

Nagy Bozsó Zsombor
Programozás II
Nagy házi feladat dokumentáció
2022. 05. 15.

Budapesti Műszaki és Gazdaságtudományi Egyetem

Sportegyesület:

Feladat kiírás: A Fitt Sportegyesület nyilvántartást szeretne vezetni a csapatairól. Minden csapat rendelkezik egy névvel és egy alaplétszámmal. A sportegyesület háromféle sportággal foglalkozik: labdarúgás, kosárlabda és kézilabda. A labdarúgó csapatnak két edzője van; a kosárlabda csapatnak elengedhetetlen kellékei a pom-pom lányok, aminek létszámát is nyilvántartják; a kézilabda csapatok pedig évente kapnak valamekkora összegű támogatást. A nyilvántartás rendelkezzen minimum az alábbi funkciókkal: új csapat felvétele, csapat törlése, listázás.

Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz ne használjon STL tárolót!

Specifikáció és felhasználói dokumentáció:

Menü vezérelt program.

Menüpontok:

1. Csapatok listázása
 - a. Akar-e csapatot törölni
 - i. Ha igen melyiket
2. Csapat felvétele
3. Kilépés

Csapatok listázása:

A program kilistázza az összes eltárolt csapatot. Adatonként egymás alá sportággal ellátva.

Csapat törlése:

Amennyiben a kiválasztott csapatra a felhasználó a továbbiakban nem tart igényt, úgy azt a csapatot törölheti az adatbázisból. Ekkor a csapat minden adata véglegesen elveszik. Ehhez a csapat törlése kérdésre 1-es bevitelt kell adni, majd a csapat sorszámával kiválasztani a törlendő csapatot. Kívül indexelés esetén a program erre figyelmeztet.

Csapat felvétele:

A felhasználó kiválasztja, hogy milyen sportágba szeretne csapatot felvenni. Ha nem létező sportágot ad meg, úgy a program hibajelzést ad és újra kell próbálni a bevitelt. Ha létező sportágot választott ki, akkor a továbbiakban megadja a csapat adatait, amelyek így bekerülnek a rendszerbe. Az adatok megadása egyesével történik mindig a program által megadott adatot kell megadni.

Kilépés:

A program ezen menüpontja csak arra szolgál, hogy a program biztonságosan bezárható legyen és elkerüljük az esetleges adatvesztést.

Tárolás:

A csapatok adataikkal együtt egy az adatok nevével ellátott .txt dokumentumban tárolódnak el. A mentés közvetlenül a feladat végrehajtása után történik, nincs szükség manuális mentésre. A fájlban a csapatok adatai egymás mellett egy sorban vesszővel elválasztva kerülnek mentésre.

Objektum orientáltság:

A Csapatok őssztály leszármazottjai lesznek a labdarúgás, kosárlabda, kézilabda. A sportágak száma dinamikusan foglalt így új beolvasásnál vagy törlésnél folyamatosan bővílhet, vagy csökkenhet az adatszerkezet korlátozások nélkül. Az alapvető tagfüggvényeken kívül megvalósítom az indexelést, ahol túlindexelés esetén hibajelzés érkezik a felhasználónak.

