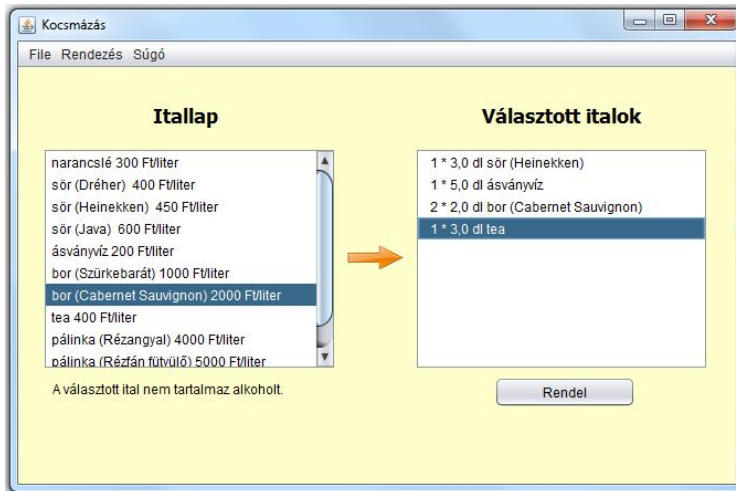


5. gyakorlat 2. feladatának megoldásrészletei

Feladat

a) Módosítsuk az eddigieket, és az egyszerűség kedvéért most minden italhoz rendeljünk egy-egy default rendelhető mennyiséget, a nyíl gomb hatására most a rendelések száma és ez a mennyiség jelenjen meg a jobboldali listafelületen.

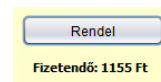


A választott italok listájára kattintva az itallap alatt jelenjen meg a kijelölt ital alkoholfoka, ha van,

A választott ital alkoholfoka: 14,0 fok

egyébként pedig az ábrán látható felirat.

A Rendel feliratú gomb hatására realizálódik a rendelés, ekkor a választott italok eltűnnek a listafelületről, és megjelenik a fizetendő érték.



Tesztelje a módosított `Ital` osztályt!

Megoldásrészletek:

Ha az előző feladatot alaposan végiggondolta és megértette, akkor ez a feladat már nem okozhat különösebb nehézséget, nem is részletezem, csak néhány ötletet adok hozzá.

Az `Ital` osztály korábbi változatába még ezek kerülnek:

```
public Ital(String fajta, String vonalKod, int literAr, double defaultDl) {
    this.fajta = fajta;
    this.vonalKod = vonalKod;
    this.literAr = literAr;
    this.defaultDl = defaultDl;
}

public int defaultAdagAr() {
    return (int) Math.round(defaultDl*literAr/10);
}

public void rendeles() {
    rendeltDb++;
}

public int rendelesAr() {
    return rendeltDb*defaultAdagAr();
}
```

```

public void fizetve() {
    osszBevetel += rendelAr();
    osszMennyisege += rendeltDb*defaultDl;
    rendeltDb = 0;
}

@Override
public String toString() {
    if(rendeltDb > 0) {
        return String.format("%3d * %3.1f dl %s",
            rendeltDb, defaultDl, fajta);
    }
    return fajta + " " + literAr + " Ft/liter";
}

```

A nyíl gomb hatása alig változik. Ennek a feladatnak a megoldásakor már akkor sem jó az az ötlet, hogy az itallapon lévő példányt rakjuk át a választott italok listájába, ha csak a külalakra akarunk figyelni, vagyis csak a megjelenő stringekre. Ugyanis a baloldali és a jobboldali listafelületen más-más alakú a kiírás, vagyis egyrészt úgy kell megírni a toString() metódust, hogy az feltételtől függően más-más eredményt adjon vissza (ld. kicsit följebb), másrészt egy objektumhoz egyszerre csak az egyik érték tartozhat, vagyis emiatt is más-más objektumpéldány szerepel a két különböző modellben.

Az ide vonatkozó kódrészlet:

```

if (!(rendeltitalModel.contains(rendeltItal))) {
    rendeltItal.rendeles();
    rendeltitalModel.addElement(rendeltItal);
}else{
    int index = rendeltitalModel.indexOf(rendeltItal);
    rendeltitalModel.get(index).rendeles();
}

```

Még egy megjegyzés: ahhoz, hogy akkor is láthassuk a változást, ha egymás után többször is ugyanazt az italt választjuk, kérnünk kell a felület frissítését (újrarajzolását) – nyilván a választott italok listáján végigfutó ciklus után:

```

lstValasztottItalok.repaint();

```

A Rendelés gomb hatása:

```

private void btnRendelActionