Autó kölcsönző

Programozás alapjai 3 házi feladat

Dokumentáció

Bárdi Csaba

BH9HDV

1 A FELADAT LEÍRÁSA

A program lényegében egy autó kölcsönző nyilvántartását teszi lehetővé. A kölcsönzőben lehetőség van többféle autót is bérelni, ezért szükség van az autók nyilvántartására. Az autóknak több tulajdonságát is kezeli a program például: rendszám, típus, férőhely. Az autókon kívül a bérlőknek is rögzíteni kell az adatait, amik a következők: név, cím, telefonszám. Továbbá az éppen aktuális bérlések nyilvántartására is szükség van, hogy eldönthető legyen melyik autókat lehet bérbe adni.

Fontos funkció még, hogy a tárolt adatok között keresni is lehessen. Ezért a programban lehet bérlő vagy autó szerint is keresni. Másik fontos funkció még a biztonság, ezért autót bérbe adni csak hitelesített felhasználó tud, amiről szerződés is készül. A többi funkciót nem hitelesített felhasználó is eléri. A programot alapvetően egy GUI által lehet irányítani, ami egy grafikus interfész. A programnak lesz egy menüje is amiben különböző gombok és text mezők vannak.

2 USE-CASE-EK

Cím	Autók kezelése
Leírás	A felhasználó hozzáad egy autót az autók nyilvántartásához, vagy eltávolít egyet abból.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A felhasználó sikeresen hozzáadta az autót a nyilvántartásához, vagy eltávolította. **2.** A hozzáadás vagy eltávolítás nem sikerült valamilyen hiba folyamán.

Cím	Bérlők kezelése
Leírás	A felhasználó hozzáad egy bérlőt az bérlők nyilvántartásához, vagy eltávolít egyet abból.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A felhasználó sikeresen hozzáadta az bérlőt a nyilvántartásához, vagy eltávolította. **2.** A hozzáadás vagy eltávolítás nem sikerült valamilyen hiba folyamán.

Cím	Keresés bérlő szerint
Leírás	A felhasználó keresni tud a bérlők nyilvántartásában.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A felhasználó megtalálta a keresni kívánt bérlőt, és látja az adatait. **2.** A keresés nem dobott ki találatot.

Cím	Keresés autó szerint
Leírás	A felhasználó keresni tud az autók nyilvántartásában.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A felhasználó megtalálta a keresni kívánt autót, és látja az adatait. **2.** A keresés nem dobott ki találatot.

Cím	Keresés korábbi bérlések között
Leírás	A felhasználó keresni tud a korábbi bérlők között.
Aktorok	Felhasználó
Főforgatókönyv	**1.** A felhasználó megtalálta a keresni kívánt bérlést, és látja az adatait. **2.** A keresés nem dobott ki találatot.

Cím	Bérbe adás
Leírás	Az admin készít egy bérleti szerződést.
Aktorok	Admin
Főforgatókönyv	**1.** A felhasználó hitelesítette magát, és elkészítette a bérleti szerződést. **2.** A felhasználónak nem sikerült a hitelesítés. **3.** A bérleti szerződés sikertelen volt valami hiba folytán.

3 STRUKTURÁLIS LEÍRÁS

3.1 AZ OSZTÁLYOK LEÍRÁSA

3.1.1 Auto

Felelősségek

Tárolja az autó adatait.

Attribútumok

-rendszam: String	Az autó rendszáma.
-tipus: String	Az autó típusa
-ferohely: int	Az aszteroida magjának típusa.

Metódusok

+toString(): String	Visszadja az autó adataiból képzett stringet.

3.1.2 Berlo

Felelősségek

Tárolja a bérlő adatait.

Attribútumok

-id: String	A bérlő azonosítója.
-nev: String	A bérlő neve.
-cim: String	A bérlő lakcíme.
-telefonszam: String	A bérlő telefonszáma.
-azonositok: Arraylist<String>	Az összes bérlőnek az azonosítója.

