**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.
2. **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.
2. **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>
3. **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
4. **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.
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.**

**Készítsen legalább 15 oldalas dokumentációt (PDF formátumban)**, amiben leírja az alkalmazás működését képernyőképekkel. (**Kötelező elem**!) A dokumentációban adja meg a GitHub projektjének URL címét **és írja le részletesen, hogy az előző feladatpontokat az alkalmazásban hogy és hol valósította meg**. Fontos, hogy ez utóbbit leírja a dokumentációban, mert a feladat ez alapján lesz javítva.

**Be kell adni** a **Neptun MeetStreet**-en keresztül a dokumentációt. Akik közösen csinálják, mindenki adja be a dokumentációt.

**Segítség az egyes részekhez**

|  |  |
| --- | --- |
| **Feladatrész** | **Fejezet az elméleti anyagban** |
| Adatbázis menü | Grafikus programozás-Adatbázissal |
| Rest1 és Rest2 menü | REST kliens és szerver |
| SoapKliens menü | SOAP kliens |
| Grafikon | Grafikus programozás |
| Adatbányászat menü | Adatbányászat-Gépi tanulás |
| Páthuzamos és Stream almenük | Szálkezelés, Lambda kifejezés, Stream, Funkcionális programozás |
| JAR futtatható fájl grafikus programhoz | Grafikus programozás |
| GitHub | GitHub-GitLab használat |
| Microsoft Azure felhő | Java-Spring-Megjelenítés az Interneten-Microsoft Azure-al |