# Házi feladat

# Programozás alapjai 2.

# Tervezés

**Stancz Levente**

**RI8WCW**

**2023.04.25.**

# 1. Feladat

**Sportegyesület**

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

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.

# 2. Feladatspecifikáció

A feladatban a Csapat ősosztályból származik a Labdarugó, Kosárlabda és kézilabda osztály, így, ha esetleg az egyesület bővül egy új sportággal könnyen bővíthető a program is. A feladat nem specifikálja, de a nevek tárolásához létrehozok egy String nevű osztályt, mely a dinamikus memória foglalásért lesz felelős. Létrehozok, valamint egy Ember nevű ősosztályt melyből származtatom az Edző és PomPomLány osztályokat. A labdarugó csapatnak két edzője van, de a többi csapatnak is lehet adni a későbbiekben ezzel a módszerrel. A kosárlabdacsapat pompom lányait egy tömbben fogom tárolni (PomPomLány példányokat tárol). A felhasználó képes lesz új csapatokat felvenni, a labdarugó csapatnál a két edzőt is meg kell adni, valamint a kosárlabda csapatnál opcionális, hogy már a csapat létrehozásakor felvesz-e a felhasználó pompom lányokat. A csapatokat és adataikat lehet módosítani, ekkor az edzőket és pompom lányokat is lehet módosítani. Lehet, valamint még csapatot törölni, ilyenkor a csapat és minden hozzá kapcsolódó adat (edző, pompom lány) törlődik. Bővíthetőség szempontjából, később az Ember osztály leszármazottja lehet a Játékos osztály is, mely a játékosok nevét tárolja és mindegyik csapatnak tömbben lenne tárolva az összes játékosa.

# 3. Terv

## 3.1 Objektum/ok terv

A legfontosabb osztály az “Egyesulet” osztály. Ebben az osztályban egy heterogén kollekcióban csapatok lesznek nyilvántartva, ezek az egyesület csapatai (labdarúgó, kosárlabda, kézilabda). Az egyesuletnek van nyilvánosan elérhető adata, az egyesület neve (String). Két privát adata, az egyik a heterogén kollekció melyben a csapatokat tároljuk, a másik pedig, hogy éppen mekkora a kollekció mérete, nincsen fix méret, hiszen egy egyesületnek rengeteg csapata lehet. Az “Egyesulet” osztálynak van alapértelmezett konstrukora, mely a méretet 0-ra állítja, illetve van destruktora is, mely felszabadítja (törli) a heterogén kollekció elemeit, a csapatokat. A felsoroltokon kívül az alábbiak még az “Egyesulet” osztály metódusai:

- **hozzaad(csapat: Csapat):bool**
  Hozzáad egy csapatot a kollekcióhoz. (Mivel memória foglalások is történnek így false-t a visszatérési érték, ha nem sikerült a csapatot hozzáadni a kollekcióhoz)
- **eltavolit(csapat: Csapat):bool**
  Eltávolít egy csapatot a kollekcióból. (Mivel memória foglalások is történnek így false-t a visszatérési érték, ha nem sikerült a csapatot eltávolítani a kollekcióból)
- **meret():size_t**
  Visszaadja a kollekció méretét
- **kiir():void**
  Kiírja az összes elem adatait.
- **operator[](index: int):Csapat***
  Visszaadja a megkapott indexhez tartozó csapat pointerét. (Hibát dob ha az index nem megfelelő)

