BKSZC Pogány Frigyes Szakgimnázium

Szoftverfejlesztő

5421305

# SPORTVERSENY

Horváth Emese Tóth József

2/14ED. Konzulens

Budapest 2020

Tartalom

[SPORTVERSENY 1](#_Toc36907677)

[1. Bevezetés 2](#_Toc36907678)

[1.1. Kitűzött cél 2](#_Toc36907679)

[1.2. Témaválasztás Indoka 3](#_Toc36907680)

[1.3. A felhasznált fejlesztő eszközök 3](#_Toc36907681)

[1.4. Segédeszközök: 3](#_Toc36907682)

[1.5. Rendszerkövetelmények 3](#_Toc36907683)

[**1.5.1.** **Windows** 3](#_Toc36907684)

[**1.5.2.** **Max OS X** 4](#_Toc36907685)

[**1.5.3.** **Linux** 4](#_Toc36907686)

[1.6. Telepítés 4](#_Toc36907687)

[2. Rendszertervezés 5](#_Toc36907688)

[3. Felhasználói dokumentáció 7](#_Toc36907689)

[3.1. Az „Országok” fül használata 7](#_Toc36907690)

[3.2. A „Versenyzők” fül használata 8](#_Toc36907696)

[3.3. Az „Egyéni sportok” fül használata 10](#_Toc36907700)

[3.4. A „Küzdősportok” fül használata 11](#_Toc36907704)

[4. Fejlesztői dokumentáció 12](#_Toc36907708)

[4.1. Adatbázis szerkezet 12](#_Toc36907709)

[5.2. Csomagok 17](#_Toc36907715)

[7. Továbbfejlesztési lehetőségek 28](#_Toc36907739)

[8. Források 30](#_Toc36907740)

# Bevezetés

## Kitűzött cél

A „Sportverseny” szakdolgozat célja, hogy egy felhasználó által áttekinthető, működésében megbízható, szoftverkörnyezetben szempontjából rugalmas program elkészítése, ami egy több versenyszámból álló sporteseményt nyomon követni.

A Java felépítésnek köszönhetően a program többféle operációs rendszerezeken lesz képes majd lefutni. A Java alapú alkalmazások futtatásához szükség van egy Java futtató környezetre, melynek technológiájából következik, hogy a szoftver nem foglal túl nagy méretet.

## Témaválasztás Indoka

Mivel az ilyen típusú alkalomhoz számos, egymással kapcsolatban álló adatot kell rögzíteni ezért megfelelő témának tekintettem ahhoz, hogy fejlesszem a java programozói és SQL kezelési képességeimet, emellett legyen lehetőségem ezeket prezentálni.

Ezt leszámítva az eredetileg idén esedékes Olimpia inspirálta témát, és régebben amúgy is érdekelt több sport végzése is, mint például a foci, kosárlabda és úszás.

## A felhasznált fejlesztő eszközök

Netbeans IDE 8.2   
Xamppserver

Gluon SceneBuilder

NotePad++

Microsoft Word

Microsoft PowerPoint

Paint (A screenshotok létrejozásához)

## Segédeszközök:

<https://stackoverflow.com/>

<https://www.reddit.com/>

<https://www.w3schools.com/>

<http://www.iconarchive.com/>

<http://soundbible.com/>

<https://www.youtube.com/>

## Rendszerkövetelmények

### **Windows**

Windows 10

Windows 8.x

Windows 7 SP1

Windows Vista SP2

Windows Server 2008 R2 SP1(64-bit)

Windows Server 2012 és 2012 R2(64-bit)

RAM 128M Merevlemez: 124 MB JRE; 10MB java Update

Processor: Minimum Pentium 2 266MHz processor

Böngésző: Internet Explore 9 és felett, Firefox, Edge, Chrome

### **Max OS X**

Intel-alapú Mac és Mac OS X 10.7.3 (Lion) vagy későbbi

Adminisztrátor jogok a telepítéshez,

64-bites böngésző

### **Linux**

Oracle Linux 5.5+

Oracle Linux 6.x (32-bit), 6.x (64-bit)3

Oracle Linux 7.x (64-bit)3 (7u67 vagy újabb)

