

Junior Java backend fejlesztő képzés

Projektmunka dokumentáció



Asztalitenisz-bajnokságot kezelő program

2023. október – 2024. augusztus.

Tartalomjegyzék

| | |
|--------------------------------------------|----|
| A program leírása | 3 |
| A program által használt adatok | 3 |
| Röviden a bajnokság lebonyolításáról..... | 3 |
| A három külön program: | 3 |
| FONTOS! Utasítások | 4 |
| Menük és képernyőképek | 5 |
| Főmenü..... | 5 |
| 1 – A bajnokság állása | 5 |
| 2 – Bejelentkezés kapcsolattartóként | 7 |
| 3 – Bejelentkezés adminként | 7 |
| Diagramok és ábrák..... | 8 |
| Folyamatábra..... | 8 |
| Osztálydiagramok..... | 9 |
| Adatbázis táblái | 12 |
| Funkciók és megoldások..... | 13 |
| Kért tartalom | 13 |
| További tartalmak | 13 |
| Összefoglalás | 14 |

A program leírása

A projektmunkám egy asztalitenisz-bajnokság lebonyolítására szolgáló program, amelyet a www.bpatsz.hu (Budapesti Asztalitenisz Szövetség) weboldal alapján készítettem el. A program Java nyelven íródott, amelyben JDBC segítségével az SQL Server Management Studio parancsait használja egy adatbázis kezelésére, melyben a csapatok, játékosok, eredmények stb. tárolása történik.

A program által használt adatok

A csapatnevek, a hazai pályák helyszínei és a csapatok által használt labdák az oldalon nyilvánosan elérhetőek (a legtöbb helyszín publikus, pl. sportegyesület vagy iskolai tornaterem), így a programban a 2023/2024-es szezon BP2 kategóriájában indult 14 csapatának valós adatait használtam fel. A játékosok személyes adatainak kezelése ettől eltér: minden csapat legalább 4 (ám jellemzően több) random generált nevű játékosból áll. A játékvezetők nevei szintén véletlenszerűek.

Röviden a bajnokság lebonyolításáról

A bajnokság célja, hogy egy félévszezon (pl. őszi) minden csapat játsszon egy csapatmérkőzést minden más csapat ellen – majd a következő szezonban egy visszavágót. Csapatonként heti egy mérkőzés van felváltva: az egyik héten hazai pályán, a következőn idegenben.

Minden csapat négy játékosból áll ki (a program a cserékkel és a hiányos felállással nem foglalkozik), és mindkét csapat mind a négy játékosát lejátszik egy egyéni meccset az ellenfél csapatának mind a négy játékosával. Ez összesen 16 egyéni mérkőzést jelent, ami a csapatmérkőzés végeredményét és a pontok sorsát is eldönti: a győzelem 2 pont, a döntetlen 1 pont, a vereség 0 pont.

A három külön program:

A program 3 részből tevődik össze, amelyek az alábbi funkciókat látják el:

1. Adatbázistáblák és alapadatok

Az első rész fájlból beolvassa a csapatneveket és -adatokat, véletlenszerűen fiktív nevű játékosokat generál, a szintúgy külső fájlban tárolt névlistákból. Itt történik még továbbá az adatbázis tábláinak létrehozása és feltöltése – utóbbinál a csapat- és egyéni mérkőzéseket tároló táblákat kivéve.

2. Sorsolás és szimuláció

A második rész a 2023/2024-es szezonhoz készít egy sorsolást, és leszimulálja az egyes csapat- és egyéni mérkőzések eredményét. A sorsoláshoz a Round Robin rendező elvet használja. Az egyéni meccseknél kvázi kockadobással dől el az eredmény egy 1–6-os skálán, ám minden játékos véletlenszerűen kap egy 1 és 6 közötti értékű „erősség” pontot, amely ezt a randomizált eredményt az erősebb játékos felé billenti – hogy az így kialakuló játékos-tabella egy valósághű képet kapjon.

3. Felhasználói felület

Ez lényegében az inspirációt adó www.bpatsz.hu oldalon is megtalálható felhasználói funkciókat valósítja meg: eredmények, tabellák és információk lekérdezése –híd a felhasználó és az adatbázis között. Kapcsolattartóként bejelentkezve a Hercules SE II. – Topspin csapatának adatai módosíthatók.

FONTOS! Utasítások

A programokat az alábbi sorrendben kell futtatni:

- Adatbázis létrehozása **TableTennisV2** néven
- Programok futtatási sorrendje:
 - Dbfiller – beolvasás fájlból, táblák létrehozása
 - Simulations – eredmények random szimulálása
 - Table_tennis_manager – felhasználó-adatbázis kapocs
- Bejelentkezés az adatbázisba:
 - Felhasználónév: java
 - Jelszó: java
- Bejelentkezés kapcsolattartóként:
 - Felhasználónév: Levi
 - Jelszó: Levi
 - Érintett csapat: Hercules SE II. – Topspin
 - DB-kapcsolattartó: Hornják Levente
- Bejelentkezés rendszergazdaként:
 - Felhasználónév: Admin
 - Jelszó: Admin
 - Érintett csapat: nincs

Menük és képernyőképek

Jelmagyarázat:

+ – az adott menüpont további menüt vagy választási lehetőséget nyit meg

X. – a pont előtti szám a menüpontnál a kódban nem szerepel, csak a jobb átláthatóságot szolgálja

Zöld – működik

Piros – fejlesztési lehetőség, egyelőre nem működik

Főmenü

+ 1 - A BAJNOKSÁG ÁLLÁSA

+ 2 - BEJELENTKEZÉS KAPCSOLATTARTÓKÉNT

+ 3 - BEJELENTKEZÉS ADMINKÉNT

0 – KILÉPÉS

```
FŐMENÜ
1 - A BAJNOKSÁG ÁLLÁSA
2 - BEJELENTKEZÉS KAPCSOLATTARTÓKÉNT
3 - BEJELENTKEZÉS ADMINKÉNT
0 - KILÉPÉS
Válasszon: █
```

1 – A bajnokság állása

Előtte: szezonválasztó

```
Eddig szezonok:
1 - 2023/2024
2 - 2024/2025
Válasszon: █
```

+ 1.1 - SORSOLÁS

1.2 - AKTUÁLIS FORDULÓ

1.3 - CSAPATTABELLA

1.4 - JÁTÉKOSTABELLA

1.5 - EGY CSAPAT JÁTÉKOSAI

1.6 - EGY CSAPAT ADATAI

0 – VISSZA

```
A BAJNOKSÁG ÁLLÁSA
1 - SORSOLÁS
2 - AKTUÁLIS FORDULÓ
3 - CSAPATTABELLA
4 - JÁTÉKOSTABELLA
5 - EGY CSAPAT JÁTÉKOSAI
6 - EGY CSAPAT ADATAI
0 - VISSZA
Válasszon: █
```

1.1 – SORSOLÁS

Fordulók:

Megjeleníti a fordulók listáját, és hogy az egyes fordulókban mely csapatok csaptak össze. A fordulók előtti sorszámokkal az egyes fordulók részletesebben is megtekinthetők.

| | | | |
|---------------------------------|------|---------------------------------|--|
| 25. FORDULÓ | | | |
| ESMTK IV. | 3:13 | Budai XI SE VIII.-Sto | |
| Budai XI SE XI. | 6:10 | Hercules SE II. - Topspin | |
| KIWI SE I. | 9:7 | Puskás DSK | |
| MAFC IV. | 9:7 | Városmajori Katolikus Egyesület | |
| FKF SK III. | 4:12 | EGIS SE II. | |
| Pestszentimrei PSE VI. | 9:7 | High Life SE II.-Diavoli | |
| Budai XI SE VII.-Kontra csoport | 8:8 | Kispest SE III. | |
| 26. FORDULÓ | | | |
| Hercules SE II. - Topspin | 6:10 | ESMTK IV. | |
| Puskás DSK | 10:6 | Budai XI SE VIII.-Sto | |
| Városmajori Katolikus Egyesület | 7:9 | Budai XI SE XI. | |
| EGIS SE II. | 6:10 | KIWI SE I. | |
| High Life SE II.-Diavoli | 8:8 | MAFC IV. | |
| Kispest SE III. | 13:3 | FKF SK III. | |
| Budai XI SE VII.-Kontra csoport | 12:4 | Pestszentimrei PSE VI. | |
| 0 - KILÉPÉS | | | |
| Válasszon: | | | |

Csapatmérkőzések:

Kilistázza a kiválasztott fordulóban lejátszott/kisorsolt mérkőzéseket. A mérkőzések előtti számokat beírva megtekinthetők az egyes csapatmérkőzések egyéni meccsei (ha már lejátszották).

| | | | |
|--------------------------------------------------------------------|----|-------------|--|
| 26. FORDULÓ MÉRKŐZÉSEI | | | |
| 1. HERCULES SE II. - TOPSPIN | -- | ESMTK IV. | |
| Nap / időpont: Hétfő / 18:00 | | | |
| Helyszín: IX. Koppány u. 2-4. I.em. (Soroksári úti TESCO, TOPSPIN) | | | |
| Játékvezető: Kiss Hanna | | | |
| Labda: JOOLA Flash | | | |
| Végeredmény: 6:10 | | | |
| 2. KISPEST SE III. | -- | FKF SK III. | |
| Nap / időpont: Hétfő / 18:30 | | | |
| Helyszín: XIX. Nádasdy u. 98. (Bejárat a Jókai utcából) | | | |
| Játékvezető: Kovács András | | | |
| Labda: JOOLA Flash | | | |
| Végeredmény: 13:3 | | | |

Egyéni mérkőzések:

A csapatmérkőzések egyéni meccseinek eredményei.

| | | | |
|-----------------------------------------------------------|---|----------------|----------|
| BUDAI XI SE VII.-KONTRA CSOPORT -- PESTSZENTIMREI PSE VI. | | | |
| Hazai játékos | - | Vendég játékos | Eredmény |
| Szűcs Gábor | - | Németh Gábor | 3:0 |
| Szűcs Gábor | - | Szabó Viktória | 1:3 |
| Szűcs Gábor | - | Török Gergő | 3:2 |
| Szűcs Gábor | - | Simon Viktória | 3:2 |
| Mészáros Andrea | - | Németh Gábor | 3:0 |
| Mészáros Andrea | - | Szabó Viktória | 1:3 |
| Mészáros Andrea | - | Török Gergő | 3:1 |
| Mészáros Andrea | - | Simon Viktória | 0:3 |
| Molnár Benedek | - | Németh Gábor | 3:0 |
| Molnár Benedek | - | Szabó Viktória | 1:3 |
| Molnár Benedek | - | Török Gergő | 3:0 |
| Molnár Benedek | - | Simon Viktória | 3:0 |
| Takács Árpád | - | Németh Gábor | 3:0 |
| Takács Árpád | - | Szabó Viktória | 3:0 |
| Takács Árpád | - | Török Gergő | 3:0 |
| Takács Árpád | - | Simon Viktória | 3:0 |
| Végeredmény: 12:4 | | | |
| 0 - VISSZA | | | |
| 1 - NYOMTATÁS FÁJLBA | | | |
| Válasszon: | | | |

2 – Bejelentkezés kapcsolattartóként

Előtte: **bejelentkezési adatok**

Bejelentkezési adatok: ld. [FONTOS! Utasítások](#)

INFORMÁCIÓK:

- A felhasználónév és a jelszó nem tartalmazhat ékezetes betűket.
- A felhasználónév és a jelszó csak betűket és számokat tartalmazhat.

0 - MÉGSEM

Kérem adja meg a felhasználónevét (min. 4 kar.):

2.1 - EREDMÉNYEK BEVITELE

2.2 - HAZAI PÁLYA MÓDOSÍTÁSA

2.3 - LABDA MÓDOSÍTÁSA

2.4 - TELEFONSZÁM MÓDOSÍTÁSA

2.5 - EMAILCÍM MÓDOSÍTÁSA

2.6 - JELSZÓ MÓDOSÍTÁSA

0 – KILÉPÉS

Üdvözljük !

1 - EREDMÉNYEK BEVITELE
2 - HAZAI PÁLYA MÓDOSÍTÁSA
3 - LABDA MÓDOSÍTÁSA
4 - TELEFONSZÁM MÓDOSÍTÁSA
5 - EMAILCÍM MÓDOSÍTÁSA
6 - JELSZÓ MÓDOSÍTÁSA
0 - KILÉPÉS
Válasszon:

3 – Bejelentkezés adminként

Előtte: **bejelentkezési adatok**

Bejelentkezési adatok: ld. [FONTOS! Utasítások](#)

3.1 - CSAPAT KIVONÁSA

3.2 - CSAPAT HOZZÁADÁSA

3.3 - JÁTÉKOS KIVONÁSA

3.4 - JÁTÉKOS HOZZÁADÁSA

3.5 - JÁTÉKVEZETŐ KIVONÁSA

3.6 - JÁTÉKVEZETŐ HOZZÁADÁSA

3.7 - EGY MECCSEREDMÉNY FELÜLÍRÁSA

3.8 - PONTLEVONÁS

3.9 - ÚJ SZEZON INDÍTÁSA

3.10 - JELSZÓ MÓDOSÍTÁSA

0 – KILÉPÉS

Üdvözljük!

