Intézmény:

**Neumann János Egyetem**

**GAMF kar,**

**Informatika tanszék**

Szak megnevezése:

**Mérnökinformatikus alapszak**

Tantárgy neve:

**Java programozás Előadás**

|  |  |  |
| --- | --- | --- |
| **Hallgatók neve:** | **Neptun kódok:** | **Emailek:** |
| Polyák Szabolcs  Cseh Dániel Kristóf | ESILHK  I1DSMV | brank321@gmail.com  kristof.daniel.cseh@gmail.com |

**Git Repository címe:**

https://github.com/CsehDanielKristof/JavaFX

Tartalomjegyzék

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

[2. A gépi és nyelvi igények 8](#_Toc120828571)

[2.1. Ajánlott specifikációk: 8](#_Toc120828572)

[3. A mappa és fájlszerkezet bemutatását név és funkció szerint 9](#_Toc120828573)

[3.1. Github repository tartalma: 9](#_Toc120828574)

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

[4. Képernyőképek 10](#_Toc120828576)

[4.1. Alaplap 10](#_Toc120828577)

[4.2. Olvasas1 11](#_Toc120828578)

[4.3. Olvasas2 12](#_Toc120828579)

[4.4. Írás 13](#_Toc120828580)

[4.5. Módosítás 14](#_Toc120828581)

[4.6. Törlés 15](#_Toc120828582)

1. Feladat Pontos leírása

Java alkalmazások Alapképzés Előadás Beadandó feladat

A feladathoz segítséget a feladat végénél talál!

Alakítsanak ki két-fős csoportokat és a feladatot közösen, projektmunkában oldják meg. A dokumentációban majd írják le, hogy a két főből ki melyik feladatrészt csinálta meg.

A csoportoknál a kapcsolatfelvételt és a közös munkát a Teams-en meg lehet oldani.

Ha valaki nem talál társat (pl. páratlanul vannak a csoportban), a GitHub-os feladatot akkor is projektmunkában csinálja meg két GitHub fiókot felhasználva.

Az Adatbázisok mappában találnak adatbázis forrásokat, amik közül kell egyet a csoportnak választani és felhasználni a feladatában. Egy adatbázist maximum 1 csoport választhat.

Az adatbázist lehet újabb táblákkal bővíteni. Importálja be a választott adatbázis adatait.

A feladatban egy menüvel ellátott grafikus asztali alkalmazást kell készíteni Java nyelven JavaFX technológiával.

Összesen 30 pont

1. Adatbázis menü: egy CRUD alkalmazás a választott adatbázissal​​​​(5 pont)

Olvas almenü: Megjeleníi 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.

1. 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. Készítsen a Microsoft Azure felhőben egy ReatFul szerver szolgáltatást a választott ​(5 pont)

adatbázis egyik táblája alapján.

Rest2 menü: Készítsen egy Restful klienst a Microsoft Azure felhőben létrehozott 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

1. 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

1. 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.

1. Készítsen egy JAR futtatható fájlt, ami a GitHub-on is megjelenik a főmappában.​

8. Használják a GitHub (github.com) verziókövető rendszert. ​​​

(Kötelező elem! A forrás ez alapján lesz javítva)

1. 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.
2. A gépi és nyelvi igények

Az alkalmazás fejlesztése a Java programozási nyelven történt. Így az alkalmazás használatához java futtató környezetet kell telepíteni.

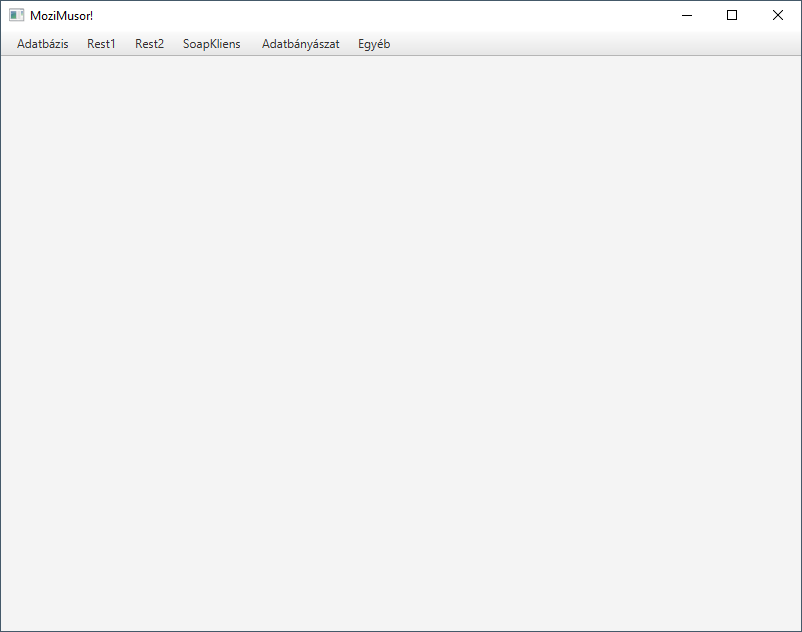
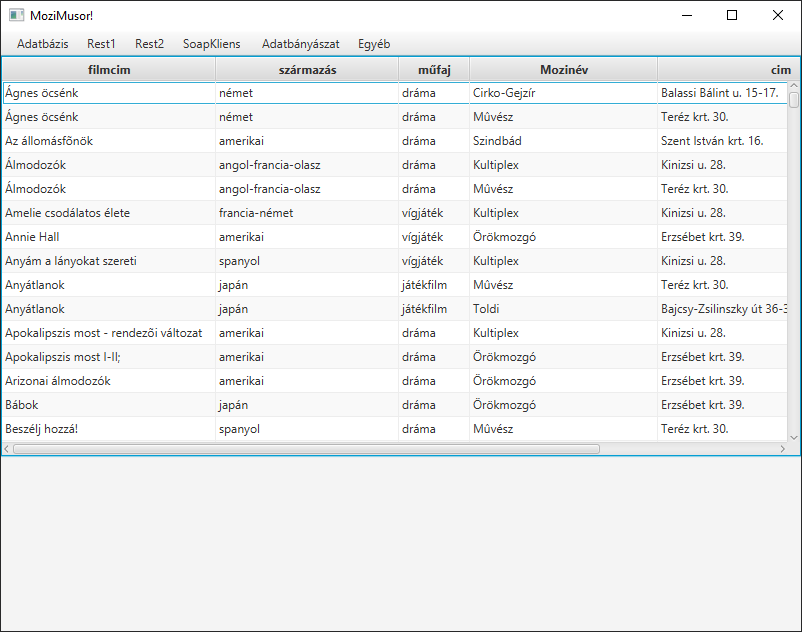
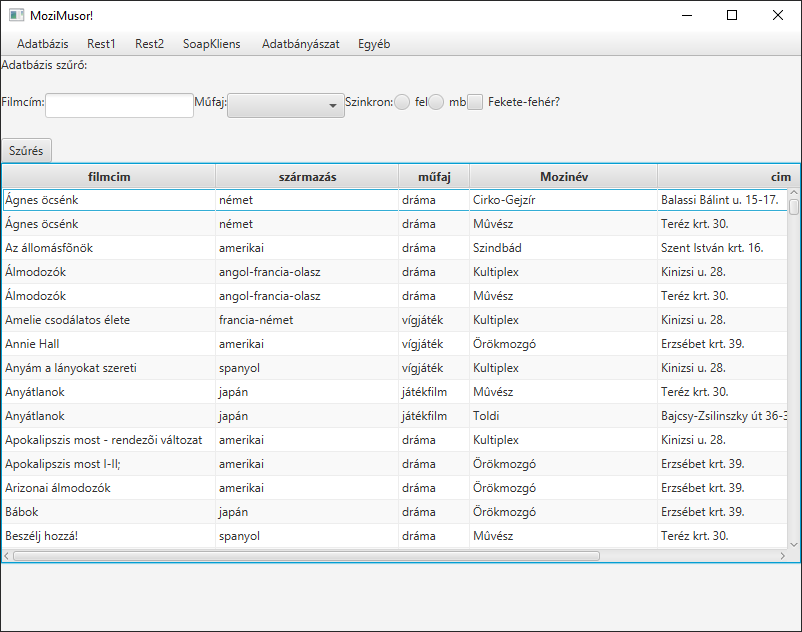
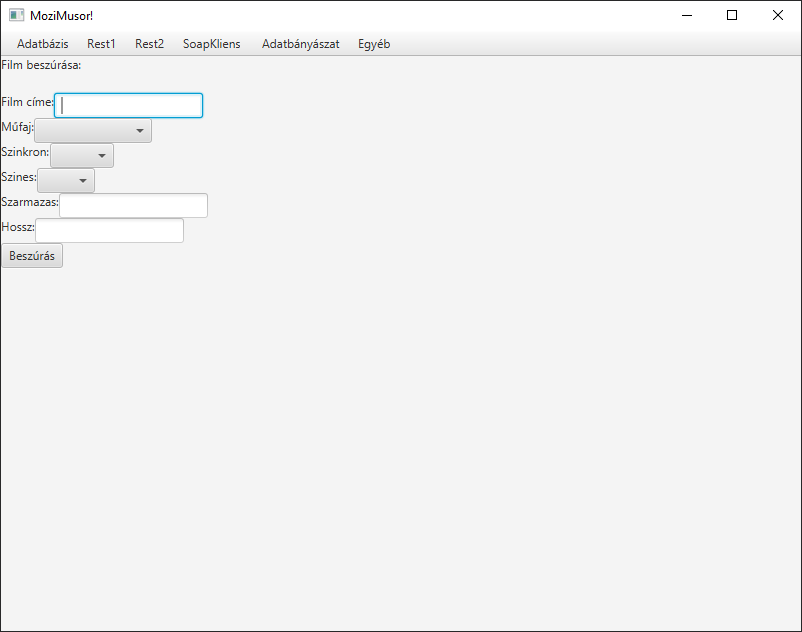
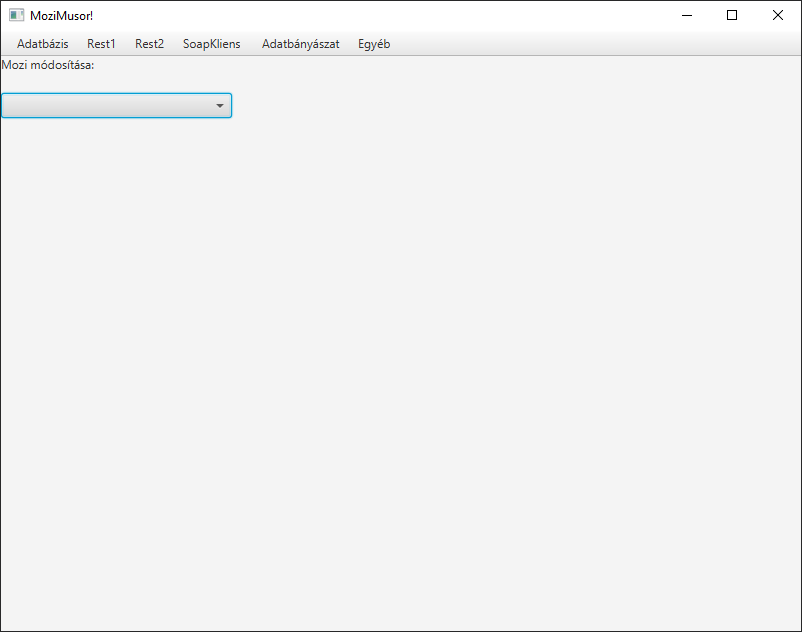
* 1. Ajánlott specifikációk:
* ARM vagy amd-64 architektúrájú rendszer
* Operációs rendszer: Bármilyen OS. amin telepíthető java futtató környezet.
* Hálózat: Széles sávú internetkapcsolat
* Tárhely: 2 GB szabad hely

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

A képen szöveg látható

Automatikusan generált leírás

* JavaFx\_Beadando mappa: A project könyvtára
* Java Előadás.docx: Dokumentáció
* mozimusor.sql: Az alkalmazás adatbázisához tartozó adatok, táblák
  1. A Java osztályok:
* Database: Az összekötött táblás lekérdezéshez szükséges modell.
* DatabaseDAO: Függvények és adattagok az összekötött táblás lekérdezéshez.
* HelloApplication: Ez tartalmazza a main függvényt.
* HelloController: A szerveroldali logikát tartalmazza, vezérli az alkalmazást és kiválasztja mely nézetet mutatja meg a felhasználónak.
* Mozi: A mozi tábla lekérdezéséhez szükséges modell.
* MoziDAO: A mozi táblával kapcsolatos adatbázis műveleteket valósítja meg.
* RestKliens: A 2-es feladatban levő rest szerverrel való kommunikációnak függvényei.
* RestUser: A 2-es feladatban levő rest szerverhez szükséges modell.
* Launcher: A projekt jar file-ba való fordítását segíti.

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