Red Hat Enterprise Linux 5.5+, 6.x (32-bit), 6.x (64-bit)3

Red Hat Enterprise Linux 7.x (64-bit)3 (7u67 vagy újabb)

Suse Linux Enterprise Server 10 SP2, 11.x

Suse Linux Enterprise Server 12.x (7u75 vagy újabb)

Ubuntu Linux 10.04 vagy újabb

Böngésző: Firefox 3.6 vagy újabb

## Telepítés

A java telepítéséhez legegyszerűbb és legbiztonságosabb módja a java.com -on keresztül való letöltés, ami ráadásul ingyenes. Valamint szükséges egy SQL adatbáziskezelő is, szintén egy egyszerű és ingyenes lehetőség az XAMPP letöltése a következő címről: <https://www.apachefriends.org/hu/download.html>

Ezek után az adatbáziskezelőben létre kell hozni egy új adatbázist „sport” néven,

utf8mb4\_hungarian\_ci egyeztetéssel.

A képen képernyőkép látható

Automatikusan generált leírás

Ha ezzel készen van, importálja a négy SQL fájlt a sport adatbázisba a következő sorrendben:

„orszag” → „versenyzok” → „egyeni” → „kuzdo”

Az utolsó kettő fájl sorrendjét fel lehet cserélni, de az „orszag” muszáj hogy mindig az első, valamint a „versenyzok” mindig a második importált fájlnak lennie, különben nem lehet többit beolvasni az adatbázisba).

Ha sikeresen importálta az SQL fájlokat, el lehet indítani a sport.jar állományt és lehet használni a programot. A program az alapértelmezett localhost-ra van állítva, így nem szükséges új felhasználót létrehozni az adatbázis kezelőben.

# Rendszertervezés

Mivel a program célja, hogy nyomon kövesse a sporteseményeket és tárolja az ebből adódó adatokat, ezért gyakorlatilag nincs definiált vége a programnak, a felhasználó addig futtathatja, ameddig igényt tart rá. Viszont ideálisan ilyen felhasználási sorrendre van tervezve a program:

Nyilak színének magyarázata:

**→** Automtikus, következmény

**→** Ha a válasz nem

**→** Ha a válasz igen

Adatok módosítása vagy törlése, ha szükséges

Helyzések, pontszámok megtekintése

Új verseny(ek) hozzáadása

Versenyzőkkénti és országos pontszámának számítása

Új versenyző(k) hozzáadása

Új ország hozzáadása

START

Adatbázis beolvasása

# Felhasználói dokumentáció

## A képen képernyőkép látható Automatikusan generált leírásAz „Országok” fül használata

### Oszlopok

### **Név** - Az ország neve **Versenyzők** – Megmutatja, hogy országonként mennyi versenyző van jelenleg regisztrálva, azonban nem számolja bele a visszavonult vagy diszkvalifikált versenyzőket **Pontszám** – Az országonként elért pontszám, ez összesíti az összes olyan versenyző pontját, aki ahhoz a bizonyos országhoz tartozik, a versenyzők oszloppal ellentétben ez figyelembe veszi a visszavonult státuszú versenyzőket is.

