Záródolgozat

Lovász Ákos

2018

Vak Bottyán János Katolikus  
Műszaki és Közgazdasági Szakgimnázium,

Gimnázium és Kollégium

Darts számláló

# 

# Készítette: Lovász Ákos

54 213 05 Szoftverfejlesztő

2018

Konzulens: Sike Zoltán

Tartalomjegyzék

[1. Bevezetés 2](#_Toc512580699)

[3. Programterv: 5](#_Toc512580700)

[3.1 ÚjJáték 6](#_Toc512580701)

[3.2 Statisztika 6](#_Toc512580702)

[3.3 Kilépés 7](#_Toc512580703)

[3.4 Fő Ablak 7](#_Toc512580704)

[4. Adatbázisterv 8](#_Toc512580705)

[5. Tesztelési terv 9](#_Toc512580706)

[6. Program dokumentáció 9](#_Toc512580707)

[6.1 ConnectorDB 10](#_Toc512580708)

[6.2 NewGame 11](#_Toc512580709)

[6.4 Starter 13](#_Toc512580710)

[6.5 DTabla 13](#_Toc512580711)

[6.6 Játékmenet 14](#_Toc512580712)

[7. Felhasználói kézikönyv 16](#_Toc512580713)

[7.1 Felhasználói felület 16](#_Toc512580714)

[7.2 Játék lebonyolítása 19](#_Toc512580715)

[8. Darts Szabályzat 19](#_Toc512580716)

[9. Tovább fejlesztési lehetőségek 20](#_Toc512580717)

# Bevezetés

A darts program témát saját érdeklődési köröm és előzetes ismereteim alapján választottam, melyek segíthetnek a feladat megkezdésében, megoldásában. Dartsozni hobbi szinten, gyakran szoktam, így gyorsan sikerült témát választanom.

A cél egy szimpla darts játék és statisztika nyilván tartó program, mely a játék lebonyolítását, játékosok segítését, pontozását és nyilván tartását hajtja végre.

A program képes egy teljes darts játékot lebonyolítani a különböző kezdő értékek beállítását, kilépési módszert (dupla, szimpla kiszálló), játékosok pontjainak tárolását és globális statisztikát beleértve.

A játék folyamán a játékosok 3 dobásonként váltanak, melyet a program jelez és kilépési lehetőség esetén dobási ajánlatot tesz, mely a lehető legkevesebb dobással lépteti ki a játékost. A statisztikát egy külön ablakban meg lehet tekinteni, mely megjeleníti a játékos nevét és a győzelmei számát.

1. Alkalmazott technika

Programozási nyelvek közül a Java-t választottam, mivel tanulmányaim folyamán kiderült, hogy talán ez a nyelv áll legközelebb hozzám, ezt tudom megérteni a legjobban és sokkal nyíltabb rendszerben működik, mint például a C#.

A technika bemutatása:

A Java egyáltalános célú, objektumorientált programozási nyelv, melyet a Sun Microsystems fejlesztett a 2009-ig, majd a céget felvásárolta az Oracle. A Java alkalmazásokat jellemzően bájtkód (Java virtuális gép (JVM) által végrehajtható utasítások megjelenési formája) formátumra alakítják, de közvetlenül gépi kód is készíthető Java forráskódból. Létezik közvetlenül Java bájtkódot futtató hardver is, az úgynevezett Java processzor. Független az operációs rendszertől. A Java-ban írt programok a legtöbb hardveren közel azonos módon futnak. Ezt úgy érik el, hogy a Java fordítóprogram a forráskódot csak egy úgynevezett Java bájtkódra fordítja le. Ez azután a virtuális gépen fut, ami az illető hardver gépi kódjára fordítja.

Léteznek szabványos könyvtárcsomagok, melyek egységes funkcionalitásként teszik elérhetővé a hardver sajátságosságait (grafika, szálak és hálózat). A Java nyelv a szintaxisát főleg a C és a C++ nyelvektől örökölte, de sokkal egyszerűbb objektum modellel rendelkezik, mint a C++. Fejlesztői környezetnek az Eclipse Oxygen-t választottam, egy platformfüggetlen open-source (nyílt forráskódú) keretrendszer, mely biztosítja a Java fejlesztés alapjait és bővítményekkel növelhető funkcionalitása, mint pl: mysql connector (mysql adatbázissal összekapcsolás) vagy Java Swing (grafikus felülettervező).

A nyelv tulajdonsága az objektumorientáltság, programozási stílusra és a nyelv struktúrájára utal. Az OO fontos szempontja, hogy a szoftvert „dolgok” (objektumok) alapján csoportosítja, nem az elvégzett feladatok a fő szempont. Az objektumok (az adatokat tartalmazó entitások) jobb alapot biztosítanak egy szoftver-rendszer megtervezéséhez.

A cél az volt, hogy nagy fejlesztési projekteket könnyebben lehessen kezelni, így csökken az elhibázott projektek száma. A Java rendszer volt az első, amely lehetővé tette a távoli gépeken való futtatást sandboxban (homokozóban). Egy kisalkalmazás futtatható a felhasználó gépén.

A kód egy biztonságos környezetben fut, amely nem engedi meg "rossz szándékú" kód futtatását. A gyártók kiadhatnak olyan tanúsítványokat, amelyeket digitálisan aláírnak, ezzel a nevüket adva ahhoz, hogy a kisalkalmazás biztonságos. Így azok a felhasználó felügyelete alatt léphetnek ki a biztonságos környezetből.

A Java nem támogatja a goto utasítást, mivel ennek használata pongyola kódot eredményezhet. Nagyon ritkán mégis szükség van a goto-ra, a Java lehetővé tesz alternatív megoldásokat, ami a címkézhető continue és break utasítás. A goto fenntartott szó és nem használható azonosítóként.

A nyelv egyszerű adattípusai, más szóval primitív típusai a következők:

|  |  |  |
| --- | --- | --- |
| **Változó típusa** | **Leírás** | **Maximum értéke** |
| byte | 8 bites előjeles egész | byte = 127; |
| short | 16 bites előjeles egész | short = 32767; |
| int | 32 bites előjeles egész | int = 2147483647; |
| long | 64 bites előjeles egész | long = 9223372036854775807L; |
| float | 32 bites egyszeres lebegőpontosságú (IEEE 754 szabvány) | float = 3.4028235E38f; |
| double | 64 bites kétszeres lebegőpontosságú (IEEE 754 szabvány) | float = 1.7976931348623157E308; |
| char | 16 bites Unicode-karakter | char = 'S'; (egyetlen karakter) |
| boolean | logikai érték *(igaz* / *hamis)* | boolean = true; |

Az egyszerű tipusok helyes használata elősegíti a memória csökkentett felhasználását, például ha egy változóban nagyon kis értékek tárolódnak, érdemes byte-ban tárolni mely 127-ig tud tárolni, ezzel szemben az int lefoglal 32 bitet, attól függetlenül, hogy mekkora értéket tárol.

A Java nyelv négyféle láthatóságot támogat a típusok, adattagok és tagfüggvények körében:

* private - privát, azaz csak a definiáló osztály belsejében látható, más osztályok nem érik el.
* nincs kulcsszó - (angolul default vagy package-private) félnyilvános, azaz a definiáló csomag belsejében látható, kívülről nem látható és nem válzotatható.
* protected - védett, azaz a definiáló osztály leszármazottjaiból látható de hozzáférési engedély nélkül nem változtatható más osztályból.
* public - nyilvános, azaz mindenhol látható, elérhető és változtatható.

Egy azonosító láthatósága, vagy hatóköre a programnak az a része, ahonnan a jelölt programozási eszköz (változó, eljárás, függvény, objektum stb.) ezen az adott néven elérhető.

Egy azonosító lokális abban a blokkban, amelyben deklaráltuk, és globális minden olyan blokkban, amelyet ebből a blokkból hívunk meg.

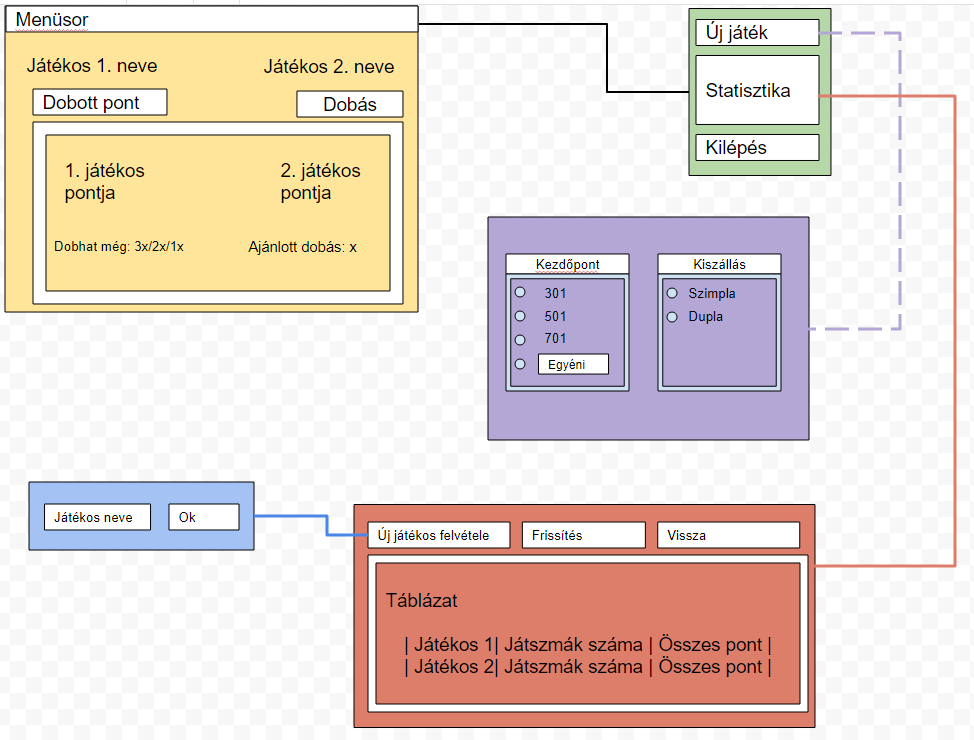
Egy adott blokkban csak arra a névre lehet hivatkozni, amely ott lokális vagy globális. Azaz vagy abban az alprogramban van deklarálva, ahol hivatkozunk rá, vagy egy olyan alprogramban amely meghívta, illetve benne van a hívási sorozatában.

A főprogramban deklarált valamennyi programelem (változó, eljárás stb.) hatásköre a teljes program, hiszen a saját blokkjában lokális, míg az összes többiben globális.

Ha egy alprogramban deklarálunk egy azonosítót, akkor az felüldefiniálja a korábban deklarált vele azonos globális nevet.

A változó élettartama a program végrehajtási idejének az a része, ameddig az adott memóriaterületet arra használhatjuk, hogy az az adott változó értékét hordozza. Ez a lefoglalt memóriaterület változótól függő méretű.

# Programterv



1. ábra Program grafikus terve

Az ábra egy korai tervezetet mutat, a végső program felépítése mind elrendezésben, mind funkcionalitásban kis mértékben eltér.

## 3.1 ÚjJáték

Az “új játék” menüpont a játékszabályok beállítására szolgál. Ez megnyit egy új ablakot melyben ki lehet választani a mérkőzés típusát, kezdőpont értékét három előre megadott értékből, vagy egyéni érték megadásával. Szintén ezen a felületen lehet kiválasztani a résztvevő játékosok nevét.

A kezdőpontot és a kiszállás típusát előre kijelölt rádiógombokkal lehet kiválasztani, mely biztosítja a hibák elkerülését pédául szöveges mezők használatával szemben, így kevesebb probléma merülhet fel a program futásában.

A rádiógombokat két csoporthoz rendeltem, mivel a rádiógombok checkbox-on alapulnak, azzal a tulajdonsággal, hogy a csoportba foglalt opciók közül egyszerre csak egyet lehet kiválasztani.

A játékosok nevét egy legördülő menüből lehet kiválasztani, amit a játékosok tárolására szolgáló adatbázis alapján van feltöltve

## 3.2 Statisztika

A “Statisztika” menüpont előhoz egy új ablakot, mely táblázatban prezentálja a regisztrált játékosokat, mely adatokat egy adatbázisból kér le, és a következő funkciókat látja el:

* Új Játékos: új játékos regisztrálása, feltölti egy adatbázisba a nevet és hozzárendel 0 győzelmet, melyet játék folyamán lehet növelni
* Frissítés: az ablakban látható táblázatban szereplő adatok feltöltésére és frissítésére szolgál, az adatbázisban tárolt adatok alajpján tölti fel a táblát.
* Vissza: Bezárja a statisztika felületet és visszalép a játékfelületre.

Az összes játszott meccs növeléséhez elég belépni egy játszmába, a győzelmeket egyértelműen nyertes meccsekkel lehet növelni.

A táblázat megjelenítését és feltöltését a játékosok tárolására szolgáló adatbázis tábla segítségével kell feltölteni. Ezt a feltöltést vagy a TableModel-be kell beleépíteni, vagy vektort alkalmazni. Én a TableModel eljárást alkalmaztam.

Tábla feltöltésére szolgáló kódrészlet:

*for (int i = sorszam-1; i >= 0; i--) {*

*((DefaultTableModel)table.getModel()).removeRow(i);}*

*while (rs.next()) { ((DefaultTableModel)table.getModel()).addRow(new Object[] { rs.getString(2), rs.getString(3)});}*

A táblát egy ResultSet lekérdezés alapján egyszerre formázza és tölti fel adatokkal, amit az SQL táblából eredményként kapott a ResultSet lekérdezésből.

## 3.3 Kilépés

Bezárja a programot, egy megerősítő üzenet után. A megjelenő ablakhoz egy egyedi Jframe ablakot hoztam létre előre megadott választási lehetőseégekkel:

JFrame frame = new JFrame("Input"); Object[] options = {"Igen", "Nem"};

int ki = JOptionPane.showOptionDialog(frame,"Biztosan ki akarsz lépni?","Kilépés",

JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE, null,  options, options[0]); if (ki == 0) { dispose(); } else;

Ez a kódrészlet írja le a programból való kilépés folyamatát, mely egy tömbben tárolja a kilépési lehetőségeket („igen” „nem”), egy új Jframe ablakot hoz létre a kilépés gomb megnyomására, a „ki” változó egyben meghívja ezt az új ablakot, melynek paraméterként a „szülő” ablakot, a címét, a feliratot, az ablak típiusát és végül a sömb elemeit kell megadni, amiben tárolva vannak az opciók majd bezárásakor eltárolja a kapott választ. A gombok automatikusan a tömb elemeihez hűen indexelt értéket adnak vissza, így a változó vagy 0-t, vagy 1-et kaphat vissza. Mivel a tömb nullától indexelt és az „igen” opció az első a sorban, az „igen” a 0 index alatt áll. Ezt követően, ha a visszatért érték 0, a program bezárja önmagát mivel a felhasználó az „igen” opciót választotta a „Biztosan ki akarsz lépni?" kérdésre.

## 3.4 Fő Ablak

Megjeleníti a két játékos nevét, mindkét játékos pontszámát a nevük alatt,aktív játékos számára hátralévő dobások számát, és a következő ajánlott dobást. (“nincs ajánlat” esetén egyértelműen bármit dobhat a felhasználó).

A dobott pontot egy szöveges mezőben kéri be a program és egy “dobás” gombbal erősíti meg a dobott pontját a játékos.

A gomb megnyomására a program az éppen aktív játékos pontszámából kivonja a dobott pontot, ellenőrzi az eredményt és ha kilépésre van lehetőség, felajánl egy optimális stratégiát arra. Ezen felül ellenőrzi, hogy a dobás nem-e túl nagy, ha többet dob, mint pontszáma, visszaugrik az utolsó pozitív álláspontra és onnan folytathatja a játékos, ha van még dobása.

Minden körben 3 dobás áll az aktív játékos számára, ha ezt felhasználta, a körnek vége és a következő játékos kezdheti a dobást, amit egy üzenet ablak is közöl a játékosokkal. A nulla pont elérésekor a program ellenőrzi a kilépés típusát, hogy a beállításnak megfelelő-e, ha igen, egy ablak jelenik meg ami közli a nyertes nevét. Ha nem érvényes a kilépés, ismét az utolsó álláspontra lép vissza a játékos pontszáma és újra kell próbálnia.

A győzelmek száma névhez van rendelve, így sajnos azonos névből nem lehet több. Név megadásakor a játék elején egy lekérdezés indul az adatbázisban, ellenőrzi, hogy létezik-e már ilyen játékos. Ha létezik, annak a játékosnak a statisztikája lesz frissítve, ha nem, a statisztika nem tárolódik.

# Adatbázisterv

Az adatbázis kezelésére a MySQL nevű többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szervert választottam, mivel a Java programokat nagyon egyszerű módon lehet összekapcsolni az adatbázisokkal és kezelni azokat a programon belül.

A szoftver eredeti fejlesztője a svéd MySQL AB cég kettős licenceléssel tette elérhetővé a MySQL-t. A GPL szabad szoftver licenc, vagy egy zárt licenc érvényes a felhasználásra. 2008 januárjában a Sun felvásárolta 800 millió dollárért a céget, majd a Sunt felvásárolta az Oracle, így a MySQL is az Oracle tulajdonába került.

A MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.

Egyedi illesztőfelületekkel az adatbázis-kezelő elérhető C, C++, C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. Egy MyODBC nevű ODBC interfész további, ODBC-t kezelő nyelvek számára is hozzáférhetővé teszi az adatbázis-kezelőt. A MySQL számára az ANSI C a natív nyelv.

A programomhoz 4 táblát használtam. Egy-egy kiszálló segéd-táblát a szimpla és a dupla kiszállók ajánlásához, melyet egy előre kiszámolt tábla alapján állítottam össze.

Ezek a táblák a játék beállításaiban meghatározott kilépési mód alapján látják el a programot a játék lebonyolítása alatt optimális dobás sorrenddel, a játékosokat biztosítja egy hatékony dobási sorral. Ezt a segítséget a játékos figyelmen kívül veheti, ha úgy látja jónak, ennek nincs valós befolyása a játékra.

Egy segédtábla, ami tárolja a játék beállításait, egyszerű tábla, csak 4 mezőt tartalmaz egy rekorddal, csak a játék kiinduló pontját tartalmazza, a kilépés típusát és a két aktív játékos nevét, amit beállítás után egy lekérdezéssel ér el a program és egy tábla mely tartalmazza a regisztrált játékosok nevét és győzelmeinek számát.

# Tesztelési terv

A hibakezelés ellenőrzését és javítását először elvárt értékek megadásával kezdtem, hogy megbizonyosodjak a program alapszintű működéséről.

Miután a helyes értékek megadása elvárt eredményt produkált, runtinvizsgálatokba kezdtem. Elsőnek a negatív számok jutottak eszembe, mely megadására pontszáma növekedett a játékosnak.

Mivel ilyen egy darts játékban nem történhet meg, minden pontlevonás előtt ellenőrizni kezdtem a dobás értékét. Ha negatív számot ad meg a felhasználó, a program nem változtatja a pontszámát az aktív játékosnak és nem von le dobást a köréből, de figyelmezteti, hogy ilyen érték dobása nem lehetséges a darts-ban.

Másodikként azt ellenőriztem, hogy szöveget se lehessen megadni a szöveges mezőben, ezt nagyon egyszerű volt kezelni.

További tesztelések folyamán kiderült, hogy ha a felhasználó túl nagyot dob, a pontszám nulla érték alá esik, de nem kapja meg a győzelmet.

Ezt úgy oldottam meg, hogy egy külön változóban tárolom az utolsó dobást, és ha nulla alá esik a pontszám, de nem érvényes a kilépés, az utolsó dobott értéket hozzáadja a negatív pontértékhez.

Ez újabb problémát hozott létre, hogy ha a túl nagyot dob a játékos, a pontszámához hozzáadódik az utolsó dobott érték, így akár többszöröse is lehet az utolsó álláspontnak. Ennek hatására vezettem be a 60-as dobási határt, mivel 3 dobásos körök vannak és a dartsban egy dobással maximum 60-at lehet dobni.

Ez azt eredményezi, hogy a felhasználó nem tud akkorát dobni, hogy a pontszáma túllépje az utolsó álláspontját, mielőtt hibás értéket adott meg.

A tört értékek dobásának lehetősége is felmerült, de a változók típusából eredendően ez nem jelent problémát mivel integerben tárolom a változók nagy részét, mely nem képes tört értékek kezelésére, így egy egyszerű hibakezelés megelőzi a problémát.

A hasonló hibákhoz vezető értékeket minden felhasználó által manipulálható mezőben teszteltem, majd ezeket a hibákat kezeltem.

# Program dokumentáció

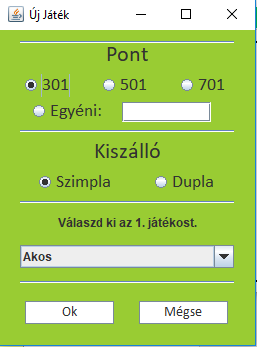
* Osztályok:
  + ConnectorDB
  + DTabla
  + NewGame
  + Starter
  + Table

## **6.1 ConnectorDB**

Egy publikus osztály, ami a program során többször is meg van hívva, ezen az osztályon keresztül éri el a program a mySQL felületet és kommunikál az adatbázissal. JDBC Driver for MySQL -t használ, ami a MySQL által fejlesztett hivatalos Java kapcsoló driver. a Class.forName("com.mysql.jdbc.Driver"); és con=DriverManager.getConnection() parancsokkal kapcsolódik az adatbázishoz a megadott elérési úton a következő módon: "jdbc:mysql://localhost/*táblanév*", "*felhasználó név*", "*jelszó*"

a public Connection getCon() { return con; } parancs adja meg a külső meghívás lehetőségét.

## **6.2 NewGame**



2. ábra Új játék grafikus felülete

A játékmenet beállításait tartalmazza. Ez egy grafikus felület, 6db rádió gombbal, 2 csoportba osztva és egy legördülő menüvel, mely tartalmazza az adatbázisban tárolt neveket.

A rádiógombokat két csoporthoz kellett rendelni, mivel a rádiógombok checkbox-on alapulnak, azzal a tulajdonsággal, hogy a csoportba foglalt opciók közül egyszerre csak egyet lehet kiválasztani.

Az első csoport 4 rádiógomb, ezekkel lehet kiválasztani a kiindulási pontot a játékhoz. Három gyakori kezdő pontot ajánl a darts szabályai alapján(301,501,701), a negyedik pedig egyéni értéket kér egy szöveges mezőben.

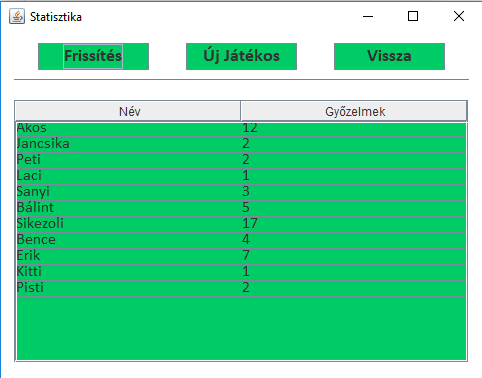
A második csoportban a kilépést lehet kiválasztani, szimpla vagy dupla kiszálló opciókból.

Az OK gomb megnyomására a program kikapcsolja a többi választási lehetőséget, kéri a második játékos kiválasztását, majd meghívja a ConncetorDB-t és feltölti a választott elemeket egy segéd-táblába egy preparedStatement segítségével ezt a frissítő SQL kódot futtatja le az adatbázison:

*UPDATE helper SET helper.kezd = (?), helper.kilep = ? WHERE helper.id = 1. ("UPDATE `helper` SET `player"+x+"`='"+pN+"'");*

Ez a parancs frissíti az értékeket a segédtáblában, a PreparedStatement utasításon keresztül a kérdőjelek lecserélődnek megadott válzotók értékeire, mely a kérdőjelek indexelésével és a változó megadásával lehetséges. Ebben az esetben az általános programozással szemben az indexelés nem nullától keződik, valós elhelyezkedést kell megadni indexként, sokkal inkább sorszámról van szó (pl az első kérdőjel 1-es indexű).

**6.3 Table**

****

**3**. ábra Statisztika és regisztrációs felület

A statisztika megjelenítésére és új játékosok felvételére képes ablak. Három gomb helyezkedik el rajta és egy táblázat amit az adatbázisból lekért adatokkal tölt fel.

Gombok:

* Frissítés
* Új játékos
* Vissza

A frissítés az ablakban látható táblázatban szereplő adatok feltöltésére és frissítésére szolgál, az adatbázisban tárolt adatok alajpján tölti fel a táblát.

Az új játékos egy szöveges mezőben bekéri a nevet, majd feltölti azt az adatbázisba, egyértelműen 0 győzelemmel. A vissza gomb bezárja ezt az ablakot és visszatér a fő ablakhoz.

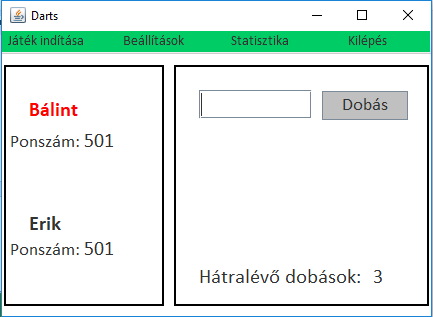
Az új játékosok feltöltését egy preparedStatement segítségével hajtom végre, ami egy előre megírt SQL parancs kérdőjelekkel helyettesített értékeit a programon belüli változókkal képes helyettesíteni, majd az értékmegadást követően az execute paranccsal a kérdőjeleket helyettesíti a megfelelő típusú és értékű változóval amit az SQL is tud értelmezni, így a Java és az SQL közötti értékátadás zökkenésmentesen valósulhat meg *("INSERT INTO `jatekosok` VALUES (?,?,?)")*.

Ez feltölti a megadott játékos nevét. Ha az új játékos felvételéhez szükséges szöveges dobozt nem tölti ki a felhasználó, egy üzenetet kap, hogy üresen hagyta és újra kéri a nevet. A táblázatot egy SELECT \* FROM `jatekosok` SQL paranccsal kérem le és ez alapján tölti fel a táblázatot az ablakban egy table.getModel() parancson keresztül.

## **6.4 Starter**

Egyetlen funkciója a program megkezdése a fő ablak meghívásával: new DTabla(); , mivel a DTabla osztály nem taralmaz Main függvényt.

## **6.5 DTabla**



4. ábra Fő kezelő feület

A program fő felülete. Itt játszódik le a darts és innen lehet megnyitni a többi ablakot. A felső sávban található a menüsor.

Itt van lehetőség a játék elindítására, ami kiírja a fő ablakon a játékosok nevét, pontjaikat, hátralévő dobásaik számát és a következő ajánlott dobást.

A második opció a beállítások ablakot nyitja meg, amiben be lehet állítani a kezdő értéket, a kilépés módját és a két játékos nevét.

A statisztika menüpont megnyitja az ablakot melyben meg lehet tekinteni a különböző játékosok nevét és győzelmeinek számát.

A kilépés menüpont egy megerősítést kérő ablakot dob fel, mely ha igen választ kap a “Biztos ki akar lépni?” kérdésre, bezárja a programot. A felület két részre van osztva: a felső része, a játékterület, a játékosok nevét és a dobási interfészt tartalmazza, az alsó pedig az információs sávot.

Az információs sáv tartalmazza a játékosok pontjait, a hátralévő dobások számát és az optimális kiszálláshoz ajánlott dobásokat.

## 6.6 Játékmenet

A Dobás gomb megnyomására a program feltölti a változókat a megfelelő értékekkel, mint például frissíti a játékosok pontját, dobások számát, kezdőértéket és kilépési módot lekérdi az adatbázisból és elhelyezi változókban, egy boolean válzotó tárolja, hogy melyik játékos aktív jelenleg és egyéb segédváltozók értékeit tölti fel.

Dobás esetén először a program piros színűre állítja az aktív játékos nevét, majd ellenőrzi, hogy dobott-e valamit a játékos, vagy üresen hagyta a dobási mezőt. Ha üresen hagyta, felkéri a játékost hogy adjon meg egy értéket.

Ha adott értéket a játékos, a dobások számát nyilvántartó változó 3-ra válzoik és egy swich-case elágazásba lép be.

Ez az elágazás a hátralévő dobások számát ellenőrzi, ez alapján dönti el a program, hogy hol tartanak a játékosok, mikor kell játékost váltani és egyszerűen lehet redukálni a játékos pontjából a dobott értéket.

Minden dobás esetén a program ellenőrzi a kilépés típusát, a játékos pontszámát, és annak dobás utáni értékét. Ezen felül minden dobás értékét is ellenőrzi, hogy túllépte e a hatvanat, mivel hatvannál nagyobbat nem lehet dobni dartsban.

Minden pont levonásakor ellenőrzi, hogy a pontszám elérte e a nullát, vagy haladhat tovább a játék. Emellett kezeli, ha a dobás nulla érték alá viszi a pontszámot, ilyenkor közli a játékossal, hogy túl nagy értéket adott meg, ezután egy segéd változóban tárolt értéket hozzáad a ponthoz, ez a változó azt tárolja, hogy utoljára mit dobott a játékos.

Ha a pont levonását követően a játékos nem érte el a nulla értéket, dobásainak száma csökken eggyel. Ez akkor is bekövetkezik, ha dobása túl nagy, a darts szabályaihoz hűen. Ha a pontszám eléri a nullát, a program ellenőrzi, hogy az utoljára dobott érték érvényes e a kiválasztott kilépési módnak. Ha érvényes, a program kiírja a győztes játékos nevét, és gratulál annak.

Ekkor az adatbázisban tárolt győzelmek számát megnöveli egyel, egy UPDATE query-vel. Ha a kiszállás nem érvényes, értesíti a játékost a program, majd a túl nagy dobáshoz hasonlóan az utolsó érvényes álláspontra pörgeti vissza a játékos pontszámát. Ezen felül minden dobás esetén ellenőrzi a program, hogy van e lehetőség optimális kiszállásra, amit a kiszállási módnak megfelelő táblázatból kérdez le.

Ha talál lehetőséget, azt lekérdezi az adatbázisból, majd megjeleníti a játékos számára az ablak jobb alsó sarkában. Ha nincs kiszállásra lehetőség, a program ezt a mezőt egy “Nincs ajánlatom” szöveggel tölti fel. A dupla kiszállás ellenőrzését minden dobás után több szintű ellenőrzés erősíti meg, mivel csak páros értékű dobással lehet kilépni, de nem minden páros dobás kilépés, ezért nem elég egy páros szám ellenőrzés minden dobás folyamán. A feltétel a következő dolgokat ellenőrzi:

*(if((Integer.parseInt(txtScored.getText()) % 2 == 0) && (p1s == 0) && (decider.equals("Dupla"))) )*

Először megerősíti, hogy a dobott érték páros, majd azt hogy a pont ténylegesen elérte e a nulla értéket, majd azt, hogy a kiszállási mód dupla kiszállás e.

Ha mind a három feltétel teljesül, csak akkor kezdi az adatbázisban az adatok frissítését és értesíteni a játékost a győzelemről. Ha csak a “dobás páros” feltétel nem teljesül, a program értesíti a játékost, hogy érvénytelen a kilépés, majd visszatér az utolsó érvényes álláspontra.

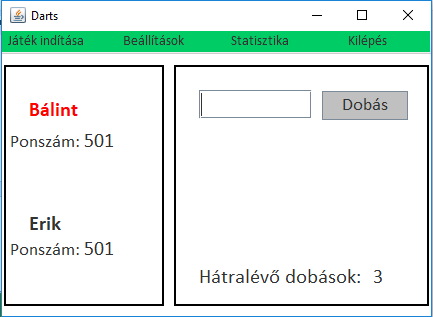
A szimpla kiszálló ennél egyszerűbb módon, csak a kiszállás típusát és a pontszámot ellenőrzi. A játék végén megjelenik egy ablak, mely közli a győztes nevét és a háttérben az adatbázisban frissül a győzelmek értéke, ha a játékos megtalálható az adatbázisban.

