|  |
| --- |
|  |
| **NASLOV DOMAĆEG ZADATKA \*** |
| Perl programski jezik |

|  |  |
| --- | --- |
| **PROLOG** | |
| **Autor\*** | *Milan Hakaj* |
| **Kategorizacija\*** | IT2008-IPT-SCRP-STRL-V-V02 |
| **Nizo težine\*** | Osnovni |
| **Ko radi ?\*** | OAS student |

|  |
| --- |
| **SADRŽAJ** |
| **Naslov sekcije** |
| ***Uvod*** |
| **Sadržaj (telo) sekcije** |
| Perl programski jezik je jezik opšte namjene koji je razvio Larry Wall 1987 godine. Perl vuče svoje korijene iz drugih jezika kao što su C, sed, awk i Unix shell. Perl je postao važan jezik za WWW razvoj, obradu teksta, Internet usluge, mail filtriranje, programiranje, i svaki drugi zadatak koji zahtijeva portabilna i lako rješiva rješenja.  Programi napisani u Perlu se nazivaju perl skripte dok se termin perl program odnosi na program koji poziva odnosno izvršava perl skirpte. Perl je implementiran kao interpretirani jezik. Otuda izvršavanje perl skripti ima tendenciju da koristi više vremena nego neki C program. Sa druge strane, računari su sve brži i brži a pisanje perl programa može uštedeti znatno puno vremena.  **Karaktetistike**   * Perl je interpretirani jezik. To znači da čim korisnik napiše program, može ga odmah pokrenuti. Ne postoji obavezna faza prevođenja ili kompilacije. Perl programi se mogu izvoditi na Unix, Windows NT, MacOS, DOS, OS / 2, VMS i Amiga OS. * Perl je kolaborativni jezik. CPAN (engl. Comprehensive Perl Archive Network) softverska arhiva sadrži besplatne Perl alate napisne zajednički čime se štedi vrijeme. * Perl je besplatan, izvorni kod i prevodilac su dostupnii bez naknade. * Perl je brz. Perl interpreter je pisan u C jeziju i nakon mnogih optimizacija je postao vrlo brz.   Perl je danas ne samo programski jezik već i vrlo aktivna zajednica programera i korisnika. Odlikuje ga kvalitetan repozitorij gotovih programskih rješenja. Perl je prema svojim karakteristikama objektno orijentisani programski jezik opšte namjene s naglaskom na funkcionalnost, proširivost te relativno lako učenje jezika. Perl je bio jezik izbora za razvoj WWW aplikacija sredinom 90-ih godina. Od samih svojih početaka to je jezik UNIX i Linux sistemskih administratora koji ga koriste u svakodnevnom radu prvenstveno za automatizaciju procesa. Danas postoji i čitav niz korisničkih komercijalnih aplikacija pisanih u Perlu.  Perl prevodilac ima dvostruku namjenu: prevođenje izvornog koda u međuoblik pogodan za neposredno izvođenje (engl. code compilation) te samo izvođenje koda (engl. code execution).  Obje funkcije dostupne su u svakom trenutku, odnosno Perl omogućava kreiranje novih Perl programa odnosno Perl funkcija u toku samog izvođenja. Time je Perl blizak i funkcionalnim programskim jezicima kao što su Smalltalk i Haskell.  Trenutna verzija Perla koja je danas u širokoj upotrebi (Perl 5) ipak sadrži previše već ponešto zastarjelih programerskih tehnika, a zbog željene potpune kompatibilnosti sa starijim inačicama iz 80-ih godina ima izvjesnih nedostataka. Ovdje je potrebno napomenuti da je današnji stil programiranja u Perlu daleko od stila iz sredine 90-ih. Naglasak je prije svega na korištenju gotovih dobro testiranih korisničkih biblioteka dostupnih preko već spomenute CPAN arhive.  Premda je Perl na glasu kao "kriptičan" jezik "hackera", uz nešto profesionalne discipline te pridržavanja određenih pravila programiranja, moguće je pisati vrlo pregledne i uredne programe  U gornjem primjeru vidimo da je Perl kod bitno kraći, nema deklaracije varijabli, a tip varijabli se određuje dinamički (ako se koristi operator **+**, a sadržaj varijable je numerički, interpreter izvršava operaciju sumiranja bez obzira što je varijabla $s inicijalno definirana kao **string**. Installation of Perl language on different OSsInstalacija Perla na različitim OS S obzirom da je za pokretanje Perl skripti neophodna Perl instalacija na računaru, u ovom delu vežbi ćemo proći kroz instalaciju nad različitim vrstama operativnih sistema. S obzirom da većina Unix odnosno Linux sistema već podržava Perl – potrebno je proveriti da li je on već instaliran. Ovo se radi izvršavanjem sledeće