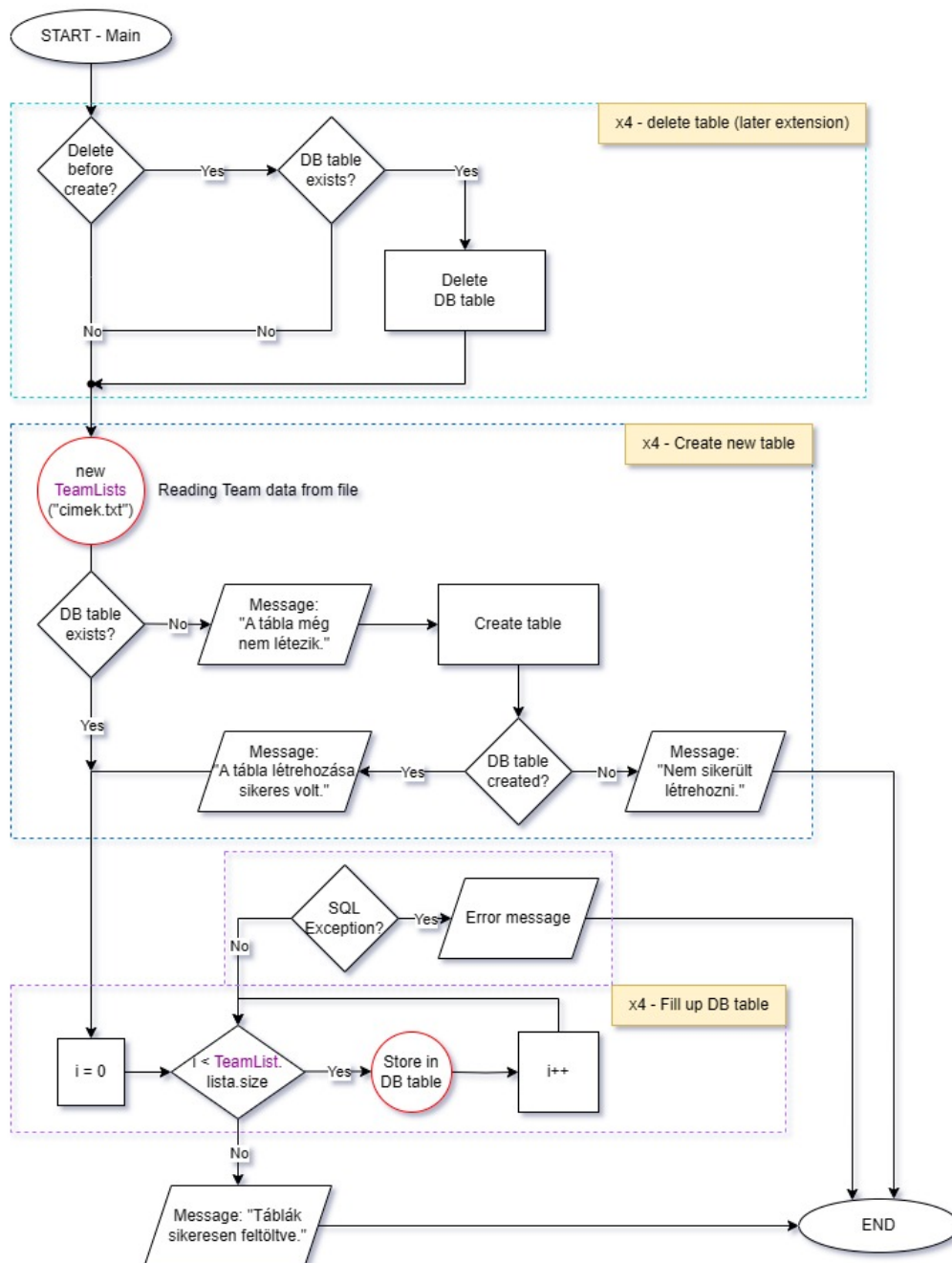
Ön rendszergazdaként lépett be!

