Zombori Péter (DSYU1P)

Nagyházi specifikáció

Feladatleírás

A feladat a Sportegyesület feladat. A program egy sportegyesület csapatait tartja nyilván. Három féle csapat lehet, Kézilabda csapat, Kosárlabda csapat és Focilabda csapat. Minden csapatnak vannak közös tulajdonságaik, például a nevük és a csapatok létszáma. Minden csapatnak vannak sajátos tulajdonságaik. A kézilabda csapatnak jegyzi az éves támogatásait, a kosárlabda csapat jegyzi a pompomlányok számát, a focicsapat jegyzi az edzőik számát.

Adatstruktúra

A feladat adatstruktúrája több osztályra van bontva. A csapat osztály a szülőkönyvtár, ahol a közös tulajdonságok jelennek meg. Illetve van a három örökös osztály a saját tulajdonságaikkal. Az adatok a csapat típusa *txt* fileban vannak tárolva. Ez 3db file: *kezi.txt*, *kosar.txt*, *foci.txt*. A fileokból beolvasott adatokat három láncolt listában tárolódnak. Azért jobb ehhez a feladattípushoz a láncolt lista, mint a tömb, mert a sok csapat esetén egy csapat törlése nagyon sok erőforrást venne igénybe (csinálni egy egyel rövidebb tömböt és bemásolni mindent), míg a láncolt listánál csak összekötöm az előtte és következőre mutatót. Ugyan ez igaz új csapat felvételére.

Fontos, hogy a program esetleges bővítésénél jól felhasználható legyen, ezért van a Nyilvántartás osztály, amely egy API-ként szolgál, egybefogja az adattípusok funkcióit, beszédes elnevezésű funkciókkal rendelkezik. Ez által a *gtest_lite* segítségével könnyedén lehet az adatokat tesztelni.

Fontos továbbá, hogy a felhasználó számára is jól nézzen ki a program. Mivel a program parancssoros program, ezért a körülményeknek megfelelően átgondolt designra van szükség. A felhasználó egy menürendszer segítségével fog tudni navigálni a funkciók között. Minden almenü egy tabulátorral beljebbről kezdődik, hogy átlátható legyen a menürendszerben való navigálás. Ezt egy Menü osztály valósítja meg, amelyen, ha meghívjuk a főMenü függvényt, 'átveszi az uralmat' a program felett, és a menürendszer lekezelését fogja megvalósítani. A menürendszer lekezeli a fileok beolvasását és az adatok fileokba lementését, és a felhasználóval való kommunikációt. Beolvasni elég a program indulásakor, menteni pedig automatikusan adatmódosuláskor, vagy programból való kilépéskor kell.

A csapatok nevei, láncolt lista hossza nem tudottak, ezeket a program dinamikusan kezeli. Általában nincs megkötés semmilyen input hosszában, kivéve a konzol sztenderd inputról való olvasásnál, ami maximum 256 karakter lehet. Ez nem befolyásolja az átlag felhasználót, mert nem életszerű egy 257 karakter hosszú csapatnév. Az adatok dinamikus kezelésénél fontos a memóriaszivárgás elkerülése, amiben a *memtrace* segít.

Dokumentáció

A dokumentációt automatikusan a Doxygen a GraphViz és a MiKTeX generálja a *header* fileokban a függvények, osztályok, struktúrák felett elhelyezett Doxygen kommentekből. Egy sok oldalas *docs.pdf* PDF dokumentum és egy interaktív weboldal a dokumentáció. A weboldal, a *html* mappában az *index.html* fileból indul. A két dokumentáció között óriási különbség nincs. A PDF verzió részletesebb egy fokkal, a HTML verzió interaktívabb. A preferált fő dokumentáció a PDF verzió, a több részlet a hossz miatt és a kompaktsága miatt.

Menü

A menürendszer fő feleadata a felhasználó és az adatok közti kommunikáció. Képes a következő feladatokra:

A főmenüben:

- Az összes adat kilistázására, csapattípusok almenüjébe navigálni

Az almenükben:

- Az adott csapattípusú csapatokat kilistázni. Csapatnév szerint csapatra keresni, és a keresett csapat almenüjére navigálni.
- Új csapatot beszúrni

A keresett csapat almenüjében:

- A csapat adatait (szintén almenüben módosítani)
- A csapatot törölni

A menük, almenük között számokkal (négynél nem nagyobb számokkal) lehet navigálni. A hibás számot a program kezeli és vár a helyes válaszra.

DevTools, Platformok

A mivel a program nem platformfüggetlen, ezért szükséges gondolni a különböző operációs rendszerek viselkedésére. Linux és UNIX típusú operációs rendszerek esetén a fileok között található *makefile* segít a program lefordításában, és egy *nagyhazi* futtatandó fileba fordul le a *make* parancs után (erről dokumentáció a *makefile* fileban). Windows alatt a *Makefile_WIN.cmd* file futtatása fog segíteni a hasonló nagyhazi.exe előállításához. A szkriptek által preferált fordító a g++. Avval rendelkeznie kell a számítógépnek.

Fontos figyelembe venni a Linux, UNIX, valamint a Windows közti sztenderd input viselkedést különbséget is. Ezt a program forduláskor automatikusan kezeli a preprocesszor, az automatikusan definiálódó operációsrendszer változókkal (*__linux__*, *__APPLE__* stb…)

Nem árt, ha a fordítók közötti külömbséget is figyelembe vesszük, ezért mindegyik fordítási scriptben a fordítón a *-std=c++11* kapcsoló be van állítva, így C++ 11 es verzió alatt fordul le minden program, minden rendszeren.

Tesztelés

A program Windows alatt és Debian 11 alatt van tesztelve. Külön *gtest_lite*-os tesztprogram nincs hozzá, de nagyon egyszerű írni a Nyilvántartás API-nak közönhetően.

Github

A feladat nyomon követhető GitHub-on, a félév során tanult módszereket használva.