komande preko terminala:  **perl –v** Perl on Unix like OSs Install 'make' through your package manager. You can then use all of the tools mentioned above. Install Perl on Linux First, you need a copy of Perl source bundle and you can download a stable version from the site "http://www.perl.com". At the moment I am writing these words, I found available the archive perl-5.8.8.tar.gz. You also need an ANSI C compiler, but don’t worry about this, Perl’s configuration program will check for one. If it will not find it, you can install a prebuilt version or install the C compiler supplied by your OS vendor, or for a free C compiler try "http://gcc.gnu.org/".Now, get a command prompt and enter the following commands:  **gunzip perl-5.8.8.tar.gz**  **tar xf perl-5.8.8.tar**  and wait the end of decompression. If you don’t have the gunzip decompression program, you can download gzip bundle from http://www.gnu.org.The next step is to launch the configure program of the Perl bundle. The installation shell script will examine your system and ask you questions to determine how perl5 package should be installed. If you want to bypass all the questions and use the computed defaults or the previous answers if there was already a config.sh file, you can run "Configure –d":  **cd perl-5.8.8**  **sh Configure –d**  After the Perl configuration step, you must build Perl by typing in the following command:  **make**  After this step, you can verify Perl’s installation, running again the following command at the command prompt:  **perl –v** Install Perl on Windows (32 and 64 bit)Perl on Windows (Win32 and Win64) **Strawberry Perl** is an open source binary distribution of Perl for the Windows operating system. It includes a compiler and pre-installed modules that offer the ability to install XS CPAN modules directly from CPAN. cpanm can be installed by running cpan App::cpanminus.  **ActiveState** provide a binary distribution of Perl (for many platforms), as well as their own  the Perl Package Manager (PPM). Some modules are not available as ppm's or have reported errors on the ppm build system, this does not mean they do not work. You to use the cpan script to build modules from CPAN against ActiveState Perl  The free ActivePerl binary distribution includes core Perl, popular modules, the Perl Package Manager (PPM), and complete documentation. You can download it from "http://www.activestate.com" and install it on your computer – the distribution is self-installing and the installation process is just the same as the one you used for other similar windows applications. ActivePerl Glavni izvor ActivePerl distribucije je sajt http://www.activestate.com/ koji uključuje:   * **Perl for Win32**   Binary for the core Perl distribution   * **Perl for ISAPI**   IIS plug-in for use with ISAPI-compliant web server   * **PerlScript**   ActiveX scripting engine   * **Perl Package Manager**   Manager for Perl modules and extension |
| **Naslov sekcije** |
| ***Perl Interpreter*** |
| **Sadržaj (telo) sekcije** |
| Perl izvršna datoteka, obično instalirana u **/usr/bin** ili **/usr/local/bin** ili **/usr/share/man/man1** na računaru, naziva se Perl interpreter.  **[root@localhost~] # whereis perl**  perl: /usr/bin/perl /usr/share/man/man1/perl.1.gz  Svaki Perl program mora proći kroz Perl interpreter kako bi se mogao izvršiti. Prva linija u mnogim Perl programa je nešto poput:  **#!/usr/bin/perl**  Za Unix sistem, ovaj #! (hash-bang ili shebang) linija kaže ljusci da traži /usr/bin/ perl program i onda mu predaje ostatak programa za izvršenje. Ponekad se može vidjeti put (engl. pathname) za Perl kao /usr/local/bin/perl. Može se vidjeti perl5 umjesto Perl na mjestima koja još uvijek ovise o starijim verzijama Perl.  Dakle, šta Perl interpreter radi? Ona interno prevodi program, a zatim ga odmah izvršava. Perl je obično poznat kao interpretirani jezik, ali to nije strogo istinito.  Budući da interpreter zapravo prevodi program u bytecode kod prije njegovog izvršenja, često se naziva interpreter /prevodilac. Iako se prevedeni program ne pohranjuje kao datoteka, Perl verzija 5.,005 uključuje radni verziju samostalnog Perl prevodioca.  Kada se napiše Perl program, treba dati točnu #! liniju na vrhu scipta, čime postaje izvršni program pomoću chmod +x naredbe i onda ga pokrećete. Za 95% Perl programera u ovom svijetu, to je sve što je potrebno. |
