**Név Neptun azonosító Gép szám Aláírás**

**Általános tudnivalók**

* Bejelentkezés:
  + A szokásos módon történik, Windows operációs rendszer, Irinyi Kabinet felhasználó.
  + A **feladat**ot tartalmazó U: és a közös K: meghajtót az asztalon található **Home\_Mount** szkripttel kell felcsatolni.
  + **Felhasználó név a saját** hallgatói h-s azonosító, és a jelszó a hozzá tartozó jelszó.
* A **ZH feladatsor** az **U: meghajtón** található a „**feladat”** könyvtárban:
  + A teljes „**feladat”** könyvtárat át kell másolni a „**C:\TEMP”** könyvtárba.
  + A teljes „**feladat”** könyvtárat óra végén **(a megoldást) vissza kell másolni** az **U: meghajtóra** a kézzel létrehozott „**megoldas”** könyvtárba (az eredeti „**feladat”** könyvtárra csak olvasási jog van).
  + **Javasolt** a megoldásokat **időnként, pl. legalább minden** **ZH** **feladat megoldása után** átmásolni az U: meghajtóra (nem csak óra végén).
  + **Begyűjtés: óra :40-kor!**
* Azonosítás:
  + A **„megoldas”** könyvtárban mindenki hozza létre a „**gepazonosíto-hallgatoneve-Neptun\_azonosito.txt**” sablonnak megfelelő szöveges fájlt, ékezetek nélkül!  
    A gépazonosítót (pl. pc22412) a gépről lehet leolvasni.  
    Pl.: pc22501-kiss-andras-kiakaat.txt  
    **FONTOS**: A fájlnak az óra végén vissza kell kerülnie az U: meghajtóra (U:\megoldas\pc22501-kiss-andras-kiakaat.txt)!
  + A gyakorlatvezető kiosztja a **ZH feladatsort papíron** is, amire a hallgatók felírják **nev**üket, **Neptun** azonosítójukat és a **gépazonosítót** **nyomtatott nagybetűvel, olvashatóan**, valamint **aláírják**.  
    Az óra végén **kötelező visszaadni**!
* A ZH-n nem használható semmilyen segédeszköz, csak ami a **K: meghajtón** található (dokumentációk, ppt-k, doc-ok).

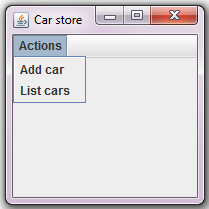
**Java GUI-DB (34 pont)**

A feladat egy egyszerű autókereskedés támogató program írása.

Ami már adott a programból:

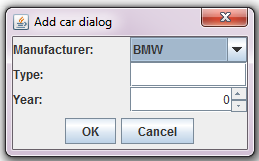
* build.xml - be lehet importálni Eclipse-be
* Main.java - példányosítja a controller-t és meghívja a startDesktop() metódusát
* CarStoreController osztály - üres, csak azért van létrehozva, hogy a main példányosítani tudja, és meg tudja hívni a startDesktop() metódust
* Adatbázis séma (kulso\db\carstore.ddl) - három jellemzőt kell csak eltárolni
* Egy üres adatbázis a fenti sémának megfelelően (kulso\db\carstore.db)

Megvalósítandó egy olyan alkalmazás, amelyik:

* Egy fő ablakkal rendelkezik, amin van egy menü Actions névvel, aminek 2 menüpontja van
  + Add car
  + List cars
* Az Add car menüt kiválasztva az Add car dialog ablak jelenjen meg
  + A gyártók (manufacturer) legördülő menü legyen inicializálva

a következő értékekkel: BMW, Opel, Nissan és Lada

* + OK gombra kattintva
    - Ellenőrizze, hogy az autó modellje (type) nem üres
    - Ha üres, akkor adjon egy hibaüzenetet
    - Mentse el az adatokat az adatbázisba
    - Sikeres mentés esetén zárja be az ablakot
    - Sikertelen mentés esetén adjon egy hibaüzenetet, de ne zárja be
    - Cancel gomb esetén zárja be az ablakot mentés nélkül
* A List cars menüpont hatására a View rétegből írassuk ki az autók listáját a konzolra (println)

További követelmények:

* A megvalósítás során kövessük az MVC modellt
* Az adatokat az adatbázisba kell menteni
  + A megadott sémát használja
  + A string konstansokat egy külön fájlba definiáljuk
* „Engedmények”
  + Nem kell környezeti változót használni az adatbázis

útvonalára: bele lehet kódolni a forráskódba

**Semmi egyéb funkcionalitást NEM kell megvalósítani, és nem is ér plusz pontot!!!**

Segítség:

**DB**

public BookShopDAODBImpl() {

try {

Class.*forName("org.sqlite.JDBC");*

} catch (ClassNotFoundException e) {

System.*out.println("Failed to load SQLite JDBC drv.");*

e.printStackTrace();

}

}

DriverManager.getConnection("jdbc:sqlite:"+DATABASE\_FILE);

**GUI**

JOptionPane.showMessageDialog(

gui.getWindow(),

*"Error message"*,

*"ERROR"*,

JOptionPane.ERROR\_MESSAGE);

**GUI indítás**

SwingUtilities.invokeLater( new Runnable() {

@Override

public void run() {

…

}

});

**XML-XSLT feladat (16 pont)**

A feladat a proglangs.xml fájlhoz egy XSLT transzformáció készítése, amely az alábbi szöveges kimeneti fájlt eredményezi. A transzfromációnak a parancssori xslt.bat fájllal, env.bat környezetben kell előállnia.

A proglangs.txt fájl tartalma:

Name,class,designer,year,vm

C,Imperative,Ritchie,1972,no

C#,Object-oriented,Anders Hejlsberg,1998,yes

Java,Object-oriented,Sun Microsystems,1994,yes

Active Oberon,Object-oriented,Jürg Gutknecht,2002,no

**Segítség:** Az xsl:stylesheet elem alá szúrjuk be a következő sort, hogy szöveges kimenetet gyártsunk:

<xsl:output method="text" encoding="UTF-8"/>

Minden kimenetre szánt szöveget tegyünk az <xsl:text></xsl:text> tag-ek közé.

Új sor beszúrása: <xsl:text>&#xa;</xsl:text>