### Gombok

### **Módosít** - A táblázatból kiválasztott ország nevének módosítása **Ment** – Az ország hozzáadása az adatbázishoz **Töröl** - A táblázatból kiválasztott ország törlése **Új** – A beviteli mező letisztítása, így a név mező üressé válik

### Beviteli mező A többivel füllel ellentétben ebben a fülben csak egy beviteli mező van, az ország neve.

## A képen képernyőkép, zöld, számítógép látható Automatikusan generált leírásA „Versenyzők” fül használata

### Oszlopok **Név** - Az versenyző neve **Ország** – Megmutatja, hogy melyik országot képviseli az adott versenyző **Pont** – A versenyző egyéni összestett pontszáma

### Gombok **Módosít** - A táblázatból kiválasztott versenyző adatainak módosítása **Ment** – A versenyző hozzáadása az adatbázishoz **Töröl** - A táblázatból kiválasztott versenyző törlése **Új** – A beviteli mező letisztítása **Keresés** – Új profilkép keresése, a felhasználó megadhat egy új .png vagy .jpeg fájlt

### Beviteli mező **Név** - A versenyző neve **Ország** - Beállítja, hogy melyik országot képviseli az adott versenyző **Kor** - Azversenyző kora **Edző** - A versenyző edzőjének neve, ha egyáltalán van neki **Státusz** - A versenyző státusza, erre négy opció van („Aktív”, „Beteg”, „Visszavonult”, „Diszkvalifikált”) **Megjegyzés** – Egyéb megjegyzés a versenyzőt illetően, ha van.

## Az „Egyéni sportok” fül használata

### A képen képernyőkép, monitor, számítógép látható Automatikusan generált leírásOszlopok **Versenyszám** – Az egyéni versenyszám neve **I. Helyezett** – Megmutatja, hogy melyik versenyző ért el a I. helyezést **II. Helyezett** – Megmutatja, hogy melyik versenyző ért el a II. helyezést **III. Helyezett** – Megmutatja, hogy melyik versenyző ért el a III. helyezést **Dátum** – Megmutatja, hogy mikor történt az adott verseny

### Gombok **Módosít** - A táblázatból kiválasztott egyéni versenyszám adatainak módosítása **Ment** – Az egyéni versenyszám hozzáadása az adatbázishoz **Töröl** - A táblázatból kiválasztott egyéni versenyszám törlése **Új** – A beviteli mező letisztítása

### Beviteli mező **Név** - Az egyéni versenyszám neve **Dátum** - Beállítja, hogy mikor történt az adott verseny **I** – Az első helyezett neve **II** – A második helyezett neve **III** – A harmadik helyezett neve

## A képen képernyőkép, zöld, monitor, számítógép látható Automatikusan generált leírásA „Küzdősportok” fül használata

### Oszlopok **Versenyszám** – A küzdősport neve **Nyertes** – Megmutatja, hogy melyik versenyző nyert **Vesztes** – Megmutatja, hogy melyik versenyző vesztett **Dátum** – Megmutatja, hogy mikor történt az adott verseny

### Gombok **Módosít** - A táblázatból kiválasztott küzdősport adatainak módosítása **Ment** – A küzdősport hozzáadása az adatbázishoz **Töröl** - A táblázatból kiválasztott küzdősport törlése **Új** – A beviteli mező letisztítása

### Beviteli mező **Név** - Az egyéni versenyszám neve **Dátum** - Beállítja, hogy mikor történt az adott verseny **Nyertes** – A nyertes neve **Vesztes** – A vesztes neve

# Fejlesztői dokumentáció

* 1. Adatbázis szerkezet  
     A program MySQL adatbázist használ, utf8mb4\_hungarian karakterkészlettel. Maga az adatbázis négy táblából áll, amelyek több helyen is hivatkoznak egymásra.  
     1. Az országok táblájaA képen képernyőkép látható

        Automatikusan generált leírás  
        Ez a tábla tartalmazza az országok adatait, ez az egyetlen tábla amit importálni lehet akár a többi nélkül is, ezért telepítéskor szükséges ezzel kezdeni.