| **Naslov sekcije** |
| ***Perl hello world*** |
| **Sadržaj (telo) sekcije** |
| S obzirom da će u ovom delu vežbi biti napravljen prvi Perl program, ne bi bilo loše znati nekoliko ključnih stvari o pisanju ovakvih skripti:   * Perl skripte se sastoje od niza izjava * Kada se program pokrene, skripta se izvršava počevši odozgo pa sve do poslednje linije koda. Takođe, treba imati u vidu da postoje slučajevi kada ovo pravilo ne važi – u slučaju pozivanja potprograma. * Svaka izjava se završava tačka-zarezom * Izjave se mogu javiti preko nekoliko linija koda * Razmaci (spaces, tabs I newlines) su uglavnom ignorisani u Perl skriptama   Kreiranje novog perl programa otpočinjemo kreiranjem fajla sa **.pl** ekstenzijom. Potrebno je razumeti da u nekim okruženjima nije neophodno da fajl bude ove ekstenzije ali se ona preporučuje zbog dobre prakse. Unosom narednog dela koda kreirali smo prvi perl program:  **helloworld.pl**   |  | | --- | | 1 *#! /usr/bin/perl*  2  3 **use** strict;  4 **use** warnings;  5 **use** 5.010;  6  7 say 'Hello from Perl 5!'; *# Prints Hello from Perl 5!* |   Dakle, prva linija koda shebang govori ljusci o tome gde se Perl interpreter nalazi te joj se predaje skripta koju potom interpreter izvršava. Naredne linije (3, 4 i 5) predstavljaju perl pragmu. Pragma je modul koji utiče na neke aspekte tokom vremena kompilacije/vreme izvršavanja Perl skripti. U ovom primeru to su **strict** i **warnings**. Oni na neki način govore Perlovom internom kompajleru da bi on trebao prisiliti skriptu da koristi neku od dobrih programerskih praksi za ostatak koda.  U ranijim verzijama perla (pre perl 5 verzije) ove komandne opcije, kojih Perl ima mnogo, su se deklarisale u nastavku same shabang linije. Jedna od najpoznatijih je **–w** koja uključuje leksička upozorenja:  **#! /usr/bin/perl -w**  **Pokretanje Perl skripte**  Pokretanje perl skripte se može izvršiti preko terminala pomoću sledeće komande:  **perl helloworld.pl**  Pokretanjem ove skripte dobićemo rezultat kao što je definisano u prethodnom primeru (Hello from Perl 5!). Takođe, može se primetiti da je izlaz napisan u novoj liniji – ovo je omogućeno korišćenjem Perl 5 metode **say**. Dakle, za izlaz se može koristiti i print metod, međutim, u tom slučaju bi se na kraju linije morao naglasiti newline pomoću već dobro poznatog „**\n**” karaktera.  Takođe, perl skripta se može pisati direktno u okviru konzole za kratke programe. Na primer:   |  | | --- | | **perl -e "print \"Hello, world\n\""** |   Još jedan od načina na koji se može pisati Perl skripta je preko standardnog input stream-a korišćenjem perl komande:   |  | | --- | | * **Perl**   **print “Hello from console! \n”;**  ^Z |   Potrebno je primetiti karakter ^Z (ctrl+z) koji predstavlja indikator za kraj fajla. Problem sa kojim se susrećemo kod ovakvog načina pisanja koda jeste taj da je ovaj kod samo privremeno sačuvan te će nakon izvršavanja biti uklonjen.  **Pokretanje Perl skripte kao izvršni fajl**  Unix-bazirani sistemi ne razumeju automatski da bi Perl skripta (ona i jeste običan tekstualni fajl povrh svega) trebala da bude izvršna. Kao rezultat toga, potrebno je promeniti privilegije za konkretan perl fajl pomoću komande **chmod**.   |  | | --- | | **chmod +x helloworld.pl** |   U ovom slučaju, komanda chmod pretvara skriptu helloworld.pl u izvršnu tako što joj dodaje **executable** privilegije. Nakon ovoga, skriptu možemo pokrenuti pomoću naredne komande iz terminala:   |  | | --- | | **./helloworld.pl** |   **Pokretanje Perl skripte u veb čitaču**  Kako bi pokretanje skripti u veb čitaču mogli uspešno da izvedemo, najpre se moramo upoznati sa HTTP header konceptom koji koristi Perl za rad sa veb čitačima. Perl koristi jednu čudnu liniju koda koja govori interpreteru da je sadržaj, za razliku od ostalih skripti, sada baziran na HTML-u.  