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](mailto:brank321@gmail.com)  kristof.daniel.cseh@gmail.com |

**Git Repository címe:**

https://github.com/CsehDanielKristof/JavaFX

Tartalomjegyzék

[1. Feladat Pontos leírása 5](#_Toc120803633)

[2. A gépi és nyelvi igények 8](#_Toc120803634)

[2.1. Ajánlott specifikációk: 8](#_Toc120803635)

[3. A mappa és fájlszerkezet bemutatását név és funkció szerint 9](#_Toc120803636)

[3.1. Github repository tartalma: 9](#_Toc120803637)

[3.2. A Java osztályok: 9](#_Toc120803638)

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. ​​​

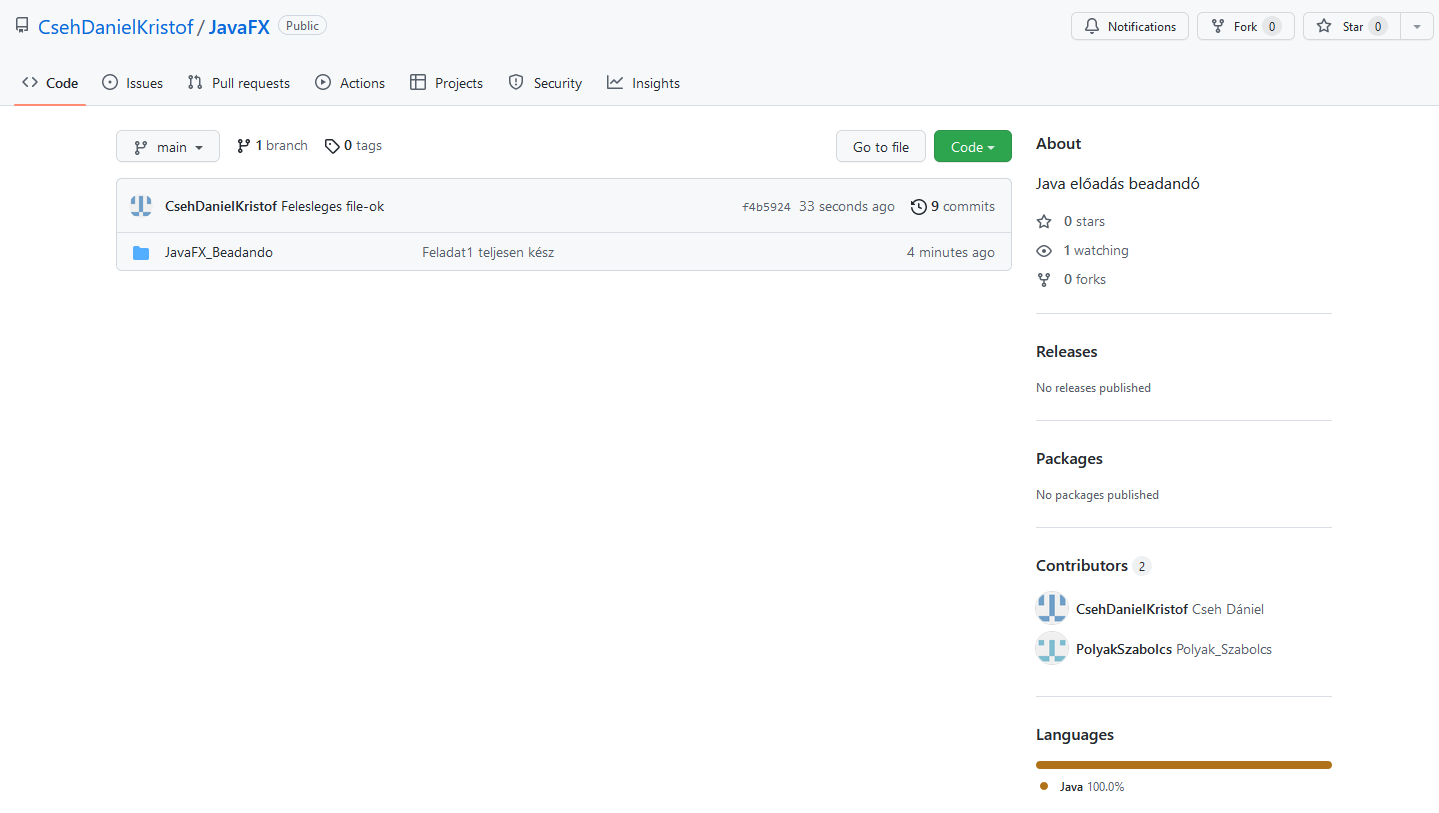
(Kötelező elem! A forrás ez alapján lesz javítva)