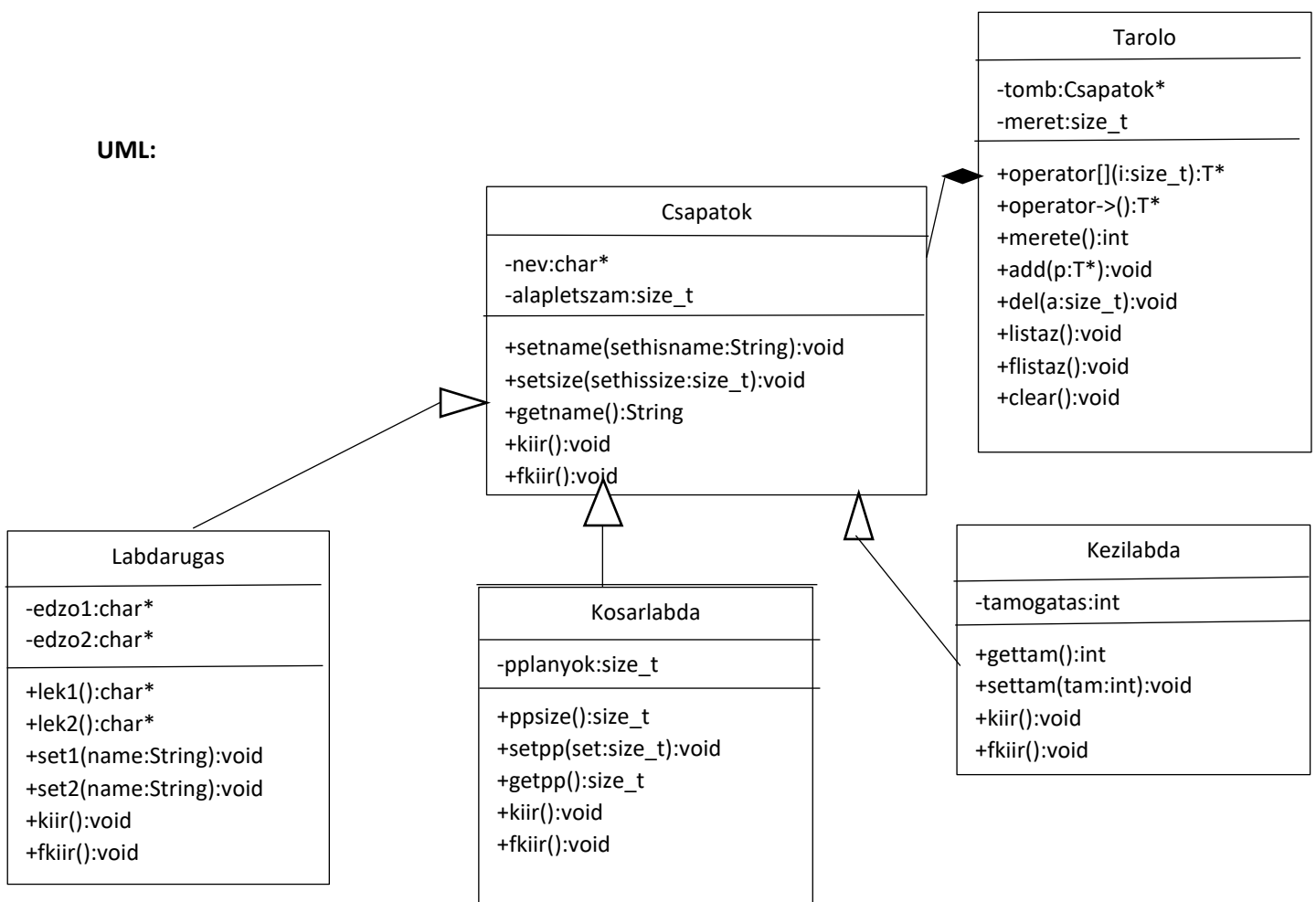
Tároló:

A csapatok egységes tárolóban tárolódnak, ehhez lehet hozzáadni és kitörölni csapatokat. A program ennek a tárolónak az adatait fileból olvassa be és oda is írja vissza.

Tesztelés:

A tesztelés során csapat kerül törlésre, és létrehozásra, valamint ahol hibalehetőség van ott ezen hibák kiírásra kerülnek.

UML:



Programozói dokumentáció:

Tarolo:

Egy heterogén kollekció. Template T* pointereket tárol egy T** tömbben, valamint ennek a tömbnek a hosszát: size_t meret.

Függvényei:

size_t merete()

//a tömb aktuális méretét adja vissza

void del(size_t a)

//a paraméterként átadott sorszámú elemét törli a tömbnek. Az elemeket átpakolja egy ideiglenes tömbbe, eggyel kisebb méretűre foglaljuk, a törölt elemet felszabadítjuk a többi pedig vissza pakoljuk. Ha a tömb határain kívül lévő elemet akarnánk törölni, a függvény kivételt dob.

void clear()

//Ezt tartalmazza a destruktork is. Felszabadítja a tárolóban eltárolt pointereket egyesével, majd felszabadítja a tároló összes pointerét is.

T* operator[](size_t i)

//a tömb valahányadik elemét adja vissza. Ha túl indexeljük kivételt dob.

T* operator->()

//Nyíl operátor, hogy a mutatóval ellátott függvényhívás megvalósítható legyen a tömb egy adott elemére.

void listaz()

//Végig járja a tömböt és minden elemére meghívja a kiir függvényt.

void flistaz(std::ofstream& filenev)

//Végig járja a tömböt és minden elemére meghívja az fkiir függvényt.

Csapatok:

Az űsosztálya a különböző sportcsapatoknak. Ez tárolja az olyan minden csapatnak levő információit, mint a név és a létszám.

Függvényei:

void Csapatok::setname(String setthisname)

//Paraméterként kapott Stringet adja a csapat nevének.

void Csapatok::setsize(size_t setthissize)

//Paraméterként kapott size_t értéket adja a csapat méretének.