Dakle, za rad sa veb čitačem potrebno je dodati: **print "content-type: text/html \n\n";**  Potom fajl sačuvati na nekom veb serveru koji podržava Perl kao što je Apache XAMPP (ukoliko se radi o XAMPP-u, unutar ovog direktorijuma postoji pod-direktorijum cgi-bin u koji je potrebno sačuvati ovaj fajl.  Nakon ovoga, fajlu pristupamo preko adrese: localhost/cgi-bin/filename.pl  **Komentari**  Komentari u Perlu počinju hesh znakom (**#**) na početku ili kraju izjave:   |  | | --- | | 1 *#! /usr/bin/perl*  2  3 **use** strict;  4 **use** warnings;  5 **use** 5.010;  6  7 *# my first comment*  8 say "Hello world!";  9 say "Welcome to CS324"; *# another comment* |   **Blok komentari**  Kako bismo komentarisali blokove teksta odnosno koda možemo koristiti POD (Plain Old Doc) tagove:   |  | | --- | | 1 *#! /usr/bin/perl*  2  3 **use** strict;  4 **use** warnings;  5 **use** 5.010;  6  7 *=begin comment*  8 *This content is commented out.*  9 *It may span many lines.*  10 *print "This statement wonâ€™t be executed by Perl\n";*  11 *=end comment*  12 *=cut* | |
| **Naslov sekcije** |
| ***Perl varijable*** |
| **Sadržaj (telo) sekcije** |
| Promenljive su ništa drugo do rezervisanih delova memorijskih lokacija u kojima se čuvaju vrednosti. To znači da se, kada se jednom kreira promenljiva, rezevriše mesto u memoriji.  U zavisnosti od tipa podataka promenljive, interpreter alocira određeni deo memorije i odlučuje šta može biti smešteno u rezervisani deo. Prema tome, dodeljivanjem različitih tipova promenljivoj mogu se definisati one tipa integer, string, itd.  Perl ima 3 različite vrste promenljivih u skladu sa čime ćemo koristiti različite tipove Perl promenljivih. To su:   * Scalars * Arrays * Hashes   Skalarnim promenljivim prethodi znak $ i u njih se mogu smestiti broj, string ili referenca.  Array promenljivim prethodi znak @ i u njih se mogu smestiti uređene liste skalara.  Hash promenljivim prethodi znak % i u njih se mogu smestiti setovi parova ključ - vrednost.  **Skalarne promenljive**   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $age = 25; *# An integer assignment*  4 $name = "John Paul"; *# A string*  5 $salary = 1445.50; *# A floating point*  6  7 **print** "Age = $age\n";  8 **print** "Name = $name\n";  9 **print** "Salary = $salary\n"; |   **String**  Stringovi su skalari. Ne postoji limit kada se govori o veličini stringa, bilo koje veličine karaktera, simbola ili reči koje mogu da se nalaze u stringu. Za definisanje stringova se mogu koristiti **single** i **double** quotation ali takođe i pomoću **q** podfunkcije.   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $single = 'This string is single quoted';  4 $double = "This string is double quoted";  5 $userdefined = q^Carrot is now our quote^; |   **Formatiranje stringova pomoću specijalnih karaktera:**   |  |  | | --- | --- | | **Character** | **Description** | | \L | Transform all letters to lowercase | | \l | Transform the next letter to lowercase | | \U | Transform all letters to uppercase | | \u | Transform the next letter to uppercase | | \n | Begin on a new line | | \r | Applys a carriage return | | \t | Applys a tab to the string | | \f | Applys a formfedd to the string | | \b | Backspace | | \a | Bell | | \e | Escapes the next character | | \0nn | Creates Octal formatted numbers | | \xnn | Creates Hexideciamal formatted numbers | | \cX | Control characters, x may be any character | | \Q | Do not match the pattern | | \E | Ends \U, \L, or \Q functions |   Specijani karakteri su jedino mogući ukoliko koristimo double quoting. U single quote slučaju oni ne bi bili prihvaćeni.  Konkatenacija dva stringa se može uraditi pomoću **dot** operatora:   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $word\_a = 'Hello';  4 $word\_b = "World";  5 $result = $word\_a . ' ' . $word\_b; |   String se može se umnožiti nekoliko puta pomoću **repetition** operatora:   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $var = 'Repetition time!';  4 **print** $var x 3; |   U Perlu se takođe može koristiti **backslash** karakter za eskejpovanje bilo kog karaktera koji bi mogao ugroziti kod. Na primer:   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $result = "This is " . "\"number\"";  4 **print** "$result"; |     Napredna interpoliacija u ovom kontekstu predstavlja koncept zamenjivanja promenljive sa njenom vrednošću.   