1 - CSAPAT KIVONÁSA
2 - CSAPAT HOZZÁADÁSA
3 - JÁTÉKOS KIVONÁSA
4 - JÁTÉKOS HOZZÁADÁSA
5 - JÁTÉKVEZETŐ KIVONÁSA
6 - JÁTÉKVEZETŐ HOZZÁADÁSA
7 - EGY MECCSEREDMÉNY FELÜLÍRÁSA
8 - PONTLEVONÁS
9 - ÚJ SZEZON INDÍTÁSA
10 - JELSZÓ MÓDOSÍTÁSA
0 - KILÉPÉS
Válasszon:

Diagramok és ábrák

Folyamatábra

FillUpTeamsV2 - Teams, Locations, ContactDetails, Balls



Osztálydiagramok

Table_Tennis_Manager osztálydiagramjai

| Team | Team |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - teamID : Integer - gameDay : String - email : String - teamName : String - matchesPlayed : Integer - matchesWon : Integer - matchesDraw : Integer - matchesLost : Integer - points : Integer - location : String - gameDay : String - gameTime : String - ball : String - phoneNumber : String - contactPersonName : String - seasonID : Integer - contactPersonID : Integer - contactDetailsID : Integer - locationID : Integer - ballID : Integer - contactPerson : Player | <ul style="list-style-type: none"> + setContactPersonID(Integer): void + setTeamID(Integer): void + setContactPerson(Player): void + setSeasonID(Integer): void + setContactDetailsID(Integer): void + setLocationID(Integer): void + setBallID(Integer): void + getContactPerson(): Player + getTeamID(): Integer + getTeamName(): String + getMatchesPlayed(): Integer + getMatchesWon(): Integer + getMatchesDraw(): Integer + getMatchesLost(): Integer + getPoints(): Integer + getLocation(): String + getBall(): String + getEmail(): String + getPhoneNumber(): String + getContactPersonName(): String + getGameDay(): String + getGameTime(): String + getSeasonID(): Integer + getContactPersonID(): Integer + getContactDetailsID(): Integer + getLocationID(): Integer + getBallID(): Integer |
| <ul style="list-style-type: none"> + Team(teamID, teamName, matchesPlayed, matchesWon, matchesDraw, matchesLost, points) + Team(teamID, teamName, location, gameDay, gameTime, ball, email, phoneNumber, contactpersonName) + Team(teamName, location, gameDay, gameTime, ball, email, phoneNumber, seasonID) | |

| TeamMatch |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - teamMatchID : Integer - homeTeam : String - guestTeam : String - roundNumber : Integer - homeTeamID : Integer - guestTeamID : Integer - isOver : Boolean - result : String - umpire : String - gameDay : String - gameTime : String - location : String - ball : String |
| <ul style="list-style-type: none"> + TeamMatch(teamMatchID , homeTeam, guestTeam, roundNumber, isOver, result, umpire, gameDay, gameTime, location, ball) + TeamMatch(teamMatchID, homeTeamID, guestTeamID, homeTeam, guestTeam, result) + getTeamMatchID(): Integer + getHomeTeam(): String + getGuestTeam(): String + getRoundNumber(): Integer + getIsOver(): Boolean + getResult(): String + getUmpire(): String + getGameDay(): String + getGameTime(): String + getLocation(): String + getBall(): String + getHomeTeamID(): Integer + getGuestTeamID(): Integer |

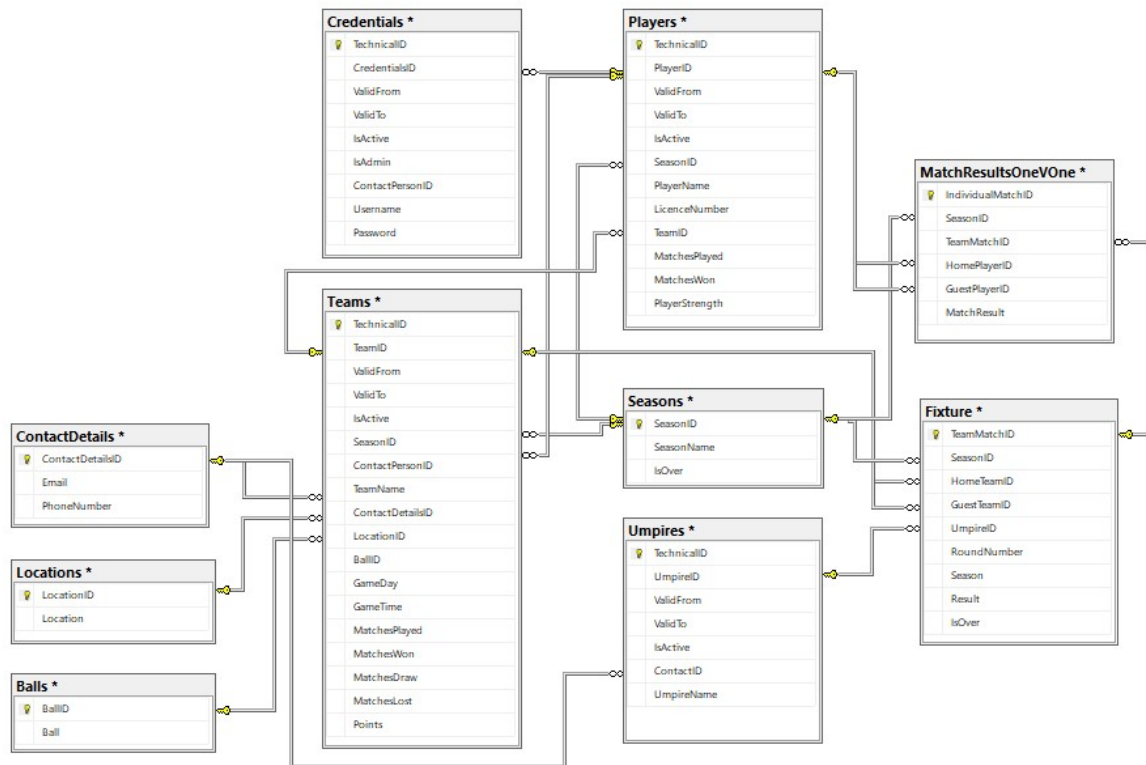
| Round |
|---------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - teamMatches: List<TeamMatch> |
| <ul style="list-style-type: none"> + Round(teamMatches) + getTeamMatches(): List<TeamMatch> |

| Player |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - playerId : Integer - playerName : String - licenceNumber : Integer - teamID : Integer - teamName : String - matchesPlayed : Integer - matchesWon : Integer - percentage : double |
| + Player(playerID, playerName, licenceNumber, teamID, teamName, matchesPlayed, matchesWon, percentage) + Player(playerName, licenceNumber) + getPlayerID(): Integer + getPlayerName(): String + setPlayerID(Integer): void + getPercentage(): double + getMatchesWon(): Integer + getMatchesPlayed(): Integer + getTeamName(): String + getTeamID(): Integer + getLicenceNumber(): Integer |

| IndividualMatch |
|----------------------------------------------------------------------------------------------------------------------------------------|
| - homePlayer : String - guestPlayer : String - result : String |
| + IndividualMatch(homePlayer, guestPlayer, result) + getHomePlayer(): String + getGuestPlayer(): String + getResult(): String |

| NameID |
|-----------------------------------------------------------------|
| - id : Integer - name : String |
| + NameID(id, name) + getId(): Integer + getName(): String |

Adatbázis táblái



Funkciók és megoldások

Kért tartalom

| | |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Öröklődés/leszármazás | Simulations > Logic classes CreateFixtureLogic => RoundRobinLogic => IndividualMatchesLogic => UpdateDBLogic A simulations program Logic osztályai a jobb átláthatóság érdekében egymás leszármazottjai a logikai sorrendet követve. |
| Implementálás | Simulations > MainRepository Megvalósítja az IMainRepository interfészt |
| Fájlból olvasás | pdf fájl -> txt fájl (cimek.txt) |
| Fájlba írás | table_tennis_manager > MainMenuUI > subMenu_FM11c_egyenimeccsek() > printTeamMatch_toFile() |
| Tárolt eljárás | FillUpTeamsV2 > fillUp_Locations() |
| Egyedi kivétel | WrongMenuItem: Table Tennis Manager > MainMenuUI > start_menu() > default |
| Felhasználóval kommunikáció | table_tennis_manager > UI |
| Unit tesztek | DBFiller = 3 Simulations = 3 |
| Kiíratós tesztek | Simulations > - DBwithSim_testingUI = 5 teszt: a szimulált statisztikák valósághűek? - RoundRobinTestingUI = a teljes sorolás kiíratása |
| Osztálydiagramok | A Table Tennis Manager osztályai |
| Folyamatábra | FillUpTeamsV2 – teams, locations, contactDetails, balls |

További tartalmak

| | |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Beágyazott lekérdezés | Table_tennis_manager > Logic > AdminMenuLogic > substract_pointsFromTeam() |
| Regex fájlolvasásánál | DBFiller > Factory > TeamLists konstruktor (cimek.txt) Megkeresi a csapatok kezdési időpontját. |
| Round Robin | Logic > RoundRobinLogic https://en.wikipedia.org/wiki/Round-robin_tournament |
| Játékosok erőssége | DBFiller > Logic > PlayerListClass > set_strengths() Kockadobás-szerűen kioszt egy random erősséget 1 és 6 között 1, 2, 5, 6 esetén „újradoz”, hogy a normál eloszlás felé konvergáljon. |
| Egyéni meccs szimuláció | A vendég VS. hazai játékos erőssége + „kockadobás” (0–5) Hatféle meccseredmény lehetséges: 3:0, 3:1, 3:2, 2:3, 1:3, 0:3 |
| Játékosok véletlenszerű kiosztása csapathoz | DBFiller > Logic > PlayerListClass > setTeamIDs() Legalább 4 minden csapatba kerül, azonfelül random |
| Testre szabott query | Simulations > Logic > RoundRobinLogic > get_umpireList() <pre>ArrayList<>(), From_table(SELECT:"UmpireName", FROM:"Umpires", WHERE:"1=1") etString(columnIndex:1));</pre> |

Összefoglalás

A programcsomag, amelyet a jelen dokumentációval egészítettem ki, jól szemlélteti az elmúlt tíz hónapban elsajátított ismereteket. A projektmunka három programrészére visszatekintve progresszív fejlődést látok, mely többek között például a statikus metódusokról objektumorientált osztályokra való folyamatos áttérésben is tetten érhető.

A projektmunka három külön részre bontása több koncepcióval való kísérletezésre is lehetőséget biztosított, ilyen volt például a különböző szintek (UI, Logic, Factory) megvalósításai. Az első programban (DBfiller) még látható, hogy a legalsó réteg két csomagban is megjelenik, a hierarchia még nem tisztult le – a Factory és a Repository rétegek egymással párhuzamosan léteznek, és a funkcióik sem letisztultak. Ez a pont a harmadik program (table_tennis_manager) megírására, úgy érzem, jól kikristályosodott. A gyakorlati programozás során fejtörést okozott még továbbá például az, hogy az egyes szinteket „fent” vagy „lent” példányosítsam-e inkább (pl. a Repositoryt a Logicban). Ez a harmadik program megírására, érzésem szerint, szintén erősebb tudássá szilárdult.

Találkoztam több olyan helyzettel is, ahol nincs feltétlenül jó megoldás, inkább helyzettől függőnek nevezhető. Ilyen volt például az a dilemma, hogy hol írjam meg az SQL-lekérdezéseket: Repository- vagy Logic-szinten. A három programban mindkettőre találni példákat. Szintén hasonló kategóriába esik a paraméterezés kérdése, amit a dokumentációban a [További tartalmak](#) utolsó pontjaként említek, vagy a metódusok szétbontása rövidebb, egyszerűbb funkciókra vagy meghagyása hosszú, komplexebb függvényekként.

Kevésbé mély megértésre jutottam azonban például az interfészek gyakorlati alkalmazása terén – noha elméleti hasznukat értem –, valamint a mélyre nyúló objektumok kiemelésénél (pl.: hogy a `roundList.getRound(1).getTeamMatchList.get(1).getPlayerList.get(1).getName` [fiktív, szemléltető objektum!]) esetén egy új, rövidebb Játékos objektum létrehozása, adott játékosal egyenlővé tétele, majd új érték hozzárendelése esetén a gyökérobjektum értéke mikor változik, és mikor nem). Ezek még inkább csak ösztönszerűen mennek, egyelőre nem látom át őket annyira, mint szeretném.