**id** - egy integer ami az ország azonosítója, automatikusan generálódik

**name** – az ország neve, legfeljebb harminc karakter hosszú lehet és nem maradhat üresen

**letszam** – integer ami jelzi hogy egy országnak hány aktív vagy beteg státuszú versenyzője van az adatbázisban.

**pont** – integer ami az országokként összpointszámot képviseli, később majd a program azoknak a versenyzőknek összesíti a pontszámát, akik ugyanazt az országot képviselik

* + 1. A versenyzők táblája

A képen képernyőkép látható

Automatikusan generált leírásEz a tábla tartalmazza a versenyzők adatait, ezt a tábla már hivatkozik az előbb említett országok adatait tartalmazó táblára, így nem ajánlatos ezt a táblát importálni előszőr.

**id** - egy integer, ami a versenyző azonosítója, automatikusan generálódik  
**nev** - a verseyző neve, legfeljebb harminc karakter hosszú lehet és nem maradhat üresen  
**orszagid** - egy integer, ami jelzi, mi az azonosítója annak az országnak, amit az adott versenyző képvisel. Nem lehet üres és szükséges, hogy legyen egy létező ország ezzel az azonosítószámmal  
**pont** - a versenyző összes elért pontszámát képviselő integer, függ attól, hogy milyen típusú versenyben vett részt, valamint, hogy milyen helyezét ért el egyéni sport esetén  
**kor** – egy legfeljebb két karakter hosszú integer, ami jelzi, hogy ilyen idős az adott versenyző  
**edzo** – a versenyző edzőjének neve, legfeljebb 30 karakter hosszú lehet. Amennyiben nincs értek megadva ehhez az oszlophoz, úgy automatikus „ismeretlen” lesz kiírva.  
**status** - a versenyző státuszát jelzi (pl. Aktív, Beteg, stb.), legfeljebb 20 karakter hosszúságú lehet. Amennyiben nincs hozzá érték megadva, az adatbázis automatikusan aktívnak tekinti a versenyzőt, azaz lehet regisztrálni új versenyekhez.  
**comment** – Amennyiben van egyéb megjegyzés a versenyzőt illetően, itt lehet jelezni, akár 255 karakter hosszú is lehet. Viszont nem szükséges ehhez az oszlophoz értéket adni, maradhat akár üresen is.  
**pic** – egy longblob ami tartalmazza a versenyző fényépét, amennyiben van róla.

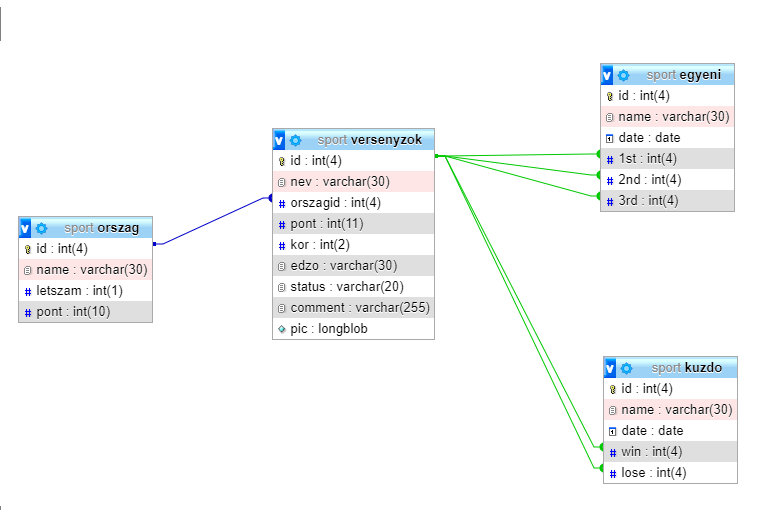
### A képen képernyőkép látható Automatikusan generált leírásAz egyéni sportok táblája