Ha nincs előre regisztrálva a felhasználó a statisztika ablakról, az adatai nem kerülnek tárolásra. Végül a biztonság kedvéért minden ablak kódja legvégén elhelyezkedik egy repaint(); és egy validate(); parancs ami arról győződik meg, hogy minden eleme megjelenjen a grafikus felületnek.

# Felhasználói kézikönyv

## 7.1 Felhasználói felület

A program indítása során megjelenik a fő felület. Ebben az ablakban történik a játék lebonyolítása, és más menük megnyitása is innen lehetséges.

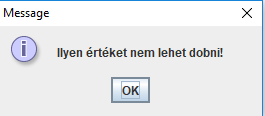


5. ábra Játékfelület

A felső sorban látható a menüsor. Ebben a sorban több opció is megtalálható, melyek a játékkal kapcsolatosak.

Az adatokat az indítás előtt a Beállítások menüpontban kell beállítani. A Játék indítása menüpont kiválasztását követően a játék beállításának megfelelően megjelennek a pontok, a nevek és kezdődhet a játék.

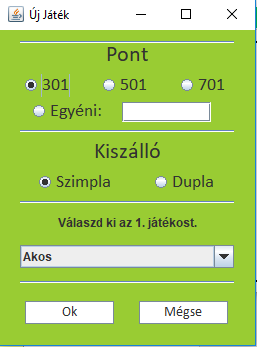
Az aktív játékos neve piros, míg az inaktív játékosé fekete. A szöveges mezőben a dobott értéket kell megadni, ami csakis szám lehet. Ha nem számot ad meg a játékos, a program figyelmezteti a játékost, majd újra kéri dobását.



6. ábra Hibakezelés

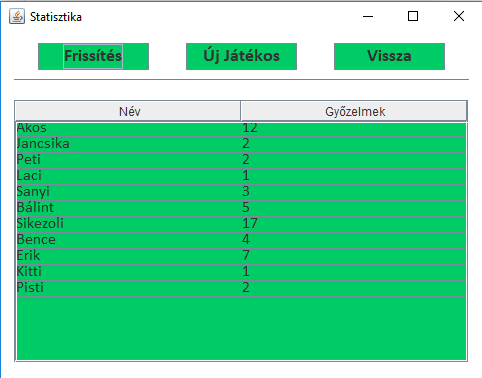
A játékosok jelenlegi pontját a nevük alatti számok jelzik, jobb alul a hátralévő dobások száma és a beviteli mező alatt jelenik meg az ajánlott dobás, ha van erre lehetőség. A második menüpont a Beállítások.

Ezt megnyitva a felhasználó kiválaszthatja a következő játék szabályait, mint a kilépés típusát és kezdőpont értékét és a résztvevő játékosokat. A kezdő érték lehet egyéni is, egyértelműen csak szám érték esetén. Ha a felhasználó nem számot ad meg, a program nem tud továbblépni, mivel szövegből nem lehet értékeket kivonni.



7. ábra Új játék beállításai

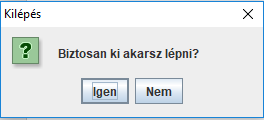
A harmadik menüpont a Statisztika, mely szintén egy ablakot nyit meg, melyben át lehet tekinteni a regisztrált játékosok nevét és győzelmeinek számát.



8. ábra Statisztika és regisztrációs felület

Az ablak tartalmaz három gombot és egy beépített táblázatot. A frissítés gomb a táblázat állapotát frissíti esetleges hiba esetén, vagy ha nem minden játékos jelenik meg, érdemes a frissítés opciót megpróbálni. Az új játékos gomb egy szöveges dobozt hoz elő, mely bekéri a játékos nevét.

Ez a regisztrációs felület. A név bekerül egy adatbázisba és onnantól követve lesz a statisztikája. FIGYELEM! Ha a játék kezdetekor megadott név egy nem regisztrált név, az adatai nem kerülnek mentésre. A vissza gomb bezárja az ablakot és visszatér a játék felületéhez. A menüsor utolsó opciója a kilépés mely egyértelműen bezárja a programot



9. ábra Kilépés kezelés

A bezárás előtt a program megkérdezi a felhasználót, hogy biztos e a döntésében.

## 7.2 Játék lebonyolítása

Új játékost a Statisztika ablakot megnyitva, majd új játékos menüpontot választva lehet regisztrálni, mivel ha a játékos neve nincs a mérkőzés előtt regisztrálva, a legördülő menüben nem lesz elérhető. Második lépésként ajánlott a Beállítások menüpontot megnyitni, mivel ebben az ablakban lehet beállítani a kiindulási pontot és a kilépés típusát.

A Pont felirat alatt található értékek a kiindulási pontot jelzik. Itt lehetőség van egyéni pont megadására is. A legalsó opció a játékos kiválasztása. A fölötte elhelyezkedő szöveg jelzi, hogy melyik játékost kell kiválasztani. Ha mind a két játékos ki lett választva, a program tovább tud lépni és ezt követően a játékfelületen a Játék indítása menüpont kiválasztásával indul a játék.

Az aktív játékos neve pirossal van jelölve, míg a várakozó fél neve fekete. A felület jobb alsó területén található a hátralévő dobások száma, mely jelöli, hogy az aktív játékos hányszor dobhat még, mielőtt véget ér az ő fordulója. A jobb oldalon található szöveges mezőben kell megadni a dobott értéket, mely csakis szám lehet.