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

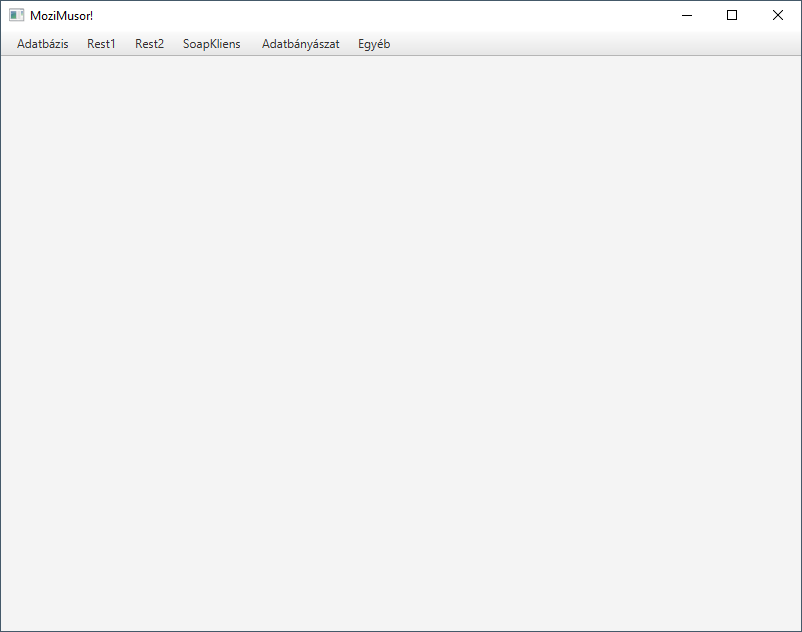
A program Java, nyelveket használ. A programnak nincs nagy követelménye. A fejlesztés egy Intel i5 4600 és egy intel i5 4700k cpu-n történt amely a program futását teljes mértékben ki tudta szolgálni 16GB ram társaságában . A rendszer windows 10 és mac os 13.0.1 alatt is tesztelve lett zökkenő mentesen élményt biztosit a fentebb emlitett hardveres követelmény .

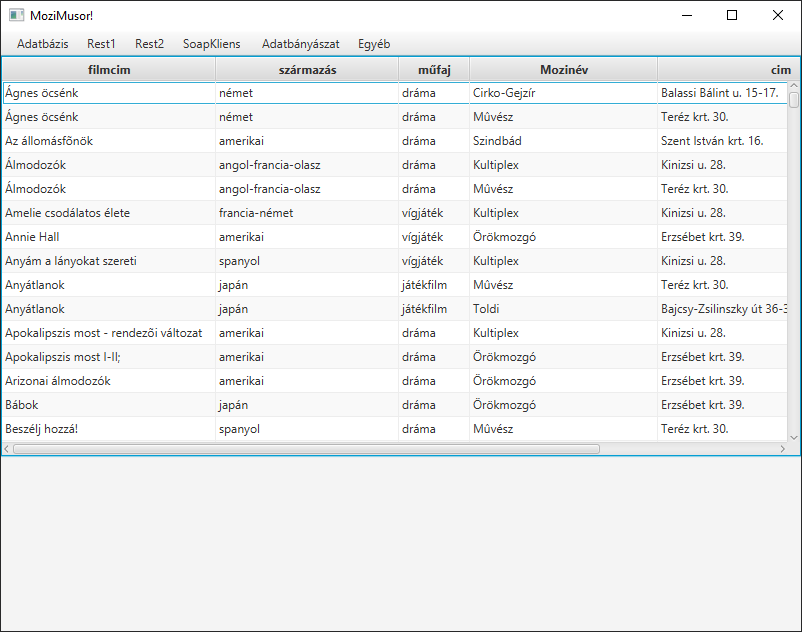
* 1. Ajánlott specifikációk:

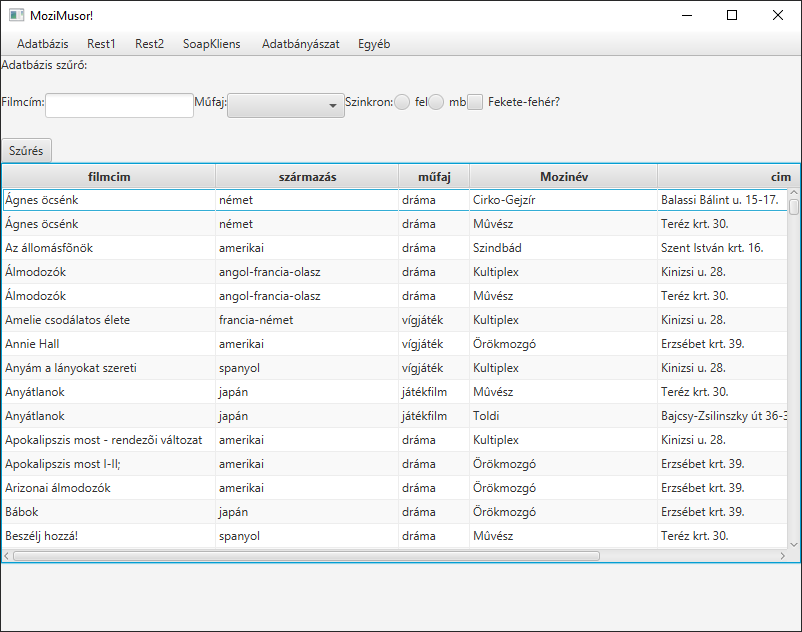
1. 64 bites processzor és operációs rendszer szükséges
2. Op. rendszer: Windows 8 vagy annál frisseb operációs rendszer
3. Processzor: 2.0 GHz quad core processor
4. Memória: 8 GB RAM
5. Hálózat: Széles sávú internetkapcsolat
6. Tárhely: 2 GB szabad hely
7. A mappa és fájlszerkezet bemutatását név és funkció szerint
   1. Github repository tartalma:

