \ $scalar;
$refList = \@list;
$refHash = \%hash;

print ref($scalar) . "\n";
print ref($refScalar) . "\n";
print ref($refList) . "\n";
print ref($refHash) . "\n";
$ ./primjer2.pl
```

```
SCALAR
ARRAY
HASH
$
```

Funkcija `bless()`

- Utječe na povratnu vrijednost funkcije `ref()`

```
$ cat bless.pl
#!/usr/bin/perl
$foo      = { };
$fooRef   = $foo;
print("data of \ $foo is "      . ref($foo)      . "\n");
print("data of \ $fooRef is "   . ref($fooRef)   . "\n");
bless($foo, "Bar");
print("data of \ $foo is "      . ref($foo)      . "\n");
print("data of \ $fooRef is "   . ref($fooRef)   . "\n");
```

```
$ ./bless.pl
data of $foo is HASH
data of $fooRef is HASH
data of $foo is Bar
data of $fooRef is Bar
```

Kreiranje paketa

- Perl podržava izgradnju paketa
 - paket (modul) je zaseban prostor imena
- Pravila za pisanje paketa
 - Paket se nalazi u zasebnoj datoteci (iako ne nužno)
 - Datoteka mora imati ekstenziju `.pm`
 - Ključna riječ `package` definira ime paketa
- Primjer (trivijalnog) paketa

```
$ cat nekipaket.pm
```

```
#!/usr/bin/perl
```

```
package nekipaket;
```

```
sub nekafunkcija { print "nekafunkcija\n"; }
```

Korištenje paketa

- Paketi se uključuju s ključnom riječi `use`
- Pretraživanje se obavlja u direktorijima navedenima u listi `@INC`
- Pakete je moguće hijerarhijski smještati u direktorije
 - paket `nekagrupa::nekipaket` se mora nalaziti u direktoriju `nekagrupa`, dok je samo ime paketa `nekipaket.pm`
 - hijerarhija može biti proizvoljno duboka

- Primjer

```
$ cat glavni.pl
#!/usr/bin/perl
use nekipaket;
```

```
&nekipaket::nekafunkcija;
```

Objekti i klase u Perlu

- Klasa je struktura koja sadrži attribute i funkcije (metode)
 - Klasa je pohranjena u datoteci (paketu)
 - Običaj je definirati funkciju (metodu) `new` koja vraća objekt (konstruktor)
- Objekt je instanca (primjerak) klase
 - Objekt je referenca (dakle, koristi se `$objekt` oblik)
 - Metode i atributi objekta referenciraju se korištenjem operatora `->`
 - Razlika između "obične" reference i objekta je u korištenju funkcije `bless`
 - Referenca s njim "postaje" objekt određenog tipa

Pisanje klase i njeno korištenje

- Primjer jednostavne klase

```
package Person;  
  
sub new {  
    my($class) = shift;  
  
    bless {  
        "NAME"    => undef,  
        "AGE"     => undef  
    }, $class;  
}
```

```
package main;  
  
use Person;  
  
$pero = Person->new();
```

Dodavanje metoda klasi/objektu

- Prvi argument metode je referenca na objekt

```
package Person;  
  
sub new {  
    ...  
}  
  
sub name {  
    my $self = shift;  
  
    if (@_) { $self->{NAME} = shift }  
  
    return $self->{NAME};  
}
```

```
$pero->name("Pero Peric");  
print $pero->name;
```

CPAN

- Perl ima arhivu paketa koji značajno olakšavaju izradu skripti
 - CPAN – Comprehensive Perl Archive Networks
 - Pretraživanje arhive na <http://search.cpan.org>
- Skoro svi paketi su objekti!
 - Nužno poznavanje barem korištenja objekata s prethodnih slajdova

Dohvat i instalacija paketa

- Linux/Unix distribucije već sadrže niz pripremljenih Perl paketa
 - Potrebna je samo instalacija
- Dohvat i instalacija paketa
 - Prvi način je skidanje izvornog koda (tar.gz) te prevođenje i instalacija

```
$ wget <URL do arhive Perl paketa>  
$ tar xzf <ime arhive>  
$ cd <direktorij>  
$ perl Makefile.PL  
$ make; make test  
$ make install
```

- Korištenje CPAN ljuske

```
$ perl -MCPAN -e shell
```

Primjer korištenja paketa (1)

- Za korištenje paketa potrebno ga je uključiti na početku skripte/programa

```
$ cat salji_mail.pl  
#!/usr/bin/perl
```

```
use Net::SMTP;
```

```
...
```

- Moduli dolaze s API dokumentacijom
 - Na Web-u u sklopu CPAN arhive
 - man stranice (man Net::SMTP)

Primjer korištenja paketa (2)

- Nakon uključivanja, koriste se objekti/potprogrami paketa

```
# Kreiranje objekta
$smtp = Net::SMTP->new('webmail.fer.hr');

# Tko šalje (mail) i kome se šalje (to)
$smtp->mail($ENV{USER});
$smtp->to('korisnik@gmail.com');

# Poruka (zaglavlje + tijelo poruke)
$smtp->data();
$smtp->datasend("To: postmaster\n");
$smtp->datasend("\n");
$smtp->datasend("A simple test message\n");
$smtp->dataend();

$smtp->quit;
```