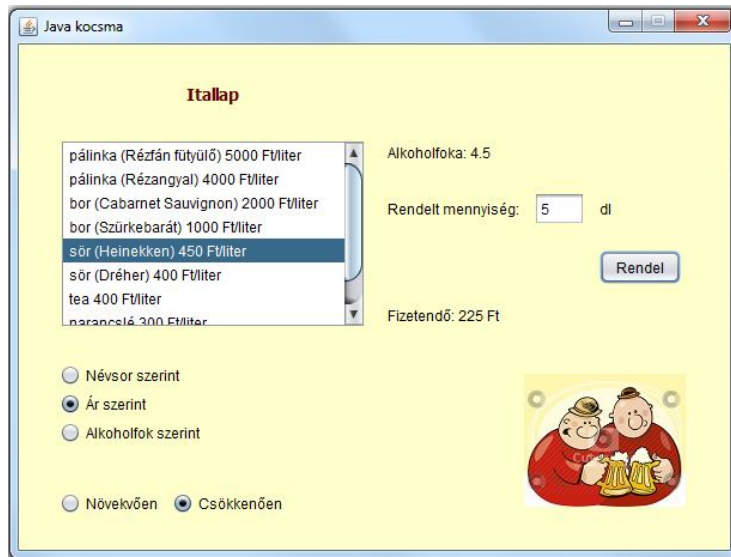


5. gyakorlat 2/c feladatának megoldása helyett a 4. gyakorlat 2/c feladata

Feladat:



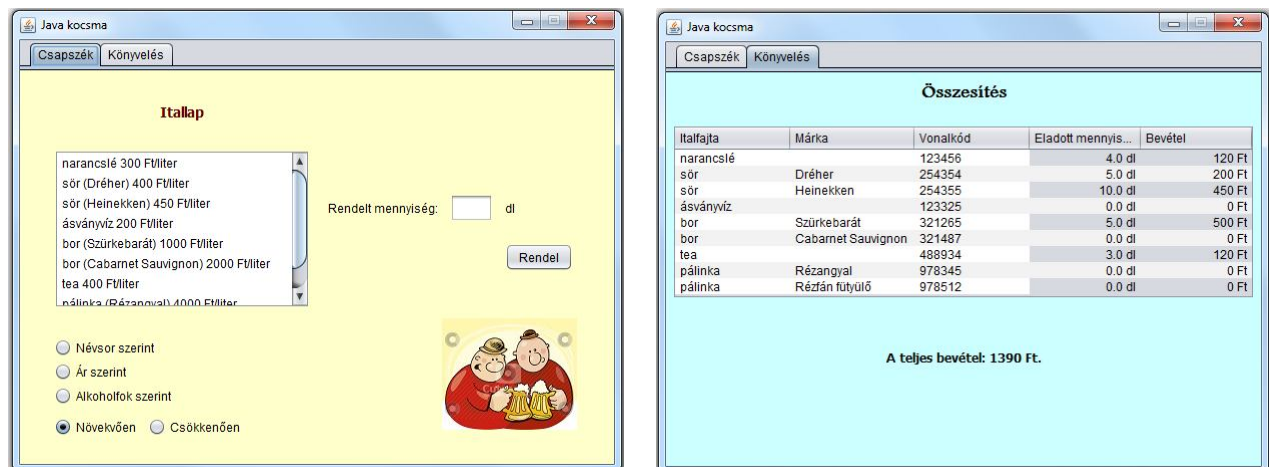
A feladat lényege:

Adatfájlban lévő adatok alapján létre kell hozni sima és alkoholos italokat, majd különböző módon rendezve megjeleníteni az itallapon.

Az itallapra kattintva mellette olvasható a kiválasztott ital alkoholfoka – persze, csak ha alkoholos italtól van szó.

Lehet rendelni, ekkor megjelenik a rendelt mennyiség ára.

A feladat további része ezt kéri:



Vagyis hirtelen meg kellene változtatni az egészet, és többregiszteres alkalmazássá alakítani. Persze, mondhatná, hogy miért nem gondoltunk időben, de sajnos az „éles” alkalmazások esetén is elég gyakori az olyasmi, hogy már csak a csaknem késznek vélt megoldás bemutatása után jön rá a megrendelő, hogy ő még mást is szeretne, vagy nem pont így gondolta. Ezért fontos az, hogy a programunk minél rugalmasabb, és minél könnyebben módosítható legyen.

Szerencsére a miénk pont ilyen. Kb. fél perc alatt alkalmassá tehetjük az előző megoldást arra, hogy több regiszteres legyen.

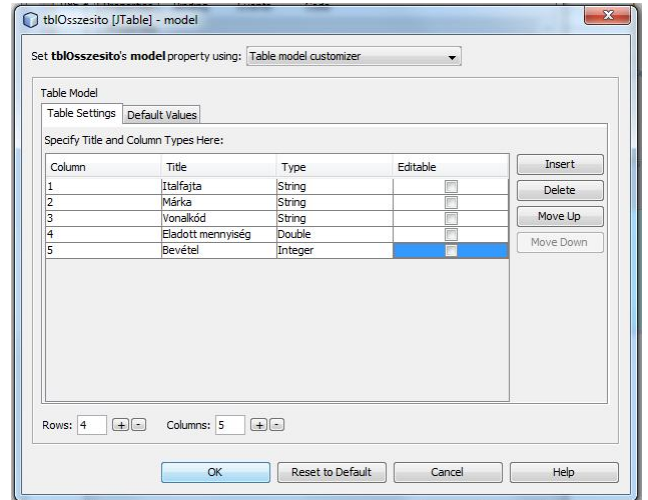
Ehhez csak ennyi kell: szedjük le a frame felületéről a `kocsmáPanel1` példányt, rakjunk rá a frame-re egy `tabbedPane` objektumot, majd erre rakjuk vissza a `kocsmáPanel` példányt.

Már is megvan az első fül. Ha erre a `tabbedPane` objektumra ráerakunk egy másik panelt, akkor megvan a második fül.

Beszéljük meg a **másik panelt** is.

Erre két `label`-t teszünk fel (a felirat, plusz a bevétel kiíratására szolgáló `label`), és egy `JTable` példányt. Ez utóbbit is ki lehet választani a palettáról, de úgy is megcsinálhatja, ahogy az előző órán tanulták. Így remélem, önállóan is meg tudná csinálni. Most azt mutatom meg, hogy hogyan lehet bánni a palettáról választott `JTable` objektummal.

A palettáról kiválasztott tábla modell tulajdonságát be tudjuk állítani (Properties/model – gomb), pl. így:



Legyen a `JTable` neve `tblOsszesito`, a bevételt tartalmazó `label`é `lblBevetel`.

A panel generált részek nélküli forrása:

```
private DefaultTableModel tablaModell ;
private int bevetel;

public OsszesitoPanel() {
    initComponents();
    // egy kis kulcsin
    tblOsszesito.setShowGrid(true);

    // Mivel a palettáról leszedett tábla modelljét már létrehoztuk akkor,
    // amikor beállítottuk a tábla fejlécét, ezért most ezt a modellt
    // akarjuk használni.
    // Sajnos a getModel() metódus TableModel típust ad vissza, ezért
    // típuskényszerítés kell (ha nem akarunk saját modell osztályt írni).
    tablaModell = (DefaultTableModel) tblOsszesito.getModel();
}
```

A `DefaultTableModel` osztály `addRow()` metódusa egy `Object[]` tömböt vár paraméterként, így a listamodellből is nagyon könnyen tudunk készíteni, ezért célszerű ezt átadni.

Már csak az a kérdés maradt hátra, hogy hol hívjuk meg ezt az `italokTablazatba()` metódust.

Nyilván ott, ahol ismerjük az italok adatait is, de elérjük ezt a metódust is.

Az ital objektumokat most a `kocsmPanel` példány listamodellje tartalmazza. Az `italokTablazatba()` metódus pedig egy `osszesitoPanel` objektumon keresztül érhető el. A két panel a frame-n szerepel együtt. No meg a `JTabbedPane` is a frame-re került. Vagyis a regiszter-fülre való kattintáskor a `kocsmPanel` ital objektumait át kellene adni az `osszesitoPanel` metódusának. Ezt az `InditoFrame` osztályban tehetjük meg.

```
public void italokTablazatba(Object[] italok) {

    // kitöröljük a tábla korábbi sorait
    int n = this.tablaModell.getRowCount();
    for (int i = n - 1; i >= 0; i--) {
        this.tablaModell.removeRow(i);
    }

    Ital ital;
    String marka;
    // létrehozzuk az új sorokat
    for (Object obj : italok) {
        ital = (Ital) obj;
        marka = (ital instanceof AlkoholosItal) ?
                ((AlkoholosItal) ital).getMarka() : "";
        Object[] tablaSor = {ital.getFajta(), marka, ital.getVonalKod(),
                ital.getOsszMennyiseg(), ital.getOsszBevetel()};
        this.tablaModell.addRow(tablaSor);

        // kiszámoljuk a bevételt
        bevetel += ital.getOsszBevetel();
    }
    lblBevetel.setText("A teljes bevétel: " + bevetel + " Ft.");
}
```

A metódusban látható példányneveket (`JTabbedPane`, `osszesitoPanel`, `kocsmPanel`) a NetBeans generálta akkor, amikor ráhúztuk az objektumokat a frame felszínére. Amikor megírjuk a metódust, figyelniük kell ezekre a generált nevekre. (De át is nevezheti őket, ha úgy szimpatikusabb.)

Még nem mutattam meg az `KocsmPanel` osztályban definiált `getItalok()` metódust. Ennyi:

```
public Object[] getItalok(){
    return italModell.toArray();
}
```

Megjegyzések:

1. Vegye észre, hogy itt is másolatot adunk át, hiszen a `toArray()` metódus hatására az `italModel` elemei átkerülnek egy `Object` típusú tömbbe.
2. Bár a módosíthatóság szempontjából elegáns megoldás az, hogy minden fülre külön saját panel osztályt írunk, nem biztos, hogy mindig élni kell vele. A nagyon összetartozó fülek esetén nem feltétlenül muszáj mindent külön panelre tenni, bár kétségtelen, hogy úgy elegánsabb.