**Java Alkalmazások Előadás Beadandó Dolgozat**

Tartalomjegyzék

[1. Feladat Pontos leírása 5](#_Toc120828570)

[2. A gépi és nyelvi igények 7](#_Toc120828571)

[2.1. Ajánlott specifikációk: 7](#_Toc120828572)

[3. A mappa és fájlszerkezet bemutatását név és funkció szerint 8](#_Toc120828573)

[3.1. Github repository tartalma: 8](#_Toc120828574)

[3.2. A Java osztályok: 9-11](#_Toc120828575)

[4. Képernyőképek 12](#_Toc120828576)

[4.1. Alaplap 12](#_Toc120828577)

[4.2. Olvasas1 13](#_Toc120828578)

[4.3. Olvasas2 14](#_Toc120828579)

[4.4. Írás 15](#_Toc120828580)

[4.5. Módosítás 16](#_Toc120828581)

[4.6. Törlés 17](#_Toc120828582)

1. Feladat Pontos leírása
2. **Adatbázis menü**: egy CRUD alkalmazás a választott adatbázissal **(5 pont)**  
   **Olvas almenü**: Megjeleníti az adatbázis adatait egy táblázatban. Az adatbázis legalább 3 táblájának adatait használja fel. Ha túl sok adatot tartalmaz az adatbázis, akkor elég csak egy részét megjeleníteni.  
   **Olvas2 almenü**: Hasonló mint az előző, de az oldalon először jelenjen meg egy Űrlap, aminek segítségével tud szűrni az adatbázis adataira. Az űrlapon használja a következő elemeket: szöveges beviteli mező, lenyíló lista, radio gomb, jelölőnégyzet.  
   **Ír almenü**: Az oldalon jelenjen meg egy űrlap, aminek segítségével új rekordot tud hozzáfűzni az adatbázis egyik táblájához.  
   **Módosít almenü**: Az oldalon jelenjen meg egy űrlap, aminek segítségével egy már meglévő rekordot tud módosítani az adatbázis egyik táblájánál. A rekord azonosítója lenyíló listából legyen választható.  
   **Töröl almenü**: Az egyik tábla egy rekordjának törléséhez. A rekord azonosítója lenyíló listából legyen választható.
3. **Rest1** menü: Készítsen egy Restful klienst az eladáson bemutatott **(5 pont)**

<https://gorest.co.in> Rest szerverhez. A Create, Read, Update, Delete almenükben valósítsa meg a hozzájuk tartozó műveletet.

1. **SoapKliens menü:** Készítsen egy SOAP klienst a Magyar Nemzeti Bank **(5 pont)**  
   webszolgáltatásához.   
   **Letöltés almenü**: Letölti az összes adatot helyi adatbázisba.  
   **Letöltés2 almenü**: Az oldalon megjelenik egy űrlap (beviteli mezők, lenyíló lista, radio gomb, jelölő négyzet). Letölti a kiválasztott adatokat helyi adatbázisba és a Microsoft Azure felhő adatbázisába is.  
   **Grafikon almenü**: Letöltés2 almenühöz hasonlóan választ egy adatsort és megjeleníti az adatokat grafikonon is.  
   Minta: <https://www.mnb.hu/arfolyam-lekerdezes>
2. **Adatbányászat menü:** **(5 pont)**  
   Válasszanak ki egy fájlt az **Adatbányászat-források** mappából (az órán vizsgát **credit-g.arff** fájlt nem lehet választani). Válasszanak ki ebben egy olyan **attribútumot**, ami nem sok fajta adatot tartalmaz. Általában az utolsó Class nevű attribútum ilyen.   
   **Döntési fa almenü**: Készítsenek döntési fával gépi tanuló algoritmus a kiválasztott attribútum előrejelzésére. Eredményül írassák ki a **Döntési fa.txt** fájlba a következő adatokat:   
   Tanító és kiértékelő halmaz mérete,   
   TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances  
   A döntési fa kiíratása  
   A kiértékelő halmaz minden egyedére a prediktált és a valós érték  
   **Több algoritmus almenü**: A választott adathalmazra futtassa le a tanult algoritmusokat. A **Gépi tanulás.txt** fájlba írassa ki mindegyikre:   
   TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances  
   Válassza ki az algoritmusok közül azt, amelyiknek a legjobb lett a Correctly Classified Instances eredménye és írassa ki az algoritmus nevét az alkalmazás ablakába.  
   **Több algoritmus almenü-2**: Lenyíló listából lehessen választani a tanult algoritmusok közül. Az algoritmust lefuttatva írassa ki az alkalmazás ablakába a következő adatokat:  
   TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances
3. **Egyéb menü**: **(5 pont)**  
   **Páthuzamos almenü:** Mutassa be a párhuzamos programvégrehajtást egy oldalon. pl. egy gombra való kattintás után egy Label-ben 1 másodpercenként, a másik Labelbe 2 másodpercenként jelenjen meg egy változó szöveg.  
   **Stream almenü**: Olvassa ki a válsztott adatbázis minden adatát egy Stream-be. Késztsen egy űrlapot (beviteli mezők, lenyíló lista, radio gomb, jelölő négyzet), ahol több szűrőfeltételt is beállíthat az adatokra. Jelenítse meg egy táblázatban a szűrt adatokat.
4. **Ez a menüpont később kerül ismertetésre (5 pont)**
5. Készítsen egy **JAR futtatható fájlt**, ami a GitHub-on is megjelenik a főmappában.   
   **(Kötelező elem! a működés ez alapján lesz javítva)**
6. Használják a GitHub (github.com) verziókövető rendszert.   
   (**Kötelező elem**! **A forrás ez alapján lesz javítva**)

A GitHub-on a projektmunka módszert alkalmazzák: látszódjék, hogy a csoport tagjai melyik részt készítették el és kb. fele-fele arányban járuljanak hozzá a projekthez. Ne csak a kész alkalmazást töltsék fel egy lépésben, hanem a részállapotokat is még legalább 5 lépésben személyenként.  
A GIT-en a főmappában jelenjen meg a választott adatbázis export fájlja (**adatbazis.sql**).

**A GIT-en saját nevet válasszanak, ami alapján be lehet azonosítani, hogy ki mit töltött fel.**

1. A gépi és nyelvi igények

Az alkalmazás Java programozási nyelven történő fejlesztésével biztosítottam a széles körű kompatibilitást és platformfüggetlenséget. Az alkalmazás Intellij IDEABAN készült Ennek eredményeként az alkalmazás futtatásához javasolt specifikációk a következők:

Ajánlott specifikációk:

* Rendszerarchitektúra:

ARM vagy amd-64 architektúrájú rendszer

* Operációs rendszer:

Kompatibilis bármilyen olyan operációs rendszerrel, amelyre telepíthető a Java futtató környezet. Ez magában foglalja, de nem korlátozódik a Windows, macOS és Linux operációs rendszereket.

* Hálózati kapcsolat:

Széles sávú internetkapcsolat, hogy az alkalmazás zavartalanul és hatékonyan kommunikálhasson a szükséges szerverekkel és egyéb online szolgáltatásokkal.

* Tárhely:

Minimum 2 GB szabad hely a tárolóeszközön az alkalmazás telepítéséhez és futtatásához.

* Az említett specifikációk biztosítják, hogy az alkalmazás széles körben elérhető legyen és optimálisan működjön a felhasználók különböző hardver- és operációs rendszerkonfigurációin.

1. A mappa és fájlszerkezet bemutatását név és funkció szerint
   1. Github repository tartalma:

A képen szöveg látható

Automatikusan generált leírás

* DankoKrisztianZYEEHM mappa:

Projekt könyvtára, amely tartalmazza az alkalmazás forráskódját és egyéb szükséges fájlokat.

* ZYEEHM-Eloadas-Dokumentáció.docx:

A dokumentáció, amely részletesen ismerteti az alkalmazás tervezését, implementációját és használatát. Tartalmazza a funkcionális és nem funkcionális követelményeket, valamint az alkalmazás architektúráját.

* mozimusor.sql:

Az alkalmazás adatbázisához tartozó SQL fájl, amely definiálja az adatokat és táblákat az alkalmazás számára.

