

## Sportegyesület

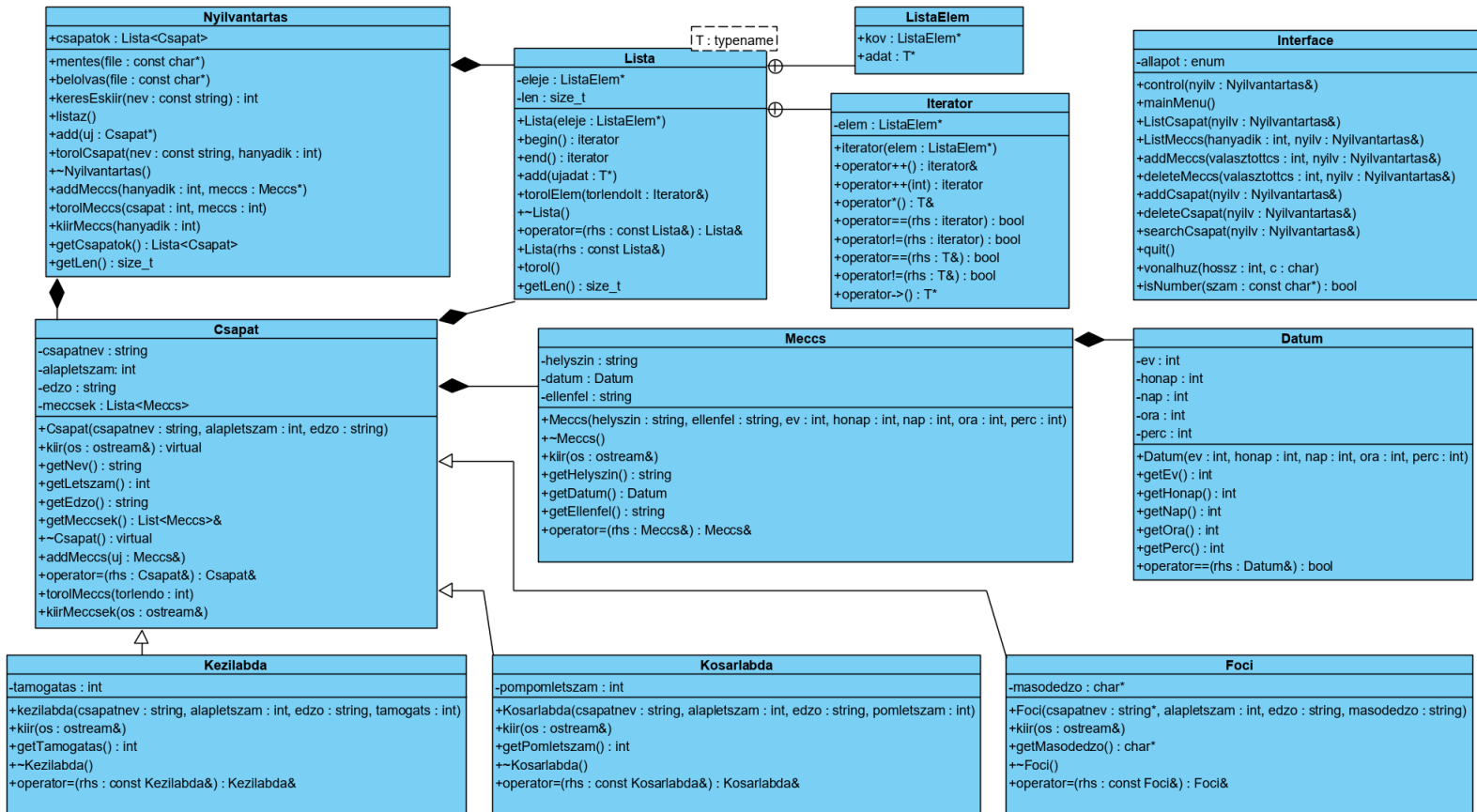
### Felhasználói Dokumentáció

A program egy nyilvántartást valósít meg egy egyesület csapatairól. Minden csapatnak van egy neve, alaplétszáma és egy edzője, illetve vannak meccsei is, amikről tudni az ellenfél csapat nevét, a meccs időpontját és a helyszínét. A sportegyesület 3 féle sportággal foglalkozik: labdarúgás, kosárlabda és kézilabda. A focicsapatnak két edzője van; a kosárlabdacsapatnak van egy pom-pom csapata, aminek nyilvántartják a létszámát; a kézilabda csapat meg évente kap támogatást, aminek az összege el van tárolva.

A programot indítva egy menü nyílik meg ahonnan lehetőség van a csapatok listázására, új csapat hozzáadására, csapat törlésére, csapatra keresésre és a programból való kilépésre.

