

Praktikum iz objektno orijentisanog programiranja (13S112POOP)

Projektni zadatak – C++

Napisati skup klasa sa odgovarajućim metodama, konstruktorima, operatorima i destruktorkama za realizaciju softverskog sistema za manipulaciju podataka Olimpijskih igara. Potrebno je obezbediti izdvajanje (parsiranje) ulaznih podataka, kao i osnovnu manipulaciju nad događajima na Olimpijskim igrama.

Korisnik (naručilac) softvera, želi da softver pruži sledeće funkcionalnosti:

- Interakciju sa korisnikom putem tekstualnog menija (konzole) ili grafičkog korisničkog interfejsa
- Učitavanje podataka o događajima sa Olimpijskih igara
 - o učitavanje podataka o sportistima
 - o učitavanje evidencije o učešću na Olimpijskim igrama
- Parsiranje prikupljenih podataka
- Osnovnu manipulaciju nad podacima
 - o prikaz podataka o sportistima, igrama i događajima
 - o filtriranje podataka
- Kraj rada

Za uspešno rešenje zadatka potrebno je izvršiti analizu zahteva. Kao rezultat analize, potrebno je dopuniti i precizirati funkcionalnu specifikaciju softverskog alata. Na osnovu specifikacije, potrebno je napisati sistem klasa u jeziku C++ koje realizuju traženi softver. U nastavku su navedeni neki elementi specifikacije. Od studenata se očekuje da dopune one stavke koje nisu dovoljno precizno formulisane, odnosno dodaju nove stavke (tamo gde to ima smisla) ukoliko uoče prostor za unapređenje. Izmene i dopune specifikacije mogu da donekle odudaraju od zahteva naručioca softvera u onoj meri u kojoj to neće narušiti traženu funkcionalnost. Takođe, priloženi UML dijagram koji opisuje zahtevani softver se ne mora obavezno poštovati, već samo predstavlja skicu potencijalnog rešenja. Prilikom izrade specifikacije voditi računa o potencijalnom unapređenju softvera na osnovu naknadnih zahteva.

Prilikom izrade rešenja, od studenata se očekuje intenzivno korišćenje svih onih mogućnosti koje pružaju specifikacija jezika C++ i biblioteka STL, kao što su šablonske funkcije, kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi i sl. **Rešenja koja ne vode računa o ovom aspektu neće moći da dobiju maksimalan broj poena.** Takođe, voditi računa o **objektno orijentisanom dizajnu rešenja**, čistoći, čitkosti i komentarisanoj programskoj kodi.

Funkcionalna specifikacija

U nastavku je zadat deo korisničkih zahteva koje treba razraditi i, po potrebi, dopuniti tako da se dobije funkcionalna aplikacija.

Interakcija sa korisnikom

Korisnik može da interaguje sa programom bilo izborom odgovarajućih opcija iz tekstualnog menija putem tastature ili izborom u datom trenutku dostupnih opcija putem grafičkog korisničkog interfejsa. Nije potrebno realizovati oba načina. Interakcija u slučaju grafičkog interfejsa može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatu opciju ili ispisuje poruku greške. Poruka greške treba da bude što je moguće detaljnija da bi korisniku pomogla da grešku otkloni. Sve eventualne parametre koji su potrebni prilikom rada aplikacije je potrebno zatražiti od korisnika. Ukoliko korisnik ne zada ništa, koristiti vrednosti fiksirane u programu.

Učitavanje podataka

Potrebno je obezbediti učitavanje podataka o događajima i sportistima iz tekstualnih fajlova i to u grupnom i pojedinačnom režimu rada. Grupni režim rada podrazumeva čitanje spiska svih postojećih evidencija o učešću, dok pojedinačni režim rada podrazumeva učitavanje podataka samo o učešćima za zadatu godinu održavanja Olimpijskih igara. Nezavisno od režima učitavanja, nakon učitavanja evidencije o učešću potrebno je učitati i sve sportiste koji su učestvovali na prethodno učitanim događajima. Evidencija učešća se učitava iz zadatog fajla u kome su u svakom redu zapisane informacije o učešću jednog takmičara (pojedinaac ili tim) na jednom događaju. Svaki red sadrži godinu održavanja i tip Olimpijskih igara, ime grada u kome su održane Olimpijske igre, ime sporta, ime discipline, tip događaja (individualan ili timski), državu za koju nastupa takmičar, jedinstvenu oznaku sportiste (ili svih sportista u timu razdvojenih separatorom zarez i navedenih u paru uglastih zagrada) i osvojenu medalju (ukoliko je takmičar osvojio neku medalju). Sve informacije su odvojene jednim znakom „!“:

2016 Summer!Rio de Janeiro!Basketball!Basketball Women's Basketball!Team!Serbia!['16977', '19205', '24064', '24765', '24767', '56046', '64194', '80039', '90547', '94165', '98178', '114347']!Bronze

2016 Summer!Rio de Janeiro!Boxing!Boxing Men's Flyweight!Individual!Cuba!125690!

2016 Summer!Rio de Janeiro!Boxing!Boxing Men's Flyweight!Individual!Uzbekistan!135205!Gold

U svakom redu datoteke koja sadrži informacije o sportistima (čije ime se zadaje prilikom učitavanja) nalazi se jedinstvena oznaka sportiste, ime sportiste, pol (M ili F) sportiste, broj godina, visina i težina. Svi podaci su dati za godinu kada je sportista prvi put učestvovao na Olimpijskim igrama. Sve informacije su odvojene znakom „!“:

13029!Usain St. Leo Bolt!M!17!196!95

Prilikom učitavanja podataka o mreži koristiti regularne izraze za parsiranje datih fajlova. Predvideti način za oporavak od grešaka u slučaju neispravnog formata datoteke ili nepostojanja tražene datoteke.