* 1. A Java osztályok:
* Database:

Az összekötött táblás lekérdezésekhez szükséges modell, amely az adatbázis táblákat reprezentálja.

* DatabaseDAO:

Tartalmazza a függvényeket és adattagokat az összekötött táblás lekérdezések elvégzéséhez.

* HelloApplication:

A fő alkalmazás, ami tartalmazza a main függvényt, és felelős az alkalmazás inicializálásáért.

* HelloController:

Az alkalmazás szerveroldali logikáját vezérli, kiválasztja, mely nézetet jeleníti meg a felhasználónak.

* Mozi:

A mozi tábla lekérdezéséhez szükséges modell, amely reprezentálja a mozi adatait az alkalmazásban.

* MoziDAO:

Implementálja a mozi táblával kapcsolatos adatbázis műveleteket, lehetővé téve az adatok hatékony kezelését.

* RestKliens:

Tartalmazza a függvényeket a 2-es feladatban levő REST szerverrel történő kommunikációhoz.

* RestUser:

A 2-es feladatban levő REST szerverhez szükséges modell, amely reprezentálja a REST API által visszaadott felhasználói adatokat.

* Launcher:

Segíti a projekt Java Archive (JAR) fájlba történő fordítását, és lehetővé teszi az alkalmazás egyszerű indítását.

* Ezen osztályok együttesen alkotják az alkalmazás alapstruktúráját és funkcionalitását, biztosítva a hatékony adatkezelést, a szerverrel való kommunikációt, valamint az alkalmazás egyszerű indítását és futtatását.

**1. Adatbázis Menü**

Olvas Almenü:

Megjeleníti az adatbázis adatait egy táblázatban, kiemelve legalább 3 tábla fontos információját. Az oldalon csak egy részét jeleníti meg az adatoknak a felhasználó túlzott terhelése elkerülése érdekében.

Olvas2 Almenü:

Hasonlóan az előzőhöz, azonban először egy űrlap jelenik meg, ahol a felhasználó szűrheti az adatokat. Az űrlapon szöveges beviteli mezők, lenyíló lista, radio gombok, és jelölőnégyzetek segítik a felhasználót a pontosabb szűrésben.

Ír Almenü:

Ezen az oldalon egy űrlap segítségével a felhasználó új rekordot adhat hozzá az adatbázis egyik táblájához.

Módosít Almenü:

Az oldalon megjelenik egy űrlap, ahol a felhasználó meglévő rekordokat módosíthat az adatbázis egyik táblájában. A rekord azonosítója lenyíló listából választható.

Töröl Almenü:

Az egyik tábla rekordjának törléséhez egy űrlap jelenik meg, ahol a felhasználó kiválaszthatja a törlendő rekordot lenyíló listából.

**2. Rest1 Menü**

Készítettem egy Restful klienst a gorest.co.in szerverhez, ahol a Create, Read, Update, Delete műveleteket is implementáltuk.

**3. SoapKliens Menü**

Letöltés2 Almenü:

Az oldalon megjelenik egy űrlap, ahol a felhasználó kiválaszthatja, mely adatokat szeretné letölteni.

Grafikon Almenü:

Hasonlóan a Letöltés2-höz, a felhasználó választhat egy adatsort és megjeleníti az adatokat grafikonon is.

**4. Adatbányászat Menü**

Döntési Fa Almenü:

Készítettem egy döntési fát gépi tanuló algoritmus segítségével a választott attribútum előrejelzésére. A Döntési fa.txt fájlban rögzítve van a tanító és kiértékelő halmaz méretét, TP, TN, FP, FN, Correctly Classified Instances, Incorrectly Classified Instances, valamint a döntési fát.

Több Algoritmus Almenü:

Futtattam és rögzítettem a Gépi tanulás.txt fájlba a tanult algoritmusok eredményeit. Az alkalmazás ablakában megjelenítődik az algoritmus neve, amely a legjobb Correctly Classified Instances eredményt produkálta.

Több Algoritmus Almenü-2:

Lenyíló listából választhatóak a tanult algoritmusok, és az eredményeket megjeleníti az alkalmazás ablakában.

**5. Egyéb Menü**

Párhuzamos Almenü:

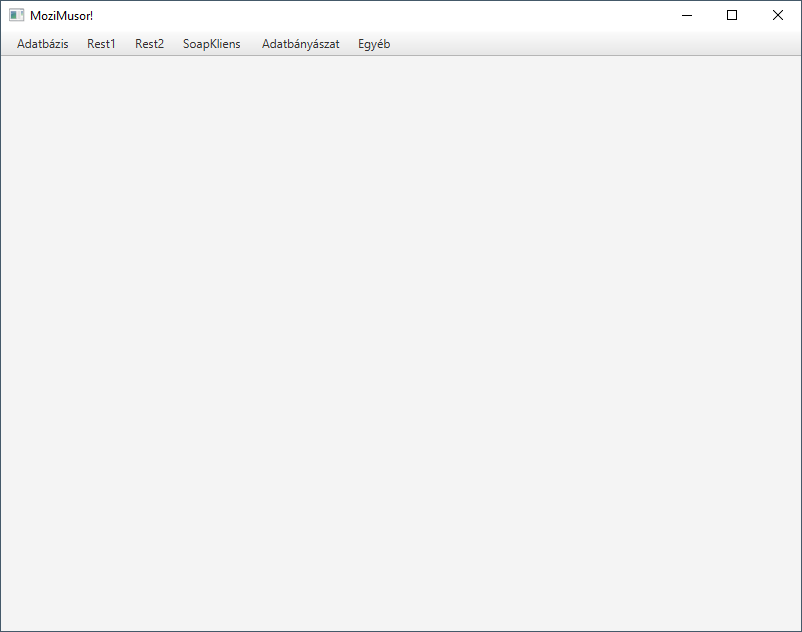
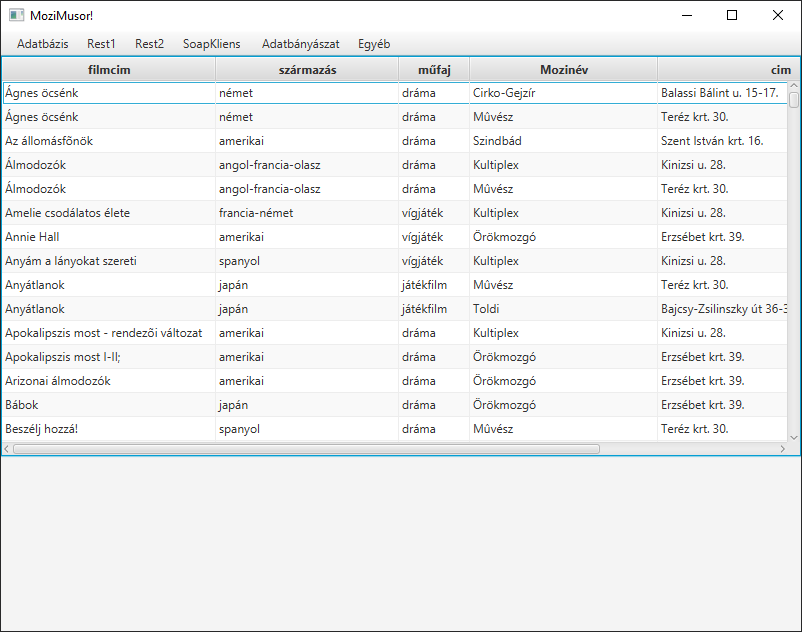
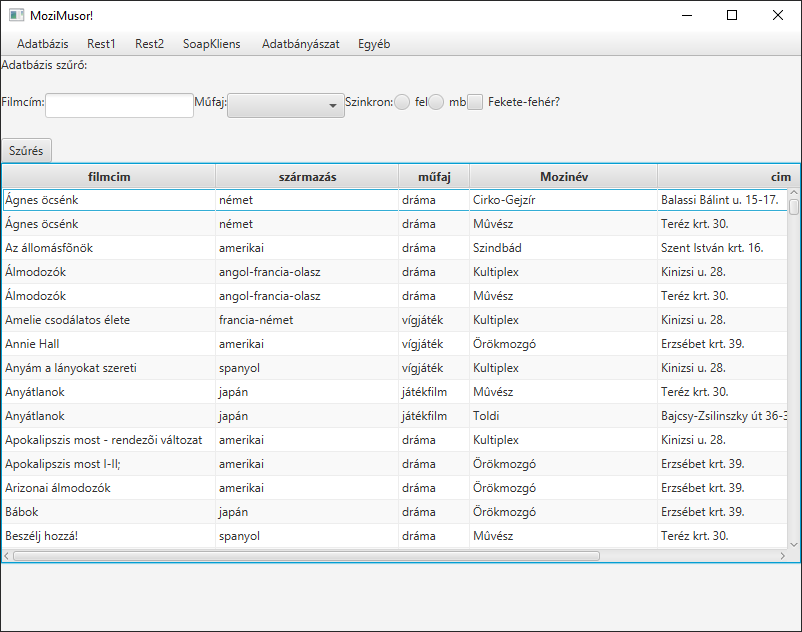
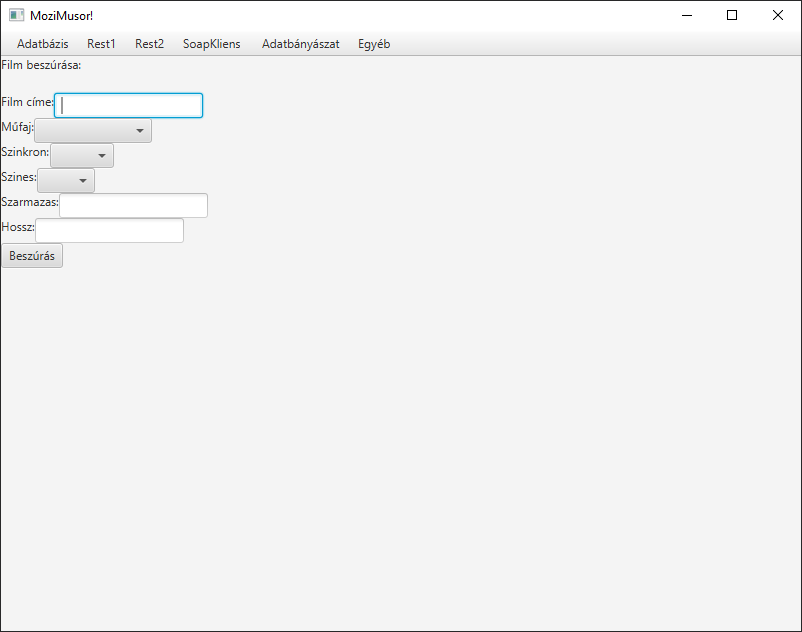