Metódusok

+toString(): String	Visszadja a bérlő adataiból képzett stringet.
+createID(azonositok: ArrayList<String>): String	A paraméterként kapott azonosító stringekből álló listától különböző azonosítót ad vissza.

3.1.3 Berles

Felelősségek

Tárolja a bérlések adatait.

Attribútumok

-auto: Auto	A bérlésben szereplő autó
-berlo: Berlo	A bérlésben szereplő bérlő
-datum: String	A bérlés kiállításának dátuma.
-aktiv: boolean	Aktív-e a bérlés.

Metódusok -

3.1.4 AutokTable

Felelősségek

Egy saját abstract table model, ami tárolja az autók listáját.

Attribútumok

-autok: Arraylist<Auto>	Az autók listája.

Metódusok

+removeRow(rowIndex: int)	Kitörli a paraméterként kapott sort és az annak megfelelő objektumot a listából.
+addAuto(rendszam: String, tipus: String, ferohely: int)	A paraméterként kapott adatokból készít egy autót, amit hozzáad a listához.
+searchAuto(rendszam: String): Auto	A paraméterként kapott rendszam segítségével megkeresi a megfelelő autót és visszaadja.
+getRowCount(): int	Visszaadja a sorok számát.
+getColumnCount(): int	Visszaadja az oszlopok számát.
+getValueAt(rowIndex: int, columnIndex: int): Object	A paraméterként kapott sornál és oszlopnál visszaadja a cella értékét.
+getColumnName(rowIndex: int): String	A paraméterként kapott sornál visszaadja az oszlopok nevét.
+getColumnClass(rowIndex: int): Class<?>	A paraméterként kapott sornál visszaadja az oszlopok osztályát.
+isCellEditable(rowIndex: int, columnIndex: int): boolean	A paraméterként kapott sornál és oszlopnál visszaadj, hogy módosítható-e a cella.
+setValueAt(o: Object, rowIndex: int, columnIndex: int)	A paraméterként kapott sornál és oszlopnál felülírja a cella értékét.

3.1.5 BerlokTable

Felelősségek

Egy saját abstract table model, ami tárolja a bérlők listáját.

Attribútumok

-berlok: Arraylist<Berlo>	A bérlők listája.

Metódusok

+removeRow(rowIndex: int)	Kitörli a paraméterként kapott sort és az annak megfelelő objektumot a listából.
+addBerlo(nev: String, cim: String, telefonszam: String)	A paraméterként kapott adatokból készít egy bérlőt, amit hozzáad a listához.
+searchBerlo (id: String): Berlo	A paraméterként kapott azonosító segítségével megkeresi a megfelelő bérlőt és visszaadja.
+getRowCount(): int	Visszaadja a sorok számát.
+getColumnCount(): int	Visszaadja az oszlopok számát.

+getValueAt(rowIndex: int, columnIndex: int): Object	A paraméterként kapott sornál és oszlopnál visszaadja a cella értékét.
+getColumnName(rowIndex: int): String	A paraméterként kapott sornál visszaadja az oszlopok nevét.
+getColumnClass(rowIndex: int): Class<?>	A paraméterként kapott sornál visszaadja az oszlopok osztályát.
+isCellEditable(rowIndex: int, columnIndex: int): boolean	A paraméterként kapott sornál és oszlopnál visszaadj, hogy módosítható-e a cella.
+setValueAt(o: Object, rowIndex: int, columnIndex: int):	A paraméterként kapott sornál és oszlopnál felülírja a cella értékét.

3.1.6 BerlesekTable

Felelősségek

Egy saját abstract table model, ami tárolja a bérlések listáját.

Attribútumok

-berlesek: Arraylist<Berles>	A bérlések listája.

