

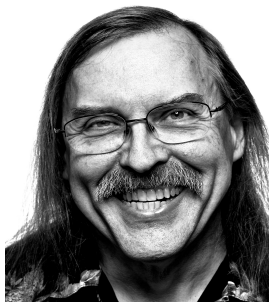
Perl - sjajno rešenje za analitičke probleme

Aleksandar Vračarević, Jovan Marić,
Petar Mičić, Tatjana Radovanović

Matematički fakultet, Univerzitet u Beogradu

May 8, 2019

- 1 Nastanak i razvoj
- 2 Osnovna namena, svrhe i mogućnosti
- 3 Osnovne osobine
- 4 Šta je ono što je važno i karakteristično?
- 5 Primer koda
- 6 Podržane paradigme
- 7 Najpoznatija okruženja
- 8 Instalacija i pokretanje



Larry Wall 1954- programer lingvista

- 1987 - Perl 1.0.
- Obrada teksta i izrada izveštaja na lakši način(nedostaci alata poput awk, sed i grep)
- Težnja da Perl konstanto napreduje i evoluira

- Obrada teksta
 - osnovna namena, najpoznatija namena
 - regularni izrazi su gradjani prvog reda ²
 - veliki broj ugradjenih funkcija
- Web programiranje
- Bezbednost - taint mode kao sigurnosta mera omogućava korišćenje Perl u brojnim serverskim aplikacijama ¹

- PHP, jedan od najpopularnijih programa za veb, preuzeo dosta funkcionalnosti Perl. Sličnosti se vide u sintaksi, nazivima promenljivih, evaluacija niski itd.
- Ruby je još jedan jezik koji preuzima obradu teksta i regularne izraze iz Perl. Online arhiva po ugledu na CPAN

- praktičnost više nego lepota
- najbolje iz C-a, awk-a, sed-a i grep-a
- interpretiran jezik - program se najpre prevodi u sintaksno drvo koje se nakon toga izvršava
- **slabo tipiziran jezik**
- brojanje referenci kao mehanizam upravljanja memorijom¹
- tri tipa ugraenih podataka:
 - skalari
 - nizovi skalara
 - asocijativni nizovi skalara ili heševi (eng. *hashes*)
- **dinamička alokacija**
- tumačenje promenljivih u odnosu na kontekst
 - `my @niz = ('Zdravo', 'svete');`
 - `my @elementi = @niz;`
 - `my $br_el = @niz`

¹nalik na C++ `std::shared_pointer`

- Velika Perl Arhiva (CPAN)
 - obimno skladište Perl programa
 - preko 18 hiljada modula, 12 hiljada programera
 - moduli koji nisu standardni
 - CPAN.pm modul
 - DBI, DateTime, XML::Parserer, Text::CSV...



Figure: Logo CPAN skladišta

Specifičnosti Perl-a (1)

- Parsiranje tekstualnog sadržaja
 - njegova glavna namena: parsiranje tekstualnog sadržaja i generisanje izveštaja
 - primer vrlo jednostavnog načina da izračunamo broj pojavljivanja reči u tekstu

```
while (my $line = <$fh>) {  
    chomp $line;  
    foreach my $str (split /\^[A-Za-z]+$/, $line) {  
        $count{$str}++;  
    }  
}
```


Specifičnosti Perl-a (2)

- Jednostavno korišćenje regularnih izraza
 - mehanizmi nasledjeni od awk, grep i sed
 - operatori `=~` i `!~`
 - podudaranje: `$str =~ m/regex/`, vraća 1 ako dodje do podudaranja, 0 inače
 - zamena: `$str =~ s/regex/substitution/`, rezultat je nova niska
 - translacija: `$str =~ tr/not_just_regex/substitution`, može da podrži i obične niske

Primer koda

```
sub zbir {  
  my(@a) = @_;  
  $s = 0;  
  for(my $i = 0; $i <= @a; $i++) {  
    $s = $s+@a[$i];  
  }  
  return $s;  
}  
  
@niz = (1, -2, -3, -4, 5);  
$br_el = @niz;  
print "Broj elemenata u nizu \(@niz\) je $br_el.\n";  
print "Indeks poslednjeg elementa je $#niz.\n";  
$suma = zbir(@niz);  
if($suma >= 0) {  
  $abs = $suma;  
}  
else {  
  $abs = -$suma;  
}  
print "Apsolutna vrednost zbira elemenata je $abs.\n";
```

Perl je jezik širokog spektra mogućnosti, te podržava neke od najzastupljenijih paradigmi kao što su:

- **proceduralna** - prirodna paradigma s obzirom na uticaj C-a
- **funkcionalna** - podržano je prosleivanje funkcija drugim funkcijama
- nije potrebna implementacija zasebnog programa za tumačenje lambda računa:
 - $\lambda_x.B \iff \text{sub}\{my\$x = \text{shift}; B\}$
 - $(PQ) \iff \$P- > (\$Q)$
- **objektno orijentisana** - omogućeno kreiranje i manipulisanje objektima
- potrebna **new**² podrutina za instanciranje objekata
- polimorfizam se postiže korišćenjem ključne reči **bless**
- nasleivanje se ostvaruje pomoću specijalog @ISA niza

²najčešće, mada je moguće nazvati je proizvoljno

- **Catalyst**

- okruženje za brz razvoj aplikacija zasnovanih na MVC arhitekturi³
- catalyst.pl ImePrograma pravi skelet web aplikacije
- klase Request i Response se brinu o komunikaciji izmeu klijenta i servera

- **Dancer**

- napravljen po uzoru na Ruby radni okvir Sinatra
- slično se koristi kao Catalyst:
- dancer -a ImePrograma
- jednostavna sintaksa se svodi na navoenje HTTP zahteva, URL-ova i metoda za obradu zahteva ka zadatim URL-ovima

- **Moose**

- radni okvir za apstrahovanje objektno orijentisanih koncepata
- konstruktori i pristupni metodi se implicitno prave te se programer štiti od detalja implementacije

³detaljan opis MVC arhitekture na sledećoj [adresi](#)

Instalacija i pokretanje

- Linux

- `sudo apt-get install perl`
- `curl -L http://xrl.us/installperlnix | bash`
- `perl -v`
- Pokretanje: `perl putanja_do_fajla`

- Windows

- Strawberry Perl
- Padre, the Perl IDE/editor
- `cpan App::cpanminus`
- `perl -v`
- Pokretanje: iz editora ili `perl putanja_do_fajla`



[1] Programming Perl, Third Edition (2000)

Wall, Larry, Tom Christiansen and Jon Orwant
Security (23), 551 – 584.



[2] Perl Programming Documentation (2000)

on line:

<https://perldoc.perl.org/perl5120delta.html#REGEXPs-are-now-first-class> Accessed:
2019-04-03.



Advanced PERL Programming (1997)

Srinivasan, Sriram

Object-Oriented Programming (7).



Perl, the first postmodern programming language (1999)

Wall, Larry

Linux World Conference, San Jose CA

Hvala na pažnji!