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $name = "Milan";  4 **print** "Hello, this is " . $name . "."; |   **Brojevi**  Perl interno čuva brojeve ili kao signed integers ili kao floating point vrednosti sa double precision. Numerički literali su definisani u jednom od sledećih formata   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 $a = 1234; *# decimal integer*  4 $b = 0b1110011; *# binary integer*  5 $c = 01234; *# octal integer*  6 $d = 0x1234; *# hexadecimal integer*  7 $e = 12.34e-56; *# exponential notation*  8 $f = "-12.34e56"; *# number specified as a string*  9 $g = "1234"; *# number specified as a string* |   Neki od osnovnih aritmetičkih operatora su:   |  |  |  |  | | --- | --- | --- | --- | | **Operator** | **Primer** | **Rezultat** | **Definicija** | | + | 7+7 | 14 | Addition | | - | 7-7 | 0 | Subtraction | | \* | 7\*7 | 49 | Multiplication | | / | 7/7 | 1 | Division | | \*\* | 7\*\*7 | 823543 | Exponents | | % | 7%7 | 0 | Modulus |   Sa ovim operatorima možemo koristiti brojeve kako bismo izvršili neke od jednostavnih matematičkih operacija.  **Array promenljive**  Niz je promenljiva koja skladišti uređenu listu skalara. Perl array promenljivama prethodi znak @. Za pristupanje konkretnom elementu koristi se znak $ propraćen indeksom elementa u uglastim zagradama.   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 @ages = (25, 30, 40);  4 @names = ("John Paul", "Lisa", "Kumar");  5  6 **print** "\$ages[0] = $ages[0]\n";  7 **print** "\$ages[1] = $ages[1]\n";  8 **print** "\$ages[2] = $ages[2]\n";  9 **print** "\$names[0] = $names[0]\n";  10 **print** "\$names[1] = $names[1]\n";  11 **print** "\$names[2] = $names[2]\n"; |   **Hash promenljive**  O ovoj vrsti Perl promenljivih će biti reči u nekim od narednih predavanja. |
| **Naslov sekcije** |
| ***Variable Context*** |
| **Sadržaj (telo) sekcije** |
| Perl tretira neke promenljive različito u zavisnosti od konteksta odnosno situacije gde je konkretna promenljiva korišćena. Na primer:   |  | | --- | | 1 *#!/usr/bin/perl*  2  3 @names = ('John Paul', 'Lisa', 'Kumar');  4  5 @copy = @names;  6 $size = @names;  7  8 **print** "Given names are : @copy\n"; *# => John Paul Lisa Kumar*  9 **print** "Number of names are : $size\n"; *# => Number of names are : 3* |   U ovom slučaju @names je niz koji se koristi u dva različita konteksta. Prvo se vrši kopiranje niza u drugi tako da se odvija prebacivanje liste iz postojećeg u novokreirani niz. Perl to posmatra kao list context. Drugi slučaj jeste kada smo pokušali da ovaj niz smestimo u promenljivu koja je tipa skalar. U tom slučaju, Perl će vratiti samo broj elemenata s obzirom da će smatrati da se radi o kontekstu skalara.  Sledeća tabela ilustruje različite slučajeve odnosno različite vrste konteksta:   |  |  | | --- | --- | | **1** | **Scalar:** Assignment to a scalar variable evaluates the right-hand side in a scalar context. | | **2** | **List:** Assignment to an array or a hash evaluates the right-hand side in a list context. | | **3** | **Boolean:** Boolean context is simply any place where an expression is being evaluated to see whether it's true or false | | **4** | **Void:** This context not only doesn't care what the return value is, it doesn't even want a return value. | | **5** | Interpolative: This context only happens inside quotes, or things that work like quotes. | |

|  |  |
| --- | --- |
| **RELEVANTNI LINKOVI** | |
| **Opis** | **URL** |
| Primer 1 | http://www.perl.com/pub/2000/10/begperl1.html |
| Primer 2 | http://www.perl.org/books/beginning-perl/ |

|  |  |
| --- | --- |
| **REČNIK** | |
| **Termin** | **Opis** |
| *Perl* | *Practical Extraction and Reporting Language* |
| *CPAN* | *Comprehensive Perl Archive Network* |

|  |
| --- |
| **DRUGI PREDLOŽENI SADRŽAJI AUTORA** |
| **Naslov** |
|  |
| **Telo** |
|  |