Az érték bevitelét követően meg kell nyomni a dobás gombot, mely végrehajtja a funkcióit. A dobás értéke nem lehet tört, negatív, vagy egyetlen dobás nem lehet hatvannál magasabb értékű, a darts szabályaiból eredendően.

Ha egy játékos pontszáma eléri a nullát és a kilépése érvényes, azaz duplát dobott, ha duplakiszállót állított, vagy szimpla kiszálló esetén egyszerűen elérte a nullát, a program gratulál a győztesnek és a győzelmét eltáraolja, amit meg lehet tekinteni a Statisztika ablakban.

# Darts Szabályzat

A darts angol eredetű játék és sport, melynek során apró nyilakkal dobnak egy kör alakú céltábla különböző pontértékű szektoraira.

A játék célja minél nagyobb összegű pontot elérni (maximálisan 180). A játékosok felváltva dobnak körönként 3-3 nyilat. A dobás értékét a nyíl által eltalált szektor száma adja meg.

A dartsban legeket, és szetteket kell nyerni. A legtöbb versenyen egy nyert leg automatikusan egy megnyert szettnek számít, a világbajnokságokon viszont három leg után nyer a versenyző egy szettet. Az előre meghatározott számú szettet nyerő játékos nyeri a mérkőzést.

Általában pubokban és kocsmákban játsszák a világ minden részén, de a legnagyobb versenyek és szervezetek az Egyesült Királyságban, az Amerikai Egyesült Államokban, Hollandiában és a skandináv országokban működnek. A sport legfőbb irányító szervezete a World Darts Federation (WDF), a legprofibb játékosok azonban a Professional Darts Corporation versenyein vesznek részt, így a világbajnokságon is. Bár a játékot már a 19. században is játszották, az angol parlament csak 2001-ben nyilvánította hivatalosan sporttá.

A dartsban két szakágat különböztetnek meg, a tábla és a nyilak anyagától függően. Steel dartsnak nevezik azt a játékot, amikor fémhegyű (steel=acél) nyíllal szizál (vagy ritkábban egyéb anyagú) táblára dobnak, soft dartsnak pedig azt, amikor a játékosok automata gépekre játszanak műanyag hegyű nyilakkal. Mivel a nemzetközi versenyeken és a Vb-n is fémhegyű nyilakkal játszanak, a steel darts-ot egyszerűen a darts szóval azonosítják.

A legenda szerint néhány száz évvel ezelőtt egy átfázott angol íjász elhatározta, hogy kellemesebb módot választ a gyakorlásra a téli hónapokban. Letette íját, levágott a nyílvesszőjéből és elvonult a közeli pub kellemes melegébe és kényelmébe, ahol ügyességét úgy gyakorolta, hogy a lerövidített nyílvesszőket a falra akasztott farönkszeletbe dobta. Mellé állt egy másik íjász, aki megpróbált az előzőnél jobb eredményt elérni és ezzel megszületett a vetélkedés, a játék, a sport.

# Tovább fejlesztési lehetőségek

A legegyértelműbb tobábbfejlesztési lehetősége a programnak egy grafikus dobási felület létrehozása, mely megkönnyítené a dobás folyamatát. Ennek az implementációja viszont a jelenlegi változatban nem biztos, hogy előnyére szolgálna, mivel eredetét tekintve a program nem egy játékszoftver, hanem sokkal inkább egy játék mellett használt segédprogram, mely átvezet a második lépésére a továbbfejlesztésnek, a program integrációja egy digitális dartstáblába. A digitális dartstáblák ugyanis hasonló szoftvereket haszálnak, így ebben a környezetben több értelmet nyerne a dobásszámlálás, a kilépés ajánlás és a dobásonkénti pontmegadás.

Edzői vagy effektívebb segítői programként jobban megállná a helyét ha sokkal több statisztikát követne, például átlagos dobások értéke, átlagos dobások meccsenként, győzelmek aránya, összes eldobott dart és hasonló statisztikák, plusz gyorsabb használat érdekében a dobási határt 60-ról 180-ra emelve és a három dobásos rendszer kiiktatása meglehetősen felgyorsítaná a folyamatot. Egyéb tobábbfejlesztéserk közé tartozhat egy számítógép ellenfél, mely véletlenszerű értékeket dobna, mely segítené a játékost a gyakorlásban.

[1. ábra Program grafikus terve 7](#_Toc512581765)

[2. ábra Új játék grafikus felülete 13](#_Toc512581766)

[3. ábra Statisztika és regisztrációs felület 14](#_Toc512581767)

[4. ábra Fő kezelő feület 15](#_Toc512581768)

[5. ábra Játékfelület 18](#_Toc512581769)

[6. ábra Hibakezelés 18](#_Toc512581770)

[7. ábra Új játék beállításai 19](#_Toc512581771)

[8. ábra Statisztika és regisztrációs felület 20](#_Toc512581772)

[9. ábra Kilépés kezelés 20](#_Toc512581773)

Források:

Java története és jellemzői: <https://hu.wikipedia.org/wiki/Java_(programoz%C3%A1si_nyelv)>

<https://hu.wikipedia.org/wiki/Java_Development_Kit>

MySQL: <https://hu.wikipedia.org/wiki/MySQL>

<http://www.oracle.com/technetwork/database/mysql/index.html>

Darts szabályzat: <https://hu.wikipedia.org/wiki/Darts>

<http://www.magyardarts.hu/letoltesek/>

Változók hatásköre és mérete: <http://tudasbazis.sulinet.hu/>