

# **YY1HDA Nagyházi Dokumentáció**

## **Feladat:**

A Fitt Sportegyesület csapatairól való nyilvántartását készítem el. Az adatbázisban minden csapathoz tartozik egy név és egy alaplétszám. Az sportegyesület 3 sportágat képvisel: labdarúgás, kosárlabda és kézilabda.

A három csapathoz egyenként jár egy információ:

- Labdarúgó csapatok: két edző
- Kosárlabda csapatok: kellékei a pom-pom lányok (létszám)
- Kézilabda csapatok: évente kapnak támogatást

A csapatokon belül játékosok is lehetnek, akikről 3 információ érhető el:

- név
- érték (mezszám)
- pontszám

A nyilvántartásban az alábbi funkciók elérhetőek:

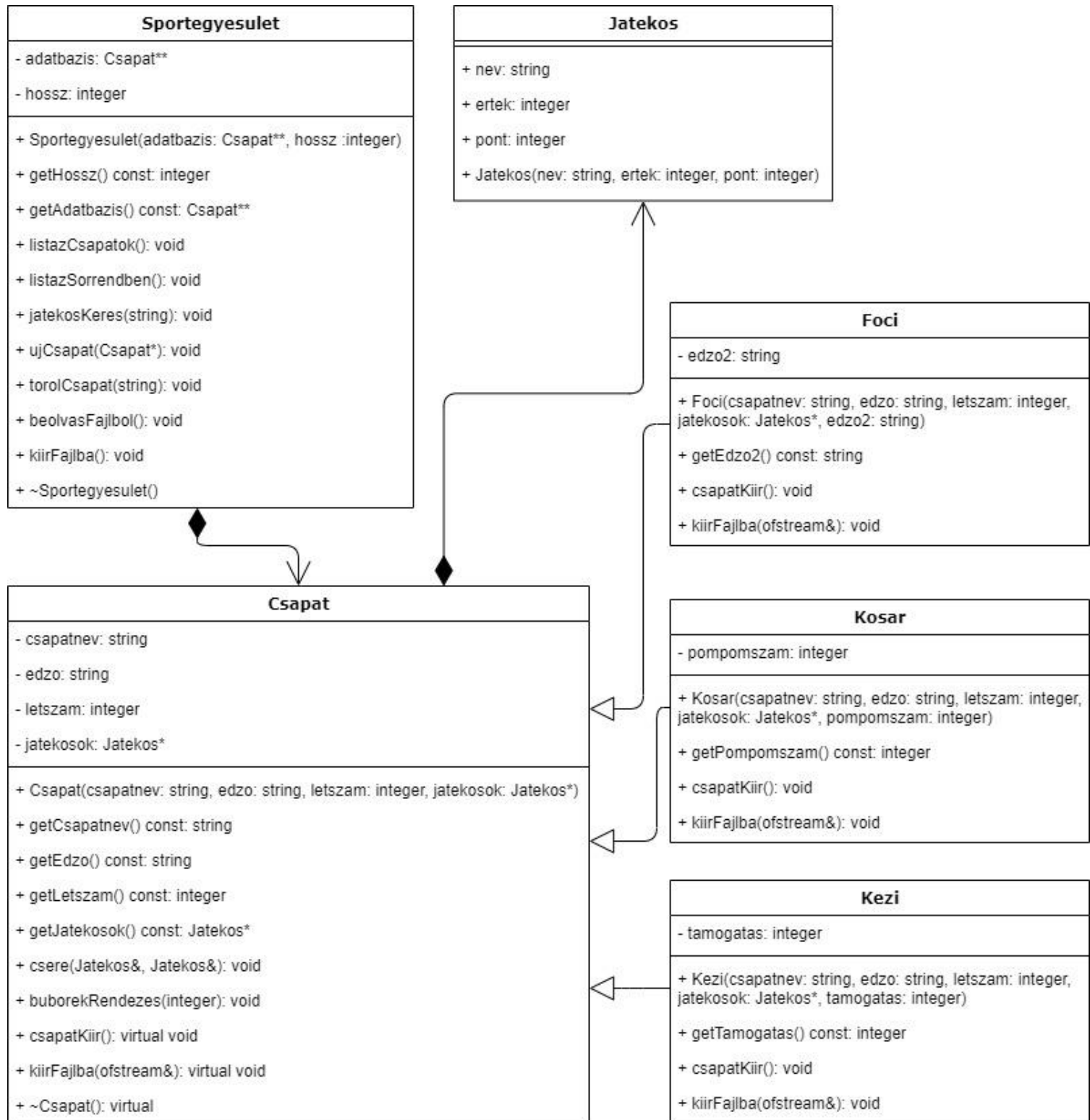
- új csapat felvétele
- csapat törlése
- keresés játékosok között
- összes csapat listázása
- összes csapat kilistázása játékos neve, mezszáma, vagy pontja alapján rendezve

## **Felhasználói Dokumentáció:**

A programom menü vezérelt, így mikor elindul egyből 6 lehetőség közül lehet választani.