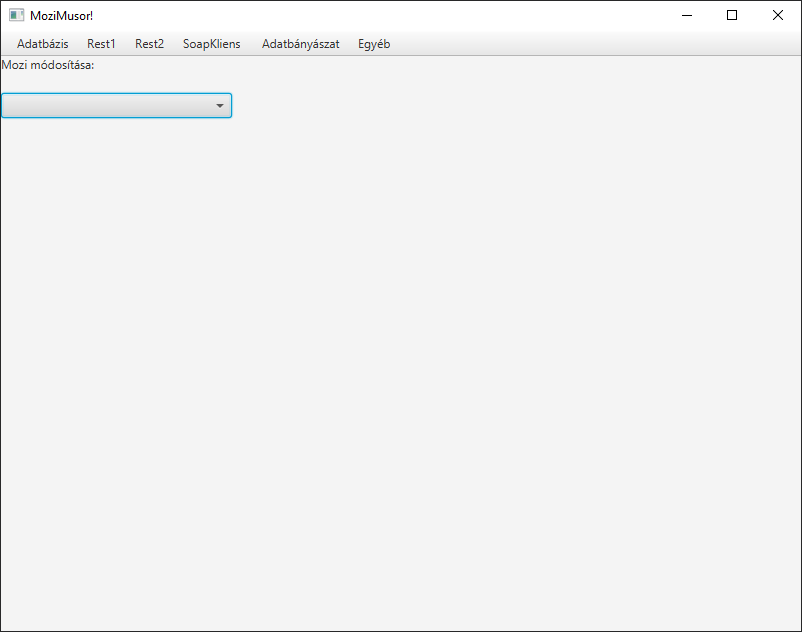
Elkészítettem a párhuzamos programvégrehajtást egy gombra kattintás után, ahol két Label-ben jelennek meg változó szövegek különböző időközönként.