Osnovna manipulacija nad podacima

Korisniku treba omogućiti ispis sledećih osnovnih metrika:

- broj učesnika na svim događajima na Olimpijskim igrama
- ukupan broj disciplina na Olimpijskim igrama
- prosečnu visinu svih sportista (ukoliko postoji podatak o visini sportiste)
- prosečnu težinu svih sportista (ukoliko postoji podatak o težini sportiste)

Korisniku je potrebno dozvoliti i da filtrira podatke prilikom ispisa osnovnih metrika, po jednom ili više sledećih kriterijuma:

- filtriranje na osnovu zadatog sporta
- filtriranje na osnovu države za koju nastupa takmičar
- filtriranje na osnovu godine održavanja Olimpijskih igara
- filtriranje na osnovu tipa događaja (individualan ili timski)
- filtriranje na osnovu osvojene medalje (zlatna, srebrna, bronzana ili bez medalje)

Pored osnovnih manipulacija potrebno je obezbediti i sledeće:

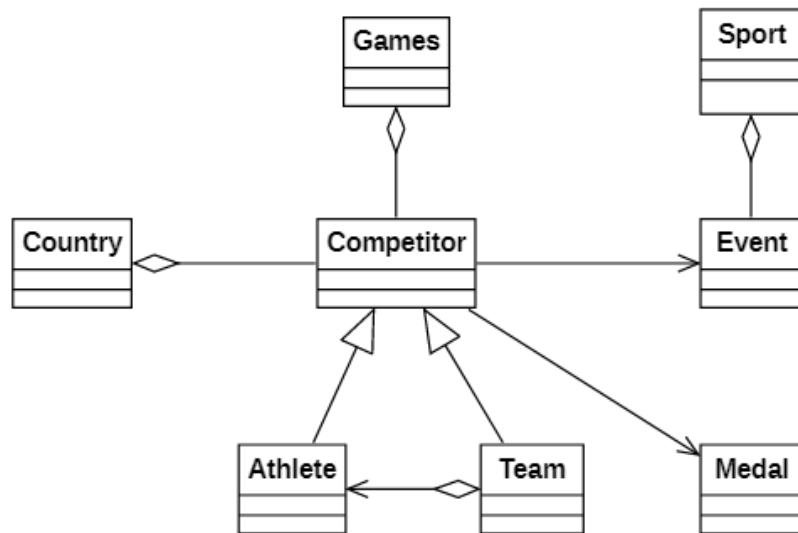
- Određivanje broja različitih sportova u kojima je zadata država osvojila barem jednu medalju
- Odrađivanje tri najbolje države po uspehu na zadatim Olimpijskim igrama (igre se zadaju godinom održavanja i godišnjim dobom kada su održane). Uspeh je bolji ukoliko je država osvojila više zlatnih medalja. Ukoliko dve države imaju isti broj zlatnih medalja bolji uspeh ima ona koja ima više srebrnih medalja, a ukoliko dve države imaju i isto broj srebrnih medalja bolji uspeh ima ona koja ima veći broj bronzanih medalja
- Određivanje država koje su na barem jednim Olimpijskim igrama ostvarile najbolji uspeh
- Određivanje deset najmlađih učesnika Olimpijskih igara koji su osvojili medalju na svom prvom učešću
- Dohvatanje svih parova država – sportista, za sve sportiste koji su osvojili barem jednu medalju u pojedinačnoj i barem jednu medalju u grupnoj konkurenciji (ne nužno iste godine)
- Određivanje svih sportista koji su učestvovali na zatom paru Olimpijskih igara
- Dohvatanje svih timova koje je zadata država imala na zadatim igrama, sortiranih prvo po broju sportista u timu opadajuće, a zatim leksikografski po imenu discipline u kojoj se tim takmiči rastuće
- Određivanje svih gradova u kojima su Olimpijske igre održane barem jednom

Kraj rada

Korisnik može da zahteva kraj rada. Od korisnika se traži potvrda za napuštanje programa.

Dijagram klasa

Na osnovu prethodne funkcionalne specifikacije formiran je sledeći dijagram klasa. Dijagram klasa nije detaljan, te ga treba tumačiti kao skicu koja načelno ukazuje na arhitekturu softvera. Studenti mogu da koriste ovaj dijagram kao referencu i, po potrebi, prošire ga da bi ga usaglasili sa eventualnim dopunama specifikacije.



Prilikom implementacije rešenja, obratiti pažnju na objektno orijentisani dizajn i intenzivno koristiti kolekcije i algoritme standardne biblioteke jezika C++ i lambda funkcije gde god je to moguće.