# Projektni zadatak – izvješće

**Ime i prezime studenta: Domin Radić**

**Godina: 2024.**

**Grupa: DE-DKB-KV1**

|  |  |
| --- | --- |
| **Datum:** | **Kratki opis:** |
| **11.01.2024.** | **Kreirane sve potrebne datoteke za izradu projekta, kreirano zaglavlje projekta, i dvije funkcije datoteke izvornog koda.** |
| **18.01.2024** | **Napisane sve funkcije u datoteci izvornog koda, napisane sve funkcije u datoteci zaglavlja, pokušaj testiranja funkcija za dodavanje elemenata u listu i ispis.** |
| **25.01.2024.** | **Testirane sve funkcije u datoteke izvornog koda, izrada izbornika aplikacije u glavnoj datoteci.** |
|  |  |

**OPIS ZADATKA:**

Napraviti modul koji predstavlja kružnu dvostruko povezanu listu. Modul treba sadržavati funkcije za inicijalizaciju liste (dinamički alocirati memoriju za listu), brisanje liste, dodavanje elemenata u listu, brisanje svih elemenata liste, ispis svih elemenata liste te funkcije za provjeru da li je lista prazna (isEmpty). Lista treba sadržavati pokazivač na prednji (head) i zadnji (tail) element. Modul izvesti kao statičku biblioteku.

Koristeći napravljeni modul za manipulaciju dvostruko povezanom listom, potrebno je napraviti aplikaciju koja omogućuje vođenje statistike nogometne ekipe. Podaci koji opisuju svakog igrača su ime, prezime, godine, pozicija koju igra , broj asistencija te broj golova. Samome korisniku potrebno je omogućiti pregled igrača po poziciji, golovima i asistencijama.

**OPIS IMPLEMENTIRANIH FUNKCIJA**

**PlayerListInitialize –** funkcija za kreiranje nove liste igrača, alocira memoriju za novu listu, postavlja head i tail liste na NULL (što označava da je lista prazna na početku) te vraća pokazivač na novu listu.  
  
**PlayerListInsert –** funkcija za dodavanje novog igrača u listu. Funkcija prima listu igrača i igrača koji se dodaje u listu. Alocira se memorija za igrača koji se dodaje u listu, kopiraju se podaci iz elementa koji je predan u novog igrača. Postavlja pokazivače next (koji pokazuje na sljedećeg igrača u listi) i prev (koji će pokazivati na prethodnog igrača) tako da odgovaraju poziciji u listi. Ako lista nema igrača, head i tail se postavljaju na novog igrača što znači da je on prvi element u listi. Ako nije prvi, dodaje se na kraj liste.

**PlayerListDestroy** - funkcija za brisanje cijele liste, prima listu igrača. Oslobađa svu zauzetu memoriju za igrače i samu listu. Prolazi kroz listu i oslobađa memoriju za svakog igrača. Na kraju oslobodi memoriju za cijelu listu

**PlayerListDelete –** funkcija za brisanje svih igrača iz liste ali koja zadržava strukturu liste, prima listu igrača. Oslobađa memoriju za svakog igrača. Postavlja head i tail liste na NULL što znači da je sada lista prazna no još je u memoriji i može se popunjavati novim igračima.

**PlayerListPrint –** funkcija za ispisivanje svih trenutnih igrača u listi. Prolazi kroz cijelu listu i ispisuje ime, prezime, poziciju, broj golova i broj asistencija svakog igrača.

**IsEmpty –** funkcija koja se koristi za provjeru da li je lista prazna, koristi se u većini funkcija na početku da se ispita da li je postoje elementi unutar strukture. Ako je head NULL, odnosno ako je prvi element u listi nula, lista je prazna. Vraća 1 ako je lista prazna, inače je 0.

**SearchByPosition –** funkcija za pretraživanje igrača na određenoj poziciji, prima listu igrača i poziciju za pretraživanje. Prolazi kroz listu i uspoređuje poziciju svakog igrača sa zadanom pozicijom. Ako su pozicije iste, ispisuje informacije o igračima koji odgovaraju toj poziciji.

**SortByGoals –** funkcija za sortiranje igrača po broju golova u silaznom redoslijedu, prima listu igrača. Koristi algoritam selekcija za sortiranje. Prolazi se kroz cijelu listu i za svakog igrača pronalazi igrača s najvećim brojem golova i mijenja im mjesta. Za svakog se igrača u listi provjerava ostatak liste u potrazi za igračem s većim brojem golova. Ako se pronađe takav igrač, zamjenjuju se pozicije tih igrača. Ako je potrebno zamijeniti igrače, ažuriraju se pokazivači kako bi pravilno napravili promjene, te pomiče se trenutni igrač na sljedećeg kako bi se nastavilo sortiranje.

**SortByAssists** – funkcija za sortiranje igrača po broju asistencija u silaznom redoslijedu. Ima identičnu strukturu kao i funkcija za sortiranje po broju golova, s tim da se igrači uspoređuju po broju asistencija.

Sve funkcije koje su tražene su implementirane i testirane. Napravljena je aplikacija za vođenje nogometne statistike u obliku izbornika gdje se biraju sve funkcije zadatka. Potrebno je unijeti najmanje dva igrača u listu da bi izbornik radio. Modul je izveden kao statička biblioteka.  
  
Prilikom testiranja primijetio sam samo da kada se pozove prvo funkcija za sortiranje igrača po golovima, pa nakon toga za sortiranje po asistencijama program se prekine iz nekog razloga. Ostale kombinacije rade normalno. Npr. nakon što se pozove prvo za asistencije pa golove nakon toga sve druge kombinacije rade normalno.