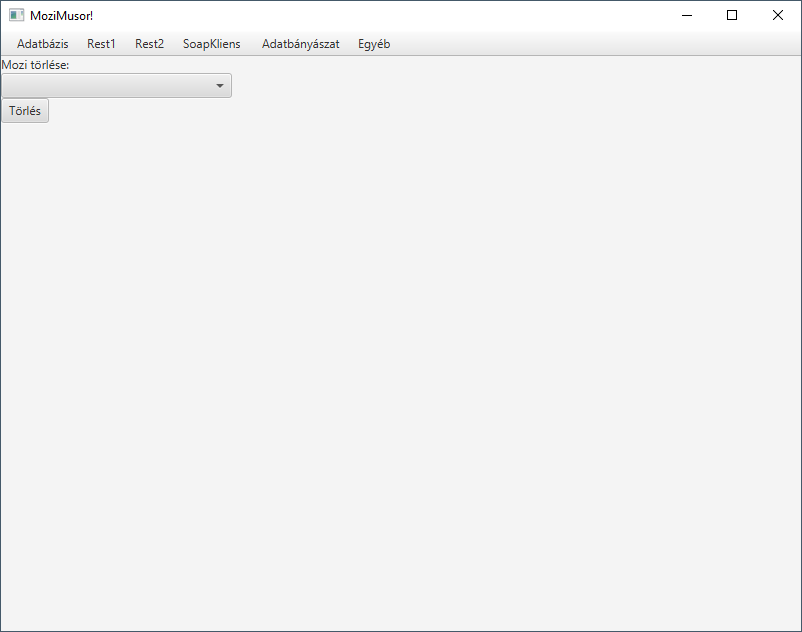
Stream Almenü:

ki olvastam a választott adatbázis összes adatát egy Stream-be, majd készítettem egy űrlapot a szűréshez, és megjelenítettem a táblázatban a szűrt adatokat.

**7. JAR Fájl és GitHub**

Készítettem egy futtatható JAR fájlt, amely a GitHub főmappájában elérhető..

1. Képernyőképek
   1.  Alaplap
   2. Olvasas1
   3. Olvasas2
   4. Írás
   5. Módosítás
   6. Törlés



Internetes címek, Irodalomjegyzék

Github: <https://github.com/kriszmas/Eloadas-Beadando>

Használt anyagok:

<https://teams.microsoft.com/_?culture=hu-hu&country=hu#/apps/3e0a4fec-499b-4138-8e7c-71a9d88a62ed/sections/MyNotebook>

<https://teams.microsoft.com/_?culture=hu-hu&country=hu#/apps/3e0a4fec-499b-4138-8e7c-71a9d88a62ed/sections/MyNotebook>

<https://teams.microsoft.com/_?culture=hu-hu&country=hu#/zip/viewer/teamsSdk/https:~2F~2Funineumannhu.sharepoint.com~2Fsites~2FCLS23241N-K-GINFBAN-JAVAALKA-1-EA01~2FClass%20Materials~2FJava%20alkalmaz%C3%A1sok-Alapk%C3%A9pz%C3%A9s-El%C5%91ad%C3%A1s-Beadand%C3%B3.zip?threadId=19:vK_jk9Xf-5SHjy1r276c1wlkSzt-YJcJ46TprgFUGkQ1@thread.tacv2&baseUrl=https:~2F~2Funineumannhu.sharepoint.com~2Fsites~2FCLS23241N-K-GINFBAN-JAVAALKA-1-EA01&fileId=5e19eac1-1e17-4b30-a5a9-210c1a59bb98&ctx=openFilePreview&viewerAction=view>