

Skriptni jezici – zadaci za 2. ciklus laboratorijskih vježbi

travanj 2020.

1 Uvod

U okviru drugog ciklusa laboratorijskih vježbi utvrđuju se i praktično primjenjuju osnove programskog jezika Perl.

Studenti su dužni pripremiti se za laboratorijske vježbe samostalnim rješavanjem niza jednostavnih zadataka. Za zadatke koji obavljaju operacije nad datotekama (pretraživanje sadržaja, promjena imena i slično) potrebno je pripremiti datoteke prikladne za ispitivanje i demonstraciju rada programa.

Preduvjet za obavljanje laboratorijske vježbe je predaja (*upload*) datoteka s rješenjima zadataka kroz sustav Ferko (<https://ferko.fer.hr/ferko/>). Prilikom postavljanja datoteka treba poštovati upute o imenovanju datoteka, kao i o uvjetima pokretanja skripti (navođenje parametara i slično).

U terminu laboratorijskih vježbi student piše kratku provjeru znanja (blic), samostalno rješava jedan programski zadatak, te brani svoja rješenja zadataka za pripremu pred asistentom.

1.1 Protokol odrade laboratorijske vježbe

Ukratko, protokol odrade laboratorijske vježbe je:

1. student je dužan riješiti postavljene zadatke i datoteke s rješenjima predati kroz sustav Ferko;
2. student je dužan doći na laboratorijske vježbe u svom terminu;
3. na početku termina laboratorijskih vježbi studenti pišu kratku provjeru znanja (kviz-pitanja s ponuđenim odgovorima; točan odgovor nosi 0.5, a pogrešan -0.125 bodova);
4. tijekom laboratorijskog termina studenti dobivaju zadatak koji trebaju riješiti na licu mjesta (izlazni test). Zadatak se rješava na računalu, te se rješenje predaje kroz sustav Ferko. **Voditi računa o imenovanju skripte, obavezno zaključati predaju!**
5. tijekom laboratorijskog termina student treba pokazati svoja rješenja zadataka za pripremu asistentu, za što ga asistent ocjenjuje;

1.2 Resursi u laboratoriju

Vježbe se obavljaju u fakultetskim laboratorijima. Na računalima je instalirana okolina Cygwin, s potrebnim alatima (`bash`, `perl`, ...).

Zadatak 1

Napišite Perl skriptu koja će od korisnika zatražiti (i učitati) niz znakova i broj ponavljanja (n). Učitani znakovni niz treba ispisati n puta (svaki put u novom redu).

Uputa za upload: Skriptu nazvati `zadatak1.pl`.

Zadatak 2

Napišite Perl skriptu koja će učitati niz brojeva u listu, te izračunati i ispisati aritmetičku sredinu učitanih brojeva.

Uputa za upload: Skriptu nazvati `zadatak2.pl`.

Zadatak 3

U repozitoriju na stranici predmeta nalaze se dvije *log*-datoteke jednog web poslužitelja. To su tekst datoteke koje se generiraju svakoga dana, a datum je sadržan u imenu datoteke. Svaki redak odgovara jednom pristupanju poslužitelju.

Napisati skriptu u Perlu koja će za svaki sat u danu ispisati broj pristupa poslužitelju. Kao argumenti naredbenog retka pri pozivu se navode imena log datoteka koje treba analizirati. Ako se ne navede niti jedna datoteka, skripta podatke treba čitati sa standardnog ulaza (sjetite se operatora `<>`). Ispis oblikovati prema primjeru:

```
bash-3.2$ ./zadatak3.pl localhost_access_log*.txt
```

```
Datum: 2008-02-24
sat : broj pristupa
```

```
-----
00 : 99
01 : 62
02 : 15
...
21 : 599
22 : 615
23 : 593
```

```
Datum: 2008-02-25
sat : broj pristupa
```

```
-----
00 : 378
01 : 68
02 : 48
...
```

Uputa za upload: Skriptu nazvati `zadatak3.pl`.

Zadatak 4

U datoteci su zapisani podaci o terminima laboratorijskih vježbi pojedinih studenata, te vrijeme njihove predaje (zaključavanja) izlaznog testa, u obliku:

```
JMBAG;Prezime;Ime;Termin;Zaključano
0036438919;Bagarić;Magdalena;2011-03-14 08:00 11:00 A209;2011-03-14 08:45:02
0036433049;Bajac;Darko;2011-03-14 08:00 11:00 A209;2011-03-14 08:48:19
...
0036436684;Lombarović;Mladen;2011-03-14 11:00 14:00 A210;2011-03-14 12:08:26
0036325839;Matošić;Luka;2011-03-14 11:00 14:00 A210;2011-03-15 11:49:26
...
```

Napišite skriptu u Perlu, koja će za svakog studenta provjeriti je li zaključao svoj izlazni test unutar prvih sat vremena laboratorijskog termina. Ispisati podatke o studentima za koje taj uvjet nije ispunjen, kao u primjeru. Ime datoteke se navodi kao argument pri pozivu skripte. Ako se ne navede ime datoteka, skripta podatke treba čitati sa standardnog ulaza.

```
bash-3.2$ ./zadatak4.pl labosi-zakljucano.csv
```

```
0036436684 Lombarović Mladen - PROBLEM: 2011-03-14 11:00 --> 2011-03-14 12:08:26
0036325839 Matošić Luka - PROBLEM: 2011-03-14 11:00 --> 2011-03-15 11:49:26
0036439146 Paar Marko - PROBLEM: 2011-03-14 11:00 --> 2011-03-14 12:16:22
...
```

Uputa za upload: Skriptu nazvati `zadatak4.pl`.