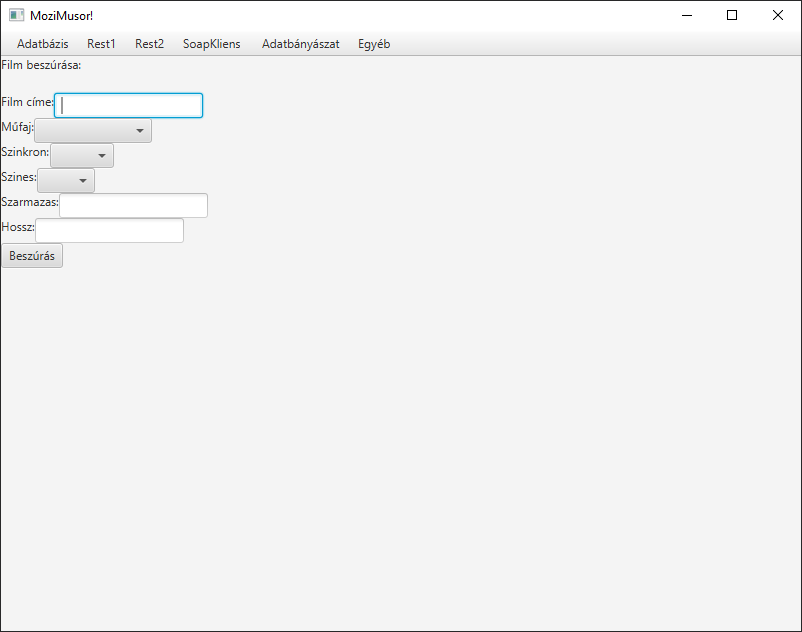


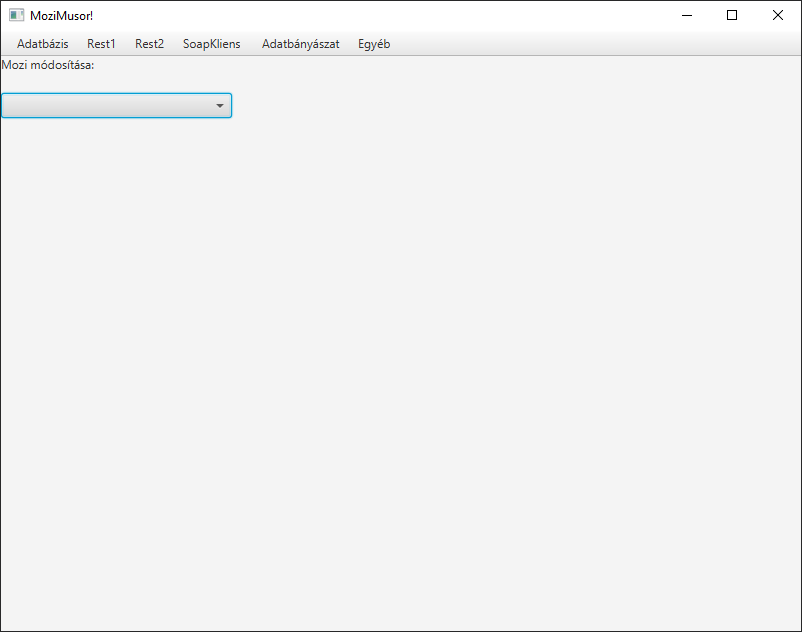
* JavaFx\_Beadando mappa: A project könyvtára
  1. A Java osztályok:
* AdatbazisFuggvenyek: A különböző adatbázist meghívó függvények helye
* AzureRestKliens: Azure API Kliensel való kommunikációért felelős
* HelloApplication:
* HelloController: A megjelenésért felelős Controllerek itt helyezkednek el
* Kategoria: Egy táblát megvalósító adat modell
* osszRendelés: Ez egy több tábla adatait összefogó osztály, egy speciális lekérdezéshez kell, melyet csak az admin ér el
* RestKliens: 2 feladatban lévő rest kommunikációját megavalósitó függvények helye
* RestUser: 2 feladatban lévő rest adatok osztályinak a kiszolgálásáért felelős

Alaplap

Olvasas1

Olvasas2

Írás

Módosítás

Törlés

