

Programski jezici

novembar 2005.

1. Događaji na rukometnoj utakmici između tima D i tima G (oštar faul nad D, faul na granici sedmerca nad D, faul unutar sedmerca nad D, prigovor sudiji od D, približno pasivna igra tima D, sigurno pasivna igra tima D, oštar faul nad G, faul na granici sedmerca nad G, faul unutar sedmerca nad G, prigovor sudiji od G, približno pasivna igra tima G, sigurno pasivna igra tima G) bivaju posmatrani od strane sudije i od strane navijača tima D. Sudija navija za tim G tako što (pseudoslučajno) u približno trećini situacija reaguje na događaj protežirajući G (njegove odluke su: sedmerac, deveterac, dva minuta isključenja, nastavak igre, proglašavanje pasivne igre). Navijač tima D posmatra igru (tj. događaje u igri), adekvatnim rečima hrabri svoje i istovremeno prati suđenje. Svaki put kada se dogodi da sudijska odluka ne odgovara odigranom događaju, navijač komentariše tu odluku i samu ličnost sudije. Napisati grafičku Java aplikaciju koja omogućuje da se događaji kreiraju pseudoslučajno sa vremenskim okidačem između 2 i 3 sekunde i da prati ponašanje sudije i navijača, sve dok korisnik ne pritisne dugme 'Stop'. Ponašanje sudije i navijača treba realizovati uz korišćenje mehanizma Observer.

(55)

2. Napisati Prolog predikat koji za datu listu izdvaja neoznačene brojeve iz te liste, ofomljuje i na izlazu prikazuje pregledno ispisano uređeno drvo u kom svaki čvor (čvor sadrži pročitani neoznačeni broj i broj njegovih pojava) ima maksimalno tri potomka – potomak x čvora y treba da se nađe u levom poddrvetu ako je $0 < x \leq y/3$, u centralnom poddrvetu ako je $y/3 < x < 3y$ i u desnom poddrvetu za $x > 3y$. Izvršenjem predikata treba da se, po prikazu drveta, na izlazu prikažu i svi neoznačeni brojevi iz liste i to sortirani u rastući poredak ako je argument predikata koji sledi i iza liste unififikabilan sa atomom rast, a sortirani u opadajući poredak ako se taj argument može unificirati sa atomom opad. Pri rešavanju zadatka se za smeštaj neoznačenih brojeva iz liste isključivo treba koristiti drvoidna struktura - ne treba koristiti nizove niti kolekcije za tu svrhu.

(25)

3. Napisati Scheme funkciju koja za datu listu L i dat ceo broj N , kao rezultat vraće listu iz koje su uklonjeni svi elementi originalne liste L na parnim pozicijama koji su prosti brojevi.

(20)