**id** - egy integer, ami az egyéni verseny azonosítója, automatikusan generálódik  
**name** – az egyéni verseny neve, legfeljebb harminc karakter hosszú lehet és nem maradhat üresen  
**date** – rögzíti, hogy melyik nap történt a megadott verseny, nem maradhat üresen  
**1st** - egy integer, ami jelzi, mi az azonosítója annak a versenyzőnek, aki az adott versenyen részt vett és első helyezést ért el. Nem lehet üres és szükséges, hogy legyen egy létező versenyző ezzel az azonosítószámmal  
**2nd** – egy integer, ami jelzi, mi az azonosítója annak a versenyzőnek, aki az adott versenyen részt vett és második helyezést ért el. Nem lehet üres és szükséges, hogy legyen egy létező versenyző ezzel az azonosítószámmal  
**3rd** – egy integer, ami jelzi, mi az azonosítója annak a versenyzőnek, aki az adott versenyen részt vett és harmadik helyezést ért el. Nem lehet üres és szükséges, hogy legyen egy létező versenyző ezzel az azonosítószámmal

### A képen képernyőkép látható Automatikusan generált leírásA küzdősportok táblája

**id** - egy integer, ami a küzdősport azonosítója, automatikusan generálódik  
**name** – a küzdősport neve, legfeljebb harminc karakter hosszú lehet és nem maradhat üresen  
**date** – rögzíti, hogy melyik nap történt a megadott küzdősport, nem maradhat üresen  
**win** - egy integer, ami jelzi, mi az azonosítója annak a versenyzőnek, aki az adott küzdősportban részt vett és nyert. Nem lehet üres és szükséges, hogy legyen egy létező versenyző ezzel az azonosítószámmal  
**lose** – egy integer, ami jelzi, mi az azonosítója annak a versenyzőnek, aki az adott küzdősportban részt vett és vesztett. Nem lehet üres és szükséges, hogy legyen egy létező versenyző ezzel az azonosítószámmal

### Kapcsolatok

Több kapcsolat is jelen van az adatbázisban, amit az alábbi ábra jelez

## Csomagok

### CSS A grafikus felület stílusát változtatja meg a benne lévő css fájlsegítségével. Igyekeztem olyan stílust létrehozni, ami nem túl világos, mivel így jobban íméli a felhasználó szemét.

### Icons Hat ikont tárol, a program futtatásakor véletlenszerűen jelenik meg ezek közül az egyik a tálcán. Forrásuk az Icon Archive.

### MP3 A SounBible-ről szerzett hang effektusokat tárolja.

### Panel Amennyiben eldöntendő kérdést tesz fel a felhasználónak a program, vagy hibaüzenetet kell jeleznie, ez a package biztosítja, hogy a megfelelő típusú ablak megjelenjen.

### pfps Szintén az Icon Archive-ról szerzet képek, arra szolgál hogy teszteléskór ne kelljen a felhasználónak külön fényképeket keresgélni az internet, hogy tesztelje a fénykép betöltésével és mentésével foglalkozó metódusok működését. Az egyetlen kép amit a program ténylegesen igényel az a „unknown.png” fájl, mivel ez az alapértelmezett profilkép, ha versenyzőnek nincs sajátja.

### sport2 A program fő csomagja, Ez tartalmazza a grafikus felületért osztályok nagy részét és az adatbázist beolvasó osztályt is. Mivel ez tartalmazza a legtöbb fájlt, ezt a következő témában fogok részletessebben kitérni rá.

* 1. A sport2 csomag osztályai

### A képen képernyőkép látható Automatikusan generált leírásA képen képernyőkép, madár látható Automatikusan generált leírásA képen képernyőkép látható Automatikusan generált leírásDB A „DB” osztály felelős a program és az adatbázis közti kommunikációért. Az adatbázishoz való csatlakozáshoz szükséges egy URL cím, valamint felhasználónév és jelszó. Az egyszerűség kedvéért az alapértelmezett „root” felhasználónevet adtam meg, valamint a jelszó szakaszhoz csak egy üres stringet adtam. Mivel négy különböző adatbázist igényel a program (Ország, Versenyzők, Egyéni Sportok, Küzdősportok) ezért mind a négy táblához külön-külön metódusok vannak azok alapvető funkcióihoz (adatbázis beolvasás. új adat hozzáadása, módosítása, vagy törlése): A képen képernyőkép látható Automatikusan generált leírás A versenyzők adatinak kezelése struktúra szempontjából hasonlít az országéhoz, azonban az SQL parancs itt már hosszabb, annak köszönhetően, hogy az adatbázissal ellentétben itt már nem a versenytő országának azonosítóját rögzíti, hanem az ország tényleges nevét. Emellett úgy döntöttem, hogy a versenyzők esetleges fényképeit nem ebben a metódusban olvasom be, igy nem is lesz szükséges a versenyző osztályban rögzíteni. Ezt azért tartottam hasznosnak, mivel így kevesebb helyett foglalnak a A képen képernyőkép látható Automatikusan generált leírásprogramban az adatok, és kevésbé lassítja le azt. Az összes tábla közül az egyéni versenyek kezelése igényelni a leghosszabb SQL parancsot, mivel nem az első három helyezettek azonosítóját fogja tárolni egyéni A képen képernyőkép látható Automatikusan generált leírássportokhoz tartozó osztály, hanem a versenyzők teljes nevét.

### A képen képernyőkép látható Automatikusan generált leírás A fent említett metódusok mellet még több törlés, mentés, beolvasás, és módosítás metódus is van, azonban mivel felépítés szempontjából hasonlóak, ezért nem tartom szükségesnek ezek bemutatását a szakdolgozat ezen részén. Azonban ezeket leszámítva még van néhány eltérőbb típusú metódus a DB osztályban. Ezek közül van két frissítést végrehajtó metódus van, az első az orszag\_PV\_frissit() metódus, ami kiszámolja az ország pontszámát (összesítve az azonos nemzetiségű versenyző pontszámait). Ezután megszámolja, mennyi olyan versenyzője van minden egyes országnak, akinek a státusza vagy aktív, vagy beteg (azaz se nem visszavonult, se nem diszkvalifikált). A másik ilyen típusú metódus a versenyzok\_pont\_frissit(), ami kiszámítja egy versenyzőnek mennyi az összestett pontszáma, a diszkvalifikált versenyzőket leszámítva, akik függetlenül attól, hogy ilyen helyezést értek el bármi versenyen, mindig nulla marad a pontszámuk. A metódus előszőr +3 pontot ad minden egyéni sportnál elért első helyezésért, +2 második helyezésért, +1 pontot harmadik helyezésért, és végül újabb +1 pontot minden küzdősport megyeréséért.

### A képen képernyőkép látható Automatikusan generált leírás

### További említésre méltó metódusok getImageById(), és a setImageById(). Ezek foglalkozna a versenyző profilképének betöltésével, valamint rögzítésével az adatbázisban.

### A képen képernyőkép látható Automatikusan generált leírás

### A képen képernyőkép látható Automatikusan generált leírás

### FXML Controller

### A képen képernyőkép látható Automatikusan generált leírásEbben az osztályban felhasznált metódusok végzik a felhasználói kérésekhez szükséges adatok feldolgozását és továbbítását a grafikus felületnek. A profilkép megadásáért felelő metódus, ellenőrzi előszőr, hogy a fájl formátuma és mérete megfelelő-e

### A képen képernyőkép látható Automatikusan generált leírásA képen képernyőkép látható Automatikusan generált leírásAz initiailze metódus gondoskodik arról, hogy az adatbázisból minden érték a megfelelő helyen jelenjen meg a grafikus felületen.