Zadatak 5

Napišite skriptu u Perlu koja će iz datoteke čije je ime navedeno kao argument pri pozivu skripte učitati podatke o rezultatima studenata. U prvom retku datoteke zapisan je niz brojeva, odvojenih znakom ";", koji predstavljaju faktore s kojima se svaka komponenta ocjene množi. Slijede retci s podacima o svakom pojedinom studentu: matični broj, prezime i ime kao jedno polje, te niz brojeva koji predstavljaju ostvarene rezultate pojedinih komponenti ocjene studenta. Pojedina polja odvojena su znakom ";". Ako neka bodovna komponenta nedostaje, označena je znakom "-". Primjer oblika datoteke:

```
# udjeli komponenti
0.15;0.20;0.30;0.10;0.10;0.10;0.05
#
# rezultati studenta
0036438919;Bagarić;Magdalena;91.5;90.75;88.25;100;87.5;87.5;90
0036439578;Bajer;Goran;64.4;72;64.75;72.5;76.25;50;62
0036436684;Lombarović;Mladen;14.7;75.75;-;-;5;20;-
...
```

Skripta treba na temelju učitanih rezultata generirati rang-listu studenata, u kojoj će biti označen rang studenta, njegovo prezime i ime, te ukupni ostvareni broj bodova (dobije se zbrajanjem svih komponenti rezultata pomnoženih s pripadnim faktorima). Funkciji `sort` može se zadati blok naredbi ili potprogram koji obavlja usporedbu dva elementa liste koja se sortira. Za detalje pogledati dokumentaciju funkcije `sort`. Primjer pozivanja skripte:

```
bash-3.2$ ./zadatak5.pl rezultati.csv
```

Lista po rangu:

```
-----  
  1. Bagarić, Magdalena (0036438919) : 90.35  
  2. Pavlinović, Matija (0036427706) : 82.20  
  3. Bandalo, Danijel (0067400961)   : 72.94  
  ...  
 13. Lombarović, Mladen (0036436684) : 19.86  
  ...
```

Uputa za upload: Skriptu nazvati `zadatak5.pl`.

Zadatak 6

Napisati Perl skriptu koja će u datoteci koja se zadaje kao argument naredbenog retka izbrojati riječi sa zajedničkim prefiksom. Duljina prefiksa zadaje se kao zadnji argument naredbenog retka, a prethode mu imena datoteka (može ih biti i više). Ako se ne navede niti jedno ime datoteke, skripta treba tekst čitati sa standardnog ulaza.

U datoteci je običan tekst, a riječi su odvojene proizvoljnim brojem razmaka (uključujući i tabulatore i prelasku u novi red) te znakovima interpunkcije. Prilikom usporedbe riječi treba zanemariti razliku između velikih i malih slova. Lista prefiksa treba biti poredana po abecedi, a iza svakog prefiksa treba navesti broj pojavljivanja. U nastavku je ilustrirano ponašanje skripte.

Pretpostavimo da se sadržaj datoteke `notturno.txt`:

```
Noćas se moje čelo žari,  
noćas se moje vjede pote;  
i moje misli san ozari,  
umrijet ću noćas od ljepote.
```

```
Duša je strašna u dubini,  
ona je zublja u dnu noći;  
plačimo, plačimo u tišini,  
umrimo, umrimo u samoći.
```

Rezultat izvođenja skripte trebao bi biti sljedeći:

```
$ ./zadatak6.pl notturno.txt 4
```

```
čelo : 1  
dubi : 1  
duša : 1  
ljep : 1  
misl : 1  
moje : 3  
noća : 3  
noći : 1  
ozar : 1  
plač : 2  
pote : 1  
samo : 1
```

```
stra : 1
tiši : 1
umri : 3
vjed : 1
zubl : 1
žari : 1
```

Za zapis hrvatskih znakova prikladno je koristiti format UTF-8, te u ljusti podesiti varijablu `$LC_ALL` na vrijednost `hr_HR.UTF-8`:

```
export LC_ALL=hr_HR.UTF-8
```

Da bi Perl ispravno učitavao i ispisivao tako kodirane znakove, dovoljno je u skriptu dodati sljedeću naredbu:

```
use open '':locale';
```

Da bi se koristila lokalna leksikografska pravila sortiranja, treba dodati i naredbu:

```
use locale;
```

Uputa za upload: Skriptu nazvati `zadatak6.pl`.