Metódusok

+removeRow(rowIndex: int)	Kitörli a paraméterként kapott sort és az annak megfelelő objektumot a listából.
+addBerles(auto: Auto, berlo: Berlo)	A paraméterként kapott adatokból készít egy bérlést, amit hozzáad a listához.
+getRowCount(): int	Visszaadja a sorok számát.
+getColumnCount(): int	Visszaadja az oszlopok számát.
+getValueAt(rowIndex: int, columnIndex: int): Object	A paraméterként kapott sornál és oszlopnál visszaadja a cella értékét.
+getColumnName(rowIndex: int): String	A paraméterként kapott sornál visszaadja az oszlopok nevét.
+getColumnClass(rowIndex: int): Class<?>	A paraméterként kapott sornál visszaadja az oszlopok osztályát.
+isCellEditable(rowIndex: int, columnIndex: int): boolean	A paraméterként kapott sornál és oszlopnál visszaadj, hogy módosítható-e a cella.
+setValueAt(o: Object, rowIndex: int, columnIndex: int):	A paraméterként kapott sornál és oszlopnál felülírja a cella értékét.

3.1.7 AutoPanel

Felelősségek

Egy saját JPanel, ami felel a kezelőfelületért és a kinézetéért. Továbbá betölti fájlból az autók listáját.

Attribútumok

-autokTable: AutokTable	Az autók abstract table modellje

Metódusok -

3.1.8 BerloPanel

Felelősségek

Egy saját JPanel, ami felel a kezelőfelületért és a kinézetéért. Továbbá betölti fájlból a bérlők listáját.

Attribútumok

-berlokTable: BerlokTable	A bérlők abstract table modellje

Metódusok -

3.1.9 AutoPanel

Felelősségek

Egy saját JPanel, ami felel a kezelőfelületért és a kinézetéért. Továbbá betölti fájlból a bérlések listáját.

Attribútumok

-berlesekTable: BerlesekTable	Az bérlések abstract table modellje

Metódusok -

3.1.10 TableFilter

Felelősségek

Egy saját JPanel, ami felel az elemek keresésért.

Attribútumok -

Metódusok

+Filter(jtFilter: JTextField, rowsorter: TableRowSorter<AbstractTableModel>	Az osztály konstruktora, ami paraméterként kapja a kereső textfieldjét és egy rowsortert, és a keresésnek megfelelőnek jeleníti meg a sorokat.

3.1.11 MainFrame

Felelősségek

A program main osztálya, ami egy saját JFrame, ami felel az alkalmazás funkcionalitásáért és kinézetéért. Ezen a framen van JTabbedPane, amin a három saját Jpanel szerepel. Továbbá felel a hitelesítésért és a fájlok mentésért bezáráskor.

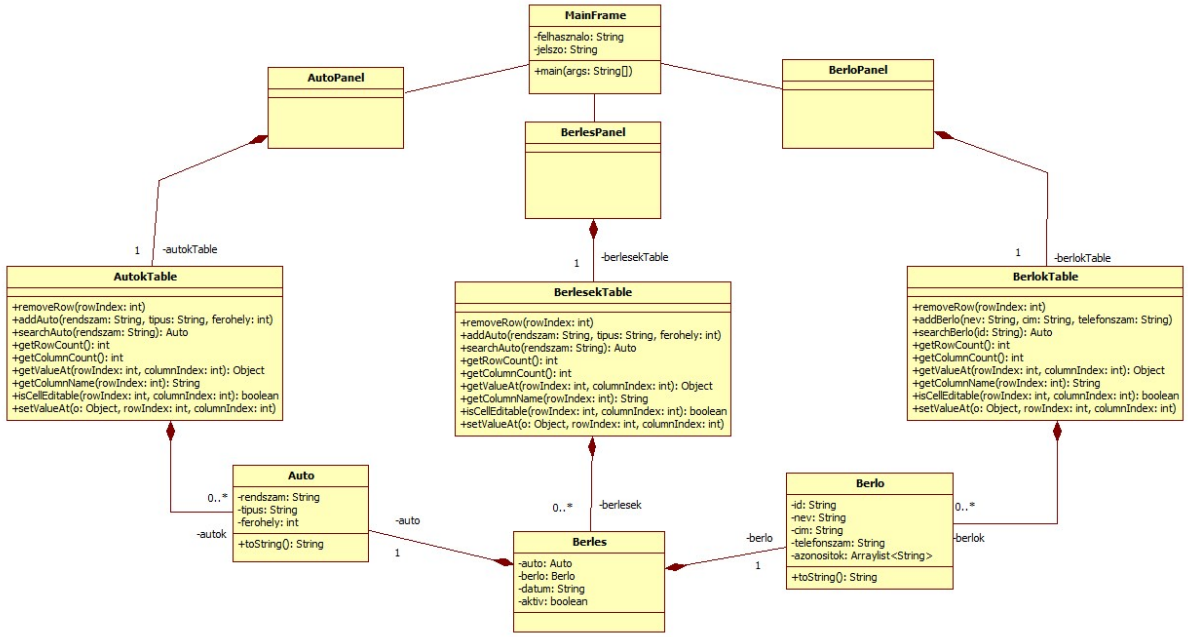
Attribútumok

-felhasznalo: String	A hitelesítéshez szükséges felhasználó név.
-jelszó: String	A hitelesítéshez szükséges jelszó.

Metódusok

+main(String[] args): static void	A program main-je.

3.2 AZ OSZTÁLYOKDIAGRAM



4 TERVEZÉSI MEGFONTOLÁSOK

A program inputjai az egyes osztályok listáját tároló fájlok (autok.txt,berlok.txt,berlesek.txt), az azonosítókat tároló (azonositok.txt) és a JTab-ok ikonjai (autoicon.png,document.png,person.png). A programból kilépéskor ezekbe a fájlokba menti a változtatást. A listák mentésére szerializálást választottam hiszen, egy objektumokból álló arraylist-et könnyedén lehet kezelni vele.

A program egy MainFrame-ből áll, amiben JTappedPane tabjai szétválasztják az autókat, bérléseket és bérlőket, amik egyenként egy saját JPanel-ek, amire azért volt szükség, mert más a formázásuk és funkcionalitásuk. Az objektumokat egy JTable-ben jelenítem meg, amihez saját AbstractTableModel osztályokat csináltam az egyes objektumoknak (Auto,Berlo,Berles), mert arraylist-ben akartam tárolni az objektumokat.

A bérlőket reprezentáló osztályban szükségem volt egyedi azonosítókra, mert a keresésnél és a szerződés készítésénél is kritikus, hogy egyértelmű legyen melyik bérlővel szeretnénk kezdeni valamit. Ezért csináltam egy azonosító generáló függvényt, ami biztosan különböző azonosítót ad. A hitelesítést egy felugró frame-ben oldottam meg. A MainFrame osztályban deklaráltam a felhasználó név és jelszó-nak a stringjeit, hogy könnyedén változtatható legyen ha szükség lenne rá.

5 FELHASZNÁLÓI ÚTMUTATÓ

A programba belépéskor az autók panelje fogad, ha más panelekre szeretne lépni, kattintson kívánt panel nevére vagy ikonjára. A funkciók lényegében ugyanúgy működnek minden panelen.

A kereséshez a ”Keresés:” után szövegdobozba írja a címszavakat, a kereső automatikusan szűri az eredményeket, nem kell gombra rákattintania. Az elemek törléshez jelölje ki a kívánt elemeket, és nyomja meg a ”Kijelölt ___ törlése” gombot. Ha görgetni szeretne a táblázatban a táblázat szélén lévő görgővel tudja megtenni. Ha elemet szeretne hozzáadni a táblázathoz, töltse ki a megfelelő mezőket a táblázat alatt, majd nyomjon a ”Felvesz” gombra.

A Bérlések panel eléréséhez meg kell adnia a megfelelő felhasználó nevet és a hozzátartozó jelszót, ha ez megtörtént nyomja meg a bejelentkezés gombot. Ha jól adta meg ezután látni fogja a bérlések panelt.