


# Prvi kolokvijum – Popravni

## Veb programiranje

U okviru ovog zadatka potrebno je napraviti web aplikaciju u kojoj se prikazuju rang liste (izgled aplikacije prikazan je na slici 1). Ceo zadatak implementira se u okviru fajlova: *index.php*, *details.php*, *RankList.php* i *RankListItem.php*. U navedenim fajlovima, u okviru komentara, nalaze se TODO tekstovi, koji sugerišu šta gde treba implementirati. Pored toga, u folderu *templates* su dati svi HTML segmenti koji se nalaze u tekstu ovog zadatka. **Pomenute HTML segmente treba kopirati iz templejt fajlova, a nikako iz PDF-a.**



**Rang liste**

Top-1 Top-5 Top-10

IMDb - Najpopularniji filmovi na dan 10. 6. 2019.		
Rang	Naziv	Vrednost
1	The Shawshank Redemption	9.2
2	The Godfather	9.2
3	The Dark Knight	9
4	The Godfather: Part II	9
5	The Lord of the Rings: The Return of the King	8.9
6	Pulp Fiction	8.9
7	12 Angry Men	8.9
8	Schindler's List	8.9
9	The Lord of the Rings: The Fellowship of the Ring	8.8
10	Fight Club	8.8

IMDb - Najpopularnije serije na dan 10. 6. 2019.		
Rang	Naziv	Vrednost
1	Chernobyl	9.6
2	Planet Earth II	9.5
3	Band of Brothers	9.4
4	Breaking Bad	9.4
5	Planet Earth	9.4
6	Game of Thrones	9.3
7	Cosmos (1980)	9.2
8	Rick and Morty	9.2
9	Cosmos	9.2
10	Our Planet	9.2

TIOBE indeks za maj 2019.		
Rang	Naziv	Vrednost
1	Java	1
2	C	2
3	C++	3
4	Python	4
5	Visual Basic .NET	5
6	C#	6
7	JavaScript	7
8	SQL	8
9	PHP	9
10	Assembly language	10

RollingStones - Najpopularnije pesme svih vremena		
Rang	Naziv	Vrednost
1	Bob Dylan, 'Like a Rolling Stone'	1

ATP lista - 27. 5. 2019.		
Rang	Naziv	Vrednost
1	Novak Djokovic	12355

Slika 1.

Podaci o rang listama nalaze se u JSON fajlu *data.json*. Struktura podataka je sledeća: svaka rang lista ima svoje ime (*name*), svoj tip (*type*), informaciju o tome da li se pri prikazu sortira rastuće ili opadajuće (*dsc*) i svoje stavke (*items*). Tip (*type*) rang liste može biti jedna od sledećih vrednosti: m - rang lista je vezana za filmove, mu - rang lista je vezana za muziku, s - rang lista je vezana za sport i o - rang lista ne potpada ni u jednu od prethodnih kategorija. Vrednost *dsc* (skraćenica od *descending*) ključa može biti 1 ili 0, i to 1 ukoliko lista treba da bude sortirana opadajuće, a 0 ukoliko lista treba da bude sortirana rastuće. Svaka rang lista ima i svoje stavke koje su sortirane (primer: u rang listi koja prikazuje filmove, jedan konkretan film bi bio stavka te rang liste). Pozicija stavke u sortiranom poretku predstavlja rang te stavke. Stavka (*item*) rang liste ima dva podatka: naziv (*name*) i broj poena po kom će se stavke sortirati (*points*).

Rang lista modelirana je klasom *RankList*, dok je jedna stavka rang liste modelirana klasom *RankListItem*. U klasi *RankList* stavke se nalaze unutar polja *items* koje onda predstavlja već sortirani niz objekata klase *RankListItem*. Stranica *index.php* treba da prikaže sve rang liste koje postoje u pomenutom JSON fajlu, dok stranica *details.php* treba da sadrži detalje o jednoj datoj rang listi sa mogućnosti pretrage stavki. Konkretni zadaci koje treba implementirati su dati ispod.

**TODO 1** - Sve stavke jedne rang liste biće prikazane u HTML tabeli, i to tako da je svaka stavka jedan red u tabeli.

Stoga je potrebno implementirati metod *getHTML* klase *RankListItem* koji vraća jedan red tabele. Taj red onda treba da prikazuje podatke na sledeći način (ovaj HTML se nalazi u templejt fajlu *rank\_list\_item.txt*):

```
<tr>
    <td>%RANK%</td>
    <td>%NAME%</td>
    <td style="text-align: center;">%POINTS%</td>
</tr>
```

Crvene tokene zameniti konkretnim vrednostima imajući u vidu sledeća njihova značenja:

- RANK - rank koji je prosleđen kroz parametar metoda,
- NAME - naziv stavke,
- POINTS - broj poena koje stavka ima.

**TODO 2** - Unutar klase *RankList* potrebno je implementirati pravljenje već pomenute HTML tabele za tu datu rang listu. Implementaciju treba smestiti u metod *getListTableHtml* koji onda treba da vrati string sa pomenutim HTML kodom. Tabela treba da se pravi po sledećem HTML templejt kodu (koji se nalazi u templejt fajlu *rank\_list.txt*):

```
<table class="table %CLASS%">
    %HEADING%
    %ROWS%
</table>
```

Crveni tokeni predstavljaju sledeće stvari:

- CLASS - ima jednu od sledećih vrednosti:
  - **music** - ukoliko je tip rang liste *mu*,
  - **movies** - ukoliko je tip rang liste *mo*,
  - **sports** - ukoliko je tip rang liste *s*,
  - **other** - ukoliko je tip rang liste *o*,
  - u svim ostalim slučajevima je vrednost prazan string.
- HEADING - string kog vraća statički metod *getTableHeading* klase *RankListItem*. Ovaj string predstavlja zaglavlje tabele.
- ROWS - redovi tabele koji se dobijaju pozivom metoda *getHTML* klase *RankListItem* (implementiranog u stavci TODO 1). Ovaj metod treba da se poziva nad objektima iz polja *items* klase *RankList*. Međutim, ne treba uvek štampati sve stavke rang liste, već treba obratiti pažnju na parametar *count*. Ukoliko je taj parametar *null* onda treba štampati sve stavke, a ukoliko nije *null* onda treba štampati onoliko stavki koliko je prosleđeno kroz parametar *count*, i to počev od prve stavke niza *items*. (Napomena: rang stavke predstavlja njenu poziciju u nizu *items*, pri čemu se preferira da rang bude 1-indeksiran, a ne 0-indeksiran)

**TODO 3** - U *index.php* stranici je potrebno odštampati sve raspoložive rang liste. Rang liste se nalaze u nizu *rank\_lists* i potrebno ih je štampati pozivom metoda *getTopListHtml* klase *RankList*. Ovaj metod prima broj stavki koji će se u rang listi prikazati, i on za početak treba da bude jednak konstanti *TOP\_LIST\_DEFAULT\_COUNT*. (Napomena: u daljem tekstu ovog zadatka ćemo rang liste u kojima je prikazano samo prvih nekoliko stavki zvati *top liste*.)

**TODO 4** - Potrebno je odštampati dugmad kojima će da se menja broj stavki koje će se prikazivati u top listama. Naime, klikom na jedno dugme će se umesto `TOP_LIST_DEFAULT_COUNT` stavki (kako je naglašeno u stavci TODO 3), prikazati onoliko stavki koliko to dugme sugerije. Na slici 1 su ta dugmad prikazana neposredno ispod naslova stranice. Broj ovih dugmadi zavisi od maksimalnog broja stavki koje neka rang lista ima; obeležimo taj broj sa `X`. U tom slučaju je potrebno prikazati sledeću dugmad: Top-1, Top-5, Top-10, Top-50, Top-100, Top-500, Top-1000, Top-5000, Top-10000, ... , i tako sve dok broj desno od stringa "Top-" ne postane veći od `X`. Broj pored stringa "Top-" treba onda da odredi koliko će se stavki u svakoj top listi prikazati nakon klika na to dugme. Ovo implementirati upotrebom query stringova. (Napomena: prilikom implementacije ove stavke, moraće malo da se izmeni implementacija stavke TODO 3)

**TODO 5** - Klikom na jednu top listu sa `index.php` stranice, korisnik se šalje na `details.php` gde mu se prikazuje cela rang lista na koju je kliknuo. U ovoj stranici je potrebno implementirati search funkcionalnost. Ideja je da korisnik može da unese deo naziva neke stavke iz prikazane rang liste, i da mu se onda ispiše kog ranga je ta stavka. Primer se može videti na slici 2, a podstavke za implementaciju ove funkcionalnosti su date ispod.

IMDb - Najpopularniji filmovi na dan 10. 6. 2019.



Rang	Naziv	Vrednost
1	The Shawshank Redemption	9.2
2	The Godfather	9.2
3	The Dark Knight	9
4	The Godfather: Part II	9
5	The Lord of the Rings: The Return of the King	8.9
6	Pulp Fiction	8.9
7	12 Angry Men	8.9

Slika 2. Forma za pretragu, zajedno sa rezultatom pretrage (označeno crvenom bojom). Korisnik je ovde uneo string "pulp" u input polje, i nakon toga mu se ispisalo "pulp ima rang 6" zato što film "Pulp Fiction" zaista jeste na šestom mestu u listi.

**TODO 5.1** - Kao što je već pominjano, svaka rang lista se sastoji od niza stavki (polje `items` klase `RankList`). Podsetimo se da pozicija stavke u tom nizu predstavlja rang te stavke. Potrebno je implementirati metod `getRank` u klasi `RankList` koji prima search string, i vraća rang stavke koja zadovoljava taj search string. Implementacija se svodi na proveru da li neka od stavki u svom nazivu kao podstring ima search string. Rang prve stavke za koju se ovo ispostavi kao tačno se vraća kao rezultat izvršavanja ovog metoda (tj. ako više stavki zadovoljavaju taj search string, vratiti manji rang). Ukoliko nijedna stavka za podstring nema search string, metod vraća -1. Pri implementaciji može biti od pomoći PHP funkcija `strpos`<sup>1</sup>. (Napomena: dovoljno je implementirati lakšu verziju po kojoj je pretraga case sensitive)

**TODO 5.2** - Potrebno je napraviti HTML POST forme za unos search stringa, kao što je prikazano na slici 2.

<sup>1</sup> <https://www.php.net/manual/en/function strpos.php>

**TODO 5.3** - Potrebno je obraditi formu iz stavke TODO 5.2, tako da se korisniku nakon search-a prikaže tekst poput onog sa slike 2. Pri implementaciji koristiti metod *getRank* (iz stavke 5.1). Imati u vidu da se tekuća rang lista nalazi u promenljivoj *rank\_list*.