String Csapatok::getname() const

//Lekérdezi a csapat nevét.

size_t Csapatok::getsize() const

//Lekérdezi a csapat méretét.

virtual void Csapatok::kiir()

//Virtuális tagfüggvény, minden sportnak más.

virtual void Csapatok::fkiir(std::ofstream& filenev)

//Virtuális tagfüggvény, minden sportnak más.

Labdarugas:

Futball csapatokat tárol. Örökli a nevet és a méretet a Csapatoktól. Saját privát adattagjai az edző1 és 2.

Függvényei:

```
String Labdarugas::lek1/2() const
```

```
//1. és 2. edző nevének lekérdezése.
```

```
void Labdarugas::set1/2(String name)
```

```
//1. és második edző nevének beállítása.
```

```
void Labdarugas::kiir()
```

```
//Kíírja a csapat összes adatát az os-re.
```

```
void Labdarugas::fkiir(std::ofstream& filenev)
```

```
//Kíírja a csapat összes adatát az ofs-re, amit meg kap paraméterként.
```

Kosarlabda:

Kosárlabda csapatokat tárol, melyek öröklík a nevüket és a méretüket a Csapatoktól. Saját privát adattagja a pom-pom lányok száma.

függvényei:

```
void Kosarlabda::setpp(size_t set)
```

```
//Paraméterként kapott size_t értéket állítja be a pom-pom lányok létszámának.
```

```
size_t Kosarlabda::getpp() const
```

```
//Megadja a csapathoz tartozó pom-pom lányok számát.
```

```
void Kosarlabda::kiir()
```

```
//Kíírja a csapat összes adatát az os-re.
```

```
void Kosarlabda::fkiir(std::ofstream& filenev)
```

```
//Kíírja a csapat összes adatát az ofs-re, amit meg kap paraméterként.
```

Kezilabda:

Kezilabda csapatokat tárol. Nevét és méretét öröklík a csapatoktól. Saját privát adattagja a támogatás összege.

Függvényei:

```
void Kezilabda::settam(int tam)
```

```
//A paraméterként átadott intet állítja be támogatásként.
```

```
int Kezilabda::gettam() const
```

```
//Megadja az adot csapathoz tartozó támogatást.
```

```
void Kezilabda::kiir()
```

```
//Kíírja a csapat összes adatát az os-re.
```

```
void Kezilabda::fkiir(std::ofstream& filenev)
```

```
//Kíírja a csapat összes adatát az ofs-re, amit meg kap paraméterként.
```

String osztály nem kötelező laborra megírandó függvényei:

```
std::istream& getlineString(std::istream& is, String& str)
```

//Egy sort olvas be is-ről. Operator >> helyett használandó és jobban is működik.

```
bool operator==(const String& c1, const String& c2)
```

//Két stringet hasonlít össze. Igazat ad ha egyformák és hamist, ha nem.

```
int String::konverttoint()
```

//Stringet konvertál át intre. Csak számokat tartalmazó intre hívjuk meg.

```
size_t String::konverttosize()
```

//Stringet konvertál át size_t-re. Csak számokat tartalmazó stringre hívjuk meg.

```
void fgetlineString(std::ifstream& is, String& str, char delim)
```

//Fileból olvas ki stringet adott elválasztó karakterig és sorvége jelig.

Teszt program

A teszt program két részre bomlik. Az első, ami a jportán fut le. Ez tartalmaz minden alapvető függvényt és tesztet. A felhasználói felület kicsit komplexebb és tartalmaz segéd függvényeket is. Felhasználói felület:

Létrehozunk egy Csapatok típusú tárolót.

Megnyitjuk az adatfileunkat.

Ameddig ennek vége nincs olvassuk ki az adatokat soronként.

Az adatokat , -nél felvágjuk és attól függően, hogy mi az első szó meghívjuk rá a megfelelő konstruktort.

Beletesszük az adott csapatot a tárolónkba.

Menüpont választó:

1.: A listáz függvény segítségével kilistázzuk a beolvasott csapatok adatait.

Ha a törlésre 1-es érkezik megadjuk az adott csapat sorszámát és azt a del függvény eltávolítja a tárolóból.

2.: Kiválasztjuk, hogy milyen csapatot akarunk hozzáadni a tárolónkhoz. Attól függően meghívódik a megfelelő sportág konstruktora és az ahhoz tartozó felvételi eljárás.

Az újonnan felvett csapatot vissza írjuk az adatfileunkba.