- Ha listázzuk a csapatokat, akkor soronként kiírja a csapatok nevét, fajtáját (foci, kosár, kézi), alaplétszámát, edzőjének a nevét (foci csapatnál mindkét edző nevét), kosárcsapatnál a pom-pom lányok számát, kézilabdacsapatnál a támogatás összegét. Minden csapathoz tartozik egy sorszám is, aminek a beírásával lehet kilistázni a meccseinek az adatait soronként sorszámmal ellátva. Itt lehet törölni meccset vagy hozzáadni új meccset. Törlés menüpontnál a meccs sorszámát kell megadni. Meccs hozzáadása menüpontban a program bekéri az ellenfél nevét, a meccs időpontját és helyszínét.
- Új csapat hozzáadása menüpontban először ki kell választani, hogy milyen típusú csapatot kívánunk létrehozni (foci, kosár, kézi), ezek után a program bekéri a csapat nevét(nem lehet benne ',', alaplétszámát, edzőjének a nevét, foci esetén a másik edző nevét, kosárnál a pom-pom csapat létszámát, kézicsapatnál a támogatás összegét).
- Ha a csapat törlése menüpontot válasszuk, akkor a program bekéri a törölni kívánt csapat pontos nevét, kilistázza azokat a csapatokat, amelyeknek ez a neve, ha több csapatnál is talál egyezést, akkor ki kell választani, hogy melyiket kívánjuk törölni.
- A keresés menüpontban egy csapat nevére lehet keresni, ha több egyezés is van akkor az összes találat kilistázódik.
- A kilépés menüpontot választva a megtörténik a mentés és kilép a program.

## Programozói dokumentáció



A csapatokat és a csapatokhoz tartozó meccseket is láncolt listában tárolja a program. A program indításakor egy .txt fájlból beolvasásra kerülnek az elmentett adatok (felépül a nyilvántartás), bezárásakor pedig mentésre kerülnek ugyanabba a .txt fájlba. A program irányítása az Interface osztályon keresztül történik.

## Modulok

### 1. Interface.h

Ebben a modulban található az Interface osztály, ami az egész program irányításáért felel. Az interface osztálynak van egy enum típusú állapot változója, aminek a segítségével az egyes menüpontok közötti lépés megvalósul. Ezek a fgv-ek végzik a menüpontoknak megfelelő utasítások kiírását és a megfelelő adatok bekérését.

```
void Interface::vonalhuz(int hossz = 100, char c = '-') const
```

Ez a fgv csak az olvashatóbb megjelenítéshez szükséges, adott hosszúságú és adott karakterekből álló sorozatot ír ki.

**bool** Interface::isNumber(const char\* szam) const

Eldönti egy char tömbről, hogy egy számot tartalmaz-e. Ha minden karaktere szám, akkor true-val tér vissza, különben false-al;

**void** Interface::control(Nyilvantartas& nyilv)

A menüpontok közötti lépéseket biztosítja, és meghívja a megfelelő menüpontok függvényeit. Paraméterként megkapja a nyilvántartást, amivel dolgozni akarunk.

**void** Interface::mainMenu()

A főmenüben történő műveletekért felelős.

**int** Interface::listCsapat(Nyilvantartas& nyilv)

A csapatok listázása menüpontban törtétekért felelős, visszatérési értéke az a szám, amit annak megfelelően adunk be, hogy mit kívánunk tenni. (Ha 0: vissza a főmenübe, különben kiválasztottunk egy csapatot, aminek a meccseit listázni kívánjuk.

**void** Interface::listMeccs(int hanyadik, Nyilvantartas& nyilv)

Megkapja, hogy a listából hanyadik csapat meccseit kívánjuk listázni és ezt végrehajtja, illetve itt történik a következő lépés választása is.

**void** Interface::addMeccs(int valasztottcsapat, Nyilvantartas& nyilv)

Meccs hozzáadása esetén ez a fgc hívódik meg, paraméterként megkapja, hogy a listában hanyadik csapathoz akarunk meccset hozzáadni. Itt történik a meccs hozzáadása.

**void** Interface::deleteMeccs(int valasztottcsapat, Nyilvantartas& nyilv)

Meccs törlése esetén ez a fgv hívódik meg, paraméterként megkapja, hogy a listából hanyadik meccset kívánjuk törölni. A megfelelő meccset kitörli.

**void** Interface::addCsapat(Nyilvantartas& nyilv)

Bekéri a létrehozandó csapat adatait és hozzáadja a nyilvántartáshoz.

**void** Interface::deleteCsapat(Nyilvantartas& nyilv)

Bekéri a törlendő csapat pontos nevét és kitörli, ha több egyezést is talál, akkor ki kell választani, hogy melyiket kívánjuk törölni.

**void** Interface::searchCsapat(Nyilvantartas& nyilv)

Bekéri a keresett csapat pontos nevét és kilistázza az egyezéseket.

## 2. Lista.hpp

A generikus Lista osztályt tartalmazza. A megfelelő típusú adatok pointereit tárolja.

**void torolElem(Iterator& torlendoIt)**

Kap egy iteratort, ami arra az elemere mutat, amit a listából törölni kívánunk. A kapott elem törlését végzi.

**void torol()**

Felszabadítja az egész listát.

**void add(T\* ujadat)**

A paraméterként kapott adatot hozzáadja a listához. A az új adatot pointerként veszi át.

### 3. Nyilvantartas.h

Itt található a Nyilvantartas osztály, ami a csapatok listáját tárolja.

**void mentes(const char\* file = "nyilvantartas.txt")**

A paraméterként kapott fájlba írja a csapatok listáját.

**void beolvas(const char\* file = "nyilvantartas.txt")**

A paraméterként kapott fájlból felépíti a csapatok listáját.

**int keresEskiir(const string nev);**

A paraméterként kapott stringet a csapatok nevével összehasonlítva egyezéseket keres, ezeket kilistázza. Visszatérési értéke az egyezések száma.

**void listaz()**

Kilistázza az összes csapatot.

**Lista<Csapat> getCsapatok() const**

Visszatérési értéke a csapatok lista egy másolata.

**void addMeccs(int hanyadik, Meccs\*)**

A paraméterként kapott Meccs pointert hozzáadja a megfelelő csapathoz, amit a paraméterként átvett szám határoz meg. (A lista annyiadik eleme)

**void torolMeccs(int csapat, int meccs)**

Átvesszi egészként, hogy a lista hányadik csapatából kell meccset törölni és hogy a megfelelő csapat meccsi között hányadikat kell törölni.

**void add(Csapat\* uj)**

Hozzáadja a paraméterként átvett csapatot a csapatok listához.

```
void torolCsapat(const string nev, int hanyadik = 1)
```

Paraméterként átveszi a törölni kívánt csapat pontos nevét, átveszi , hogy ez azonos nevéek közül hányadikat akarjuk törölni, arra az esetre ha több csapat neve is azonos. Kitorli a listából a megfelelő csapatot.

```
void kiirMeccs(int hanyadik)
```

Paraméterként megkapja, hogy a listából hányadik csapat meccseit kívánjuk listázni, majd ezt végrehajtja.

```
size_t getLen() const
```

Visszatér a csapatok lista hosszával.

#### 4. Csapat.h

A Csapat osztály található itt.

```
virtual void kiir(std::ostream& os = std::cout)
```

Kiírja a csapatot a megfelelő formátumban a paraméterként átvett ostreamre.

```
std::string getNev() const
```

```
int getLetszam() const
```

```
std::string getEdzo()
```

```
Lista<Meccs> getMeccsek() const
```

Visszatérnek a Csapat osztály megfelelő privát adattagjával.

```
void addMeccs (Meccs* uj)
```

A paraméterként átvett Meccs pointert hozzáfüzi a meccsek listához.

```
void torolMeccs(int torlendo)
```

Paraméterként átveszi hogy hányadik elemet akarjuk törölni a meccsek listából és ennek megfelelően kitorli.

```
void kiirMeccsek(std::ostream& os = std::cout)
```

A paraméterként átvett ostreamre kiírja a meccsek lista minden elemét.

## 5. Meccs.h

A Meccs osztályt tartalmazza.

```
void kiir(std::ostream& os=std::cout)
```

Kiírja a meccset a megfelelő formátumban a paraméterként átvett ostreamre.

```
Std::string getHelyszin() const
```

```
Std::string getEllenfel() const
```

```
Datum getDatum() const
```

Visszatérnek a Meccs osztály megfelelő privát adattagjával.

## 6. Datum.h

A Datum osztályt tartalmazza.

```
int getEv() const
```

```
int getHonap() const
```

```
int getNap() const
```

```
int getOra() const
```

```
int getPerc() const
```

Visszatérnek a Datum osztály megfelelő privát adattagjával.

## 7. Foci.h

A Foci osztályt tartalmazza, ami származtatott osztálya a Csapat osztálynak.

```
void kiir(std::ostream& os = std::cout)
```

Kiírja a foci csapatot a megfelelő formátumban a paraméterként átvett ostreamre.

```
std::string getMasodedzo() const
```

Visszatér a másodedzőj nevével.

## 8. Kosarlabda.h

A Kosarlabda osztályt tartalmazza, ami származtatott osztálya a Csapat osztálynak.

```
void kiir(std::ostream& os = std::cout)
```

Kiírja a kosár csapatot a megfelelő formátumban a paraméterként átvett ostreamre.

```
int getPomletszam() const
```

Visszatér a pom-pom csapat létszámával.

## 9. Kezilabda.h

A Kezilabda osztályt tartalmazza, ami származtatott osztálya a Csapat osztálynak.

```
void kiir(std::ostream& os = std::cout)
```

Kiírja a kézilabda csapatot a megfelelő formátumban a paraméterként átvett ostreamre.

```
int getTamogatas() const
```

Visszatér a támogatás értékével.