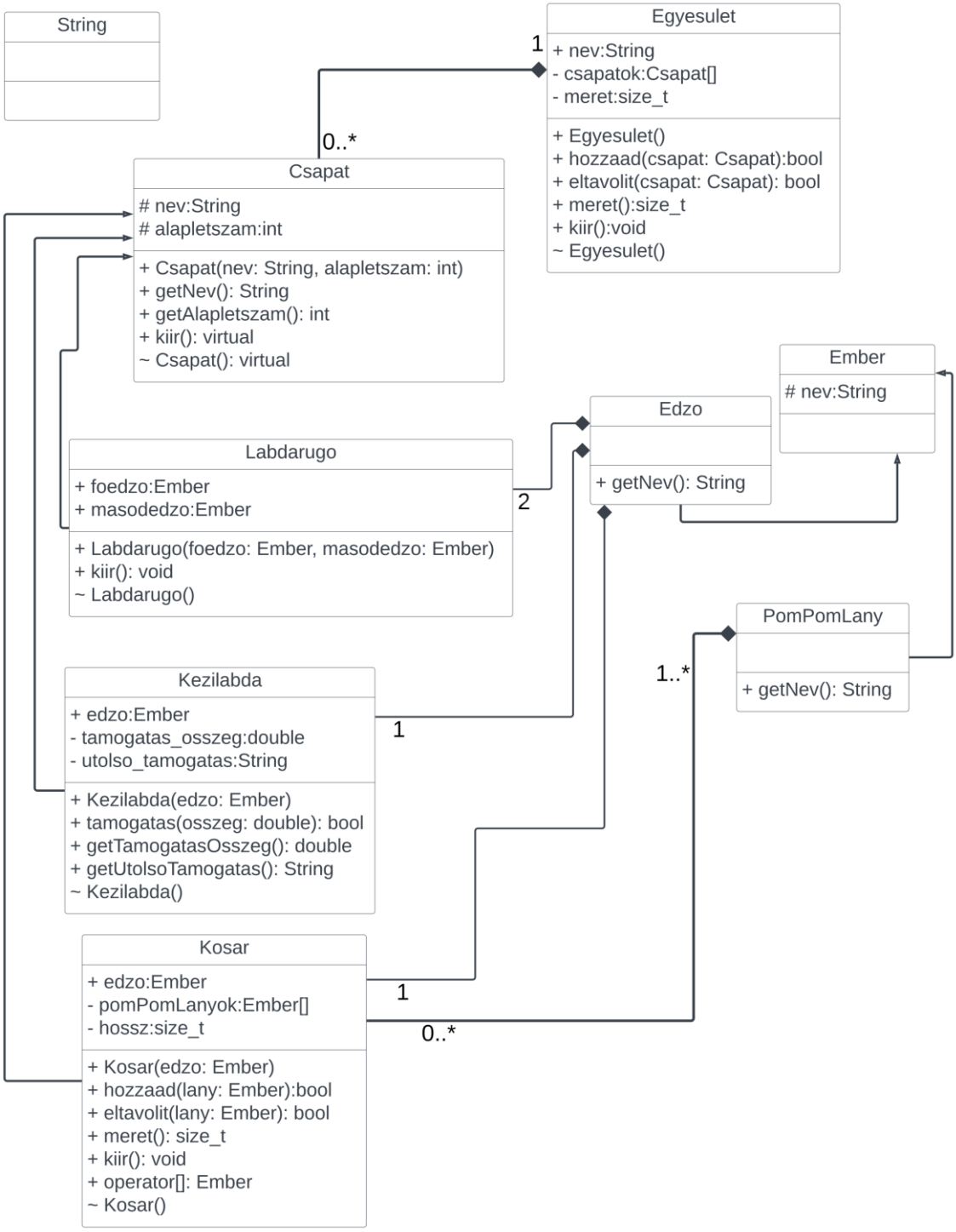
A Csapat ősosztály adattagjai a következők:

- **nev:String**
  Neve minden csapatnak van (Például: Darazsak U19).
- **alapletszam:int**
  Minden csapatnak van alaplétszáma.

A Csapat ősosztály metódusai:

- **Csapat(nev: String, alapletszam: int)**
  Alapértelmezett konstruktor.
- **getNev():String**
  Visszaadja a “nev” értékét.
- **getAlapletszam():int**
  Visszaadja az “alapletszam” értékét.
- **kiir():virtual**
  Kiírja a csapat adatait a konzolra.
- **Csapat():virtual**
  Destruktor.

***A további osztályokat és az osztályok közötti kapcsolatot a következő UML osztálydiagram jelképezi és magyarázza el:***



Fontosabb kapcsolatok leírása:

- **Csapat --> Egyesulet**
  Egy csapat csak egy egyesülethez tartozhat, egy egyesületnek több csapata is lehet.
- **Labdarugo --> Edzo**
  Feladatspecifikáció szerint a labdarúgó csapatnak két edzője van.
- **Kezilabda --> Edzo**
  A kezilabda csapatnak egy edzője van.
- **Kosar --> Edzo**
  A kosárlabda csapatnak egy edzője van.
- **Kosar --> PomPomLany**
  A kosárlabda csapatnak legalább egy pompomlánya kell legyen, egy pompomlány szurkolhat több kosárlabda csapatnak is.

## 3.2 Algoritmusok

Tartományon kívüli elemek lekérése

if i < 0 akkor

throw alulindexelés

else if i >= N akkor

throw túlindexelés

else

…

Minden hiba esetén a program az alapértelmezett std:*_exception hibát dob, valamint a konzolon egy értelmezhető és a hibához releváns hibaüzenet jelenik meg a felhasználónak.

## 3.3 Tesztprogram

A tesztprogramot a főprogramban lehet elindítani. Mikor a tesztet lefuttatjuk az egy fájlból olvas lehetséges adatokat, melyekkel a program függvényeit és algoritmusait teszteli. A tesztesetek és eredményeiket a felhasználó számára a konzolon megjelennek. A tesztesetek között tesztelve van a túlindexelés és hibás adatok megadása adat felvétel vagy módosítás esetén.