### Emellett fölhasználtam néhány általános SQL parancsokon alapuló metódust is, így például nem kell az adatokat módosító műveleteknél több, hasonló szerkezetű metódust létrehozni, lehet ugyanazt a metódust előhívni. A képen képernyőkép látható Automatikusan generált leírás

### Sport2 osztály

### A sport2 osztály az applikáció leszárazottja, ebben az osztályban

1. választja ki a program a tálcai ikont (6 megadott fájl közül választ ki véletlenszerűen), a grafikus felület stílusát megnevezi, beállítja a program címét, valamint átméretezhetetlenné alakítja a programot.  
   A képen képernyőkép látható

   Automatikusan generált leírás

### A képen beltéri, asztal, laptop, tartás látható Automatikusan generált leírásorszagok osztály Az országok adatait tárolja és hívja elő a hozzá tartozó konstruktort, valamint getter és setter metódusokat.

### A képen tartás látható Automatikusan generált leírásversenyzok osztály A versenyzők adatait tárolja (fényképet leszámítva) és hívja elő a hozzá tartozó konstruktort, valamint getter és setter metódusokat.

### A képen asztal, képernyő, telefon, mobiltelefon látható Automatikusan generált leírásegyeni osztály Az egyéni sportok adatait tárolja és hívja elő a hozzá tartozó konstruktort, valamint getter és setter metódusokat.

### A képen képernyő, tartás, monitor, férfi látható Automatikusan generált leíráskuzdo osztály A küzdősportok adatait tárolja és hívja elő a hozzá tartozó konstruktort, valamint getter és setter metódusokat.

### A képen képernyőkép látható Automatikusan generált leírássounds osztály A PlaySound() metódust tartlmazza, a fájl útvonalán az egyetlen változó maga a lejátszandó mp3 fájl neve.

# Továbbfejlesztési lehetőségek

Több ötletem is lett volna, hogy az program autentikus legyen, viszont ezeket részben időhiány miatt, részben azért, mert átgondolva ezek a tervezett adatbázisok túlságosan hasonlítottak már eredetileg létező adatbázisokhoz, túl repetitívnek tűnt.  
  
Eredetileg szerettem volna külön adatbázist létrehozni a versenyzők edzőinek, ne csak a nevük legyen feltüntetve a versenyzők adatbázisában. A program figyelembe vette volna az edzők nemzetiségét, és biztosította volna, hogy csak azonos országot képviselő versenyzőkhöz regisztrálhatta volna hozzá a felhasználó. Ezt leszámítva az edző adatainak felépítése hasonlított volna a versenyzőkhöz, például tartalmazott volna nevet, kort, státuszt, stb.  
  
Emellett lehetet volna külön adatbázist létrehozni sportcsapatoknak is, ahol a versenyzőkhöz hasonlóan jelen lett volna, milyen országot képviselnek. A csapatokhoz külön lett volna egy új sport kategória, amihez csak csapatokat lehet regisztrálni, a csapatjátékok (az egyéni- és küzdősportokban viszont nem vehettek volna részt). Ezt követően a program hozzá adta volna a csapatok által szerzet pontszámokat az országos pontszámokhoz.  
  
További fejlesztési opció a versenyzők adatainak részletesebbé alakítása, például magasság, születésnap, hány éve profi sportoló (figyelembe véve, hogy pl. egy 18 éves sportolónak nem lehet 25 év tapasztalata), magasság, súly, stb. Hasznos fejlesztési lehetőség a versenyző nemének megadása, amivel a program ráadásul ellenőrizni tudná, hogy csak azonos neműek versenyezzenek egymással. Mivel idősporolásért a nem beállítását kihagytam, jelenleg csak női nevek szerepelnek az adatbázisban a program első indításakor (bár ezt a jelenlegi program verzióban a felhasználó akármikor átírhat).

# Források

1. Stack Overflow - <https://stackoverflow.com/>
2. NetBeans - <https://netbeans.org/>
3. XamppServer - <https://www.apachefriends.org/hu/download.html>
4. Reddit - <https://www.reddit.com/>
5. W3schools - <https://www.w3schools.com/>
6. Icon Archive - <http://www.iconarchive.com/>
7. SoundBible - <http://soundbible.com/>
8. Youtube - <https://www.youtube.com/>