- (1) Új csapat hozzáadása: Először a típust kell kiválasztani: (1) Foci (2) Kosár (3) Kézi. Ezután kérdésenként meg kell adni a csapatnevet, edző nevét, csapat létszámát és a plusz adatot. Ha ez megtörtént, és a csapatnak nagyobb 0-nál a létszáma, akkor rákérdez, hogy szeretne-e hozzáadni játékost. Igen (1) esetén kérdésenként kell beadni a játékos nevét, értékét és pontszámát. Nem (2) esetén visszakerülünk a menübe. Annyi játékost lehet begépelni, amekkora a létszám, ha megteklik a játékosok száma, akkor automatikusan visszakerülünk a menübe.
- (2) Csapat törlése: Beírja a csapatnevet, amely ezután törlődni fog az adatbázisból, valamint visszakerülünk a menübe.
- (3) Keresés a játékosok között: Beírja a játékos nevét, vagy a nevének egy részletét, akire rá szeretne keresni. A megtalált játékos csapatát kiírja a szabványos kimenetre a program, ezután visszakerülünk a menübe.
- (4) Csapatok kilistázása: Ha ezt választja, akkor kiíródik az összes csapat minden adatával együtt a szabványos kimenetre, majd visszakerülünk a menübe.
- (5) Listázás sorrendben: Ha ezt választja, akkor ki kell a felhasználónak választania, hogy melyik adat alapján legyenek a játékosok sorrendbe rendezve, (1) név (2) mezszám (3) pont, majd eszerint kiírja az összes csapatot a szabványos kimenetre, ezután visszakerülünk a menübe.
- (6) Kilépés

## Fejlesztői Dokumentáció:



## Függvények pontosítása:

- **Sportegyesület:**

- `Sportegyesület(Csapat**, integer):`
  - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként az adatbázisnak nullptr-t ad, míg a hosszt 0-ra állítja. Egyébként a megadott értékeket állítja be.
- `getHossz():`
  - A függvény paraméter nélküli és visszatér az adatbázis hosszával.
- `getAdatbazis():`
  - A függvény paraméter nélküli és visszatér a csapatok tömbjével.
- `listazCsapatok():`
  - Paraméter nélküli függvény, ami ha az adatbázis hossza üres, akkor kiírja, hogy az adatbázis üres, ha pedig nem, egy ciklus segítségével meghívja a `csapatKiir()` függvényt az adatbázis minden elemére.
- `listazSorrendben(integer):`
  - A függvény megkapja paraméterül azt, ami szerint listázni akar a felhasználó, ha az adatbázis hossza üres, akkor kiírja, hogy az adatbázis üres, ha pedig nem, akkor egy ciklus segítségével meghívja a `buborekRendezes()` függvényt, majd ezután a már sorrendbe rendezett adatbázis minden elemére meghívja a `csapatKiir()` függvényt.
- `jatekosKeres(string):`
  - A függvény a keresett játékos egész nevét, vagy a nevének egy részletét kapja meg paraméterül, majd a megkapott név és egy ciklus segítségével megvizsgálja, hogy a játékos szerepel-e az adatbázisban szereplő játékosok között. Ha nem találja, akkor kiírja, hogy nincs ilyen játékos, ha pedig megtalálja, akkor a játékoshoz tartozó csapatra meghívja a `csapatKiir()` függvényt. Ha több ugyanolyan nevű játékos is megtalálható az adatbázisban, akkor az összes csapatot kiírja, amelyben szerepel a név.
- `ujCsapat(Csapat*):`
  - Paraméterként megkap egy csapatra mutató pointert, foglal egyel nagyobb helyet a dinamikusan foglalt adatbázisnak egy ideiglenes adatbázisba, végig fut rajta egy ciklus segítségével, átmásolja az összes elemet, majd a végére beteszi a csapatot, amit megkapott a függvény. Ezután az adatbázist felszabadítja, majd átállítja az adatbázis pointerét az új ideiglenes adatbázisra. A függvény végén a hosszt megnöveli.
- `torolCsapat(string):`
  - Paraméterként megkapja a törlendő csapat nevét, egyel kisebb helyet foglal egy ideiglenes adatbázisba, és egy ciklus segítségével, ha nem egyezik a név a törlendő csapatével, akkor belemásolja az ideiglenesbe, ha pedig egyezik, akkor felszabadítja a törlendő csapat helyét. Ezután a hosszt csökkentjük egyel, felszabadítjuk a régi adatbázist, és az ideiglenes adatbázisra állítjuk a pointerét.
- `beolvasFajlbol():`
  - Paraméter nélküli függvény, amely megnyit egy fájlt, és abból először beolvassa az adatbázis hosszába az első sort, ha ez 0, akkor return-el kiugrik a függvényből. Ha nem, akkor egy ciklus segítségével végigfut a fájlban és egyesével beolvassa először a típust, a csapat alapadatait majd azon belül a

játékosokat is ciklussal. Ezután foglal helyet típus segítségével és az adatbázisba belemásolja. A függvény a végén bezárja a fájlt.

- kiirFajlba():
  - Paraméter nélküli függvény, amely megnyit egy fájlt és beleírja az első sorába az adatbázis hosszát, utána pedig egy ciklus segítségével meghívja az adatbázis minden tagjára a kiirFajlba(ofstream&) függvényt. A függvény a végén bezárja a megnyitott fájlt.
- ~Sportegyesulet():
  - Egy ciklus segítségével végig fut az adatbázison és minden elemét felszabadítja, majd a végén az adatbázisra mutató pointert is törli.

- **Csapat:**

- Csatat(string, string, integer, Jatekos\*):
  - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként csapatnévnek, edzőnek egy üres stringet ad meg, létszámnak 0-át és a játékos tömbnek nullptr-t. Egyébként a megadott értékeket állítja be.
- getCsatatnev():
  - A függvény paraméter nélküli és visszatér a csapat nevével.
- getEdzo():
  - A függvény paraméter nélküli és visszatér az edző nevével.
- getLetszam():
  - A függvény paraméter nélküli és visszatér a csapat létszámával.
- getJatekosok():
  - A függvény paraméter nélküli és visszatér a játékosok tömbjével.
- csere(Jatekos&, Jatekos&):
  - A függvény a két paraméterként kapott Jatekos helyét felcseréli.
- buborekRendezes(integer):
  - A függvény paraméterként megkapja, hogy mi alapján akarja a felhasználó rendezni a játékosokat, és aszerint buborék rendezéssel sorbaállítja a játékos tömb tagjait.
- csapatKiir():
  - Paraméter nélküli virtual függvény, ami a leszármazott osztályban valósítódik meg, egy csapatot kiír a szabványos kimenetre.
- kiirFajlba(ofstream&):
  - Virtual függvény, ami egy kimeneti fájl változót kap paraméterül, a leszármazott osztályban valósítódik meg, hogy egy csapatot kiír a kimeneti fájlba.
- ~Csapat():
  - Felszabadítja a játékos tömböt.

- **Foci:**

- Foci(string, string, integer, Jatekos\*, string):
  - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként csapatnévnek, edzőnek és másodedzőnek egy üres stringet ad meg, létszámnak 0-át és a játékos tömbnek nullptr-t. Egyébként a megadott értékeket állítja be.
- getEdzo2():

- A függvény paraméter nélküli és visszatér a foci csapat másodedzőjének nevével.
  - csapatKiir():
    - Paraméter nélküli függvény, amely kiír a szabványos kimenetre egy csapatot, amire meg lett hívva a függvény. Először kiírja a csapat alapadatait, majd egy ciklus segítségével kilistázza a játékosokat, ha a megadott létszám nem 0.
  - kiirFajlba(ofstream&):
    - Egy kimeneti fájlt kap paraméterül, majd a megkapott fájlba tabulátorokkal elválasztva beírja a csapat alapadatait, majd egy enterrel elválasztva megint csak tabulátorokkal elválasztva beírja a fájlba a játékosok adatait egy ciklus segítségével.
- Kosar:
- Kosar(string, string, integer, Jatekos\*, integer):
    - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként csapatnévnek, edzőnek egy üres stringet ad meg, létszámnak, pom-pom lányok számának 0-át és a játékos tömbnek nullptr-t. Egyébként a megadott értékeket állítja be.
  - getPompomszam():
    - A függvény paraméter nélküli és visszatér a kosár csapat pom-pom lányainak létszámával.
  - csapatKiir():
    - Paraméter nélküli függvény, amely kiír a szabványos kimenetre egy csapatot, amire meg lett hívva a függvény. Először kiírja a csapat alapadatait, majd egy ciklus segítségével kilistázza a játékosokat, ha a megadott létszám nem 0.
  - kiirFajlba(ofstream&):
    - Egy kimeneti fájlt kap paraméterül, majd a megkapott fájlba tabulátorokkal elválasztva beírja a csapat alapadatait, majd egy enterrel elválasztva megint csak tabulátorokkal elválasztva beírja a fájlba a játékosok adatait egy ciklus segítségével.
- Kezi:
- Kezi(string, string, integer, Jatekos\*, integer):
    - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként csapatnévnek, edzőnek egy üres stringet ad meg, létszámnak, támogatásnak 0-át és a játékos tömbnek nullptr-t. Egyébként a megadott értékeket állítja be.
  - getTamogatas():
    - A függvény paraméter nélküli és visszatér a kézi csapat évi támogatásával.
  - csapatKiir():
    - Paraméter nélküli függvény, amely kiír a szabványos kimenetre egy csapatot, amire meg lett hívva a függvény. Először kiírja a csapat alapadatait, majd egy ciklus segítségével kilistázza a játékosokat, ha a megadott létszám nem 0.
  - kiirFajlba(ofstream&):
    - Egy kimeneti fájlt kap paraméterül, majd a megkapott fájlba tabulátorokkal elválasztva beírja a csapat alapadatait, majd egy enterrel elválasztva megint csak tabulátorokkal elválasztva beírja a fájlba a játékosok adatait egy ciklus segítségével.

- **Jatekos:**

- `Jatekos(string, integer, integer):`
  - Paraméterrel és paraméter nélkül is meghívható konstruktor, ahol, ha nem adunk értéket, akkor alapértékként névnek egy üres stringet ad meg, míg értéknek és pontnak 0-át. Egyébként a megadott értékeket állítja be.

- **Gui függvények:**

- `menu(Sportegyesulet*):`
  - Paraméterként egy sportegyesületre mutató pointert kapó függvény, ami kiírja a menüt a szabványos kimenetre, és bekéri, hogy a felhasználó mit szeretne tenni, illetve meghívhatja a `bekerUjcsapat()`, `bekerTorolcsapat()`, `bekerJatekoskeres()`, `listazCsapatok()` és `listazSorrendben()` függvényt a megkapott sportegyesületre.
- `csapatJatekosai(integer):`
  - A függvény a csapat létszámát kapja meg paraméterül, majd a létszám segítségével foglal helyet a játékos tömbnek, ha a létszám 0, akkor üres tömbbel tér vissza. A foglalt tömböt feltölti üres játékosokkal. Ha a program használója szeretne hozzáadni játékosokat, akkor bekéri a szabványos bemenetről, majd végig fut a tömbön és az első üres játékos helyére beteszi az új játékos adatait. Ha nem akar többet hozzáadni többet, akkor egyből visszatér az üresekkel feltöltött játékos tömbbel. Ha az összes játékost megadja a felhasználó akkor `return`-öl a függvény a feltöltött játékos tömbbel.
- `bekerUjcsapat(Sportegyesulet*):`
  - Paraméterként egy sportegyesületre mutató pointert kapó függvény, ami a szabványos kimenetről, miután megadta a csapat típusát, egyesével bekéri az új csapat adatait, amit konstruktor segítségével összeállít. Ezután meghívja a megkapott sportegyesületre az `ujCsapat()` függvényt.
- `bekerTorolcsapat(Sportegyesulet*):`
  - Paraméterként egy sportegyesületre mutató pointert kapó függvény, amely szabványos bemenetről kéri be a program használatjától a törlendő csapat nevét, és egy ciklus segítségével megvizsgálja, hogy megtalálható-e a csapat az adatbázisban. Ha nem találja meg, akkor kiírja a szabványos kimenetre, hogy nem talált ilyen csapatot, ha pedig megtalálja, akkor meghívja a `torolCsapat()` függvényt a megkapott sportegyesületre.
- `bekerJatekoskeres(Sportegyesulet*):`
  - Paraméterként egy sportegyesületre mutató pointert kapó függvény, amely a felhasználótól bekéri a keresett játékos nevét, majd a megkapott sportegyesületre meghívja a `jatekosKeres()` függvényt.
- `bekerListazsorrendben(Sportegyesulet*):`
  - Paraméterként egy sportegyesületre mutató pointert kapó függvény, amely a felhasználótól bekéri, hogy mi alapján szeretné a játékosokat rendezni, majd a megkapott sportegyesületre meghívja a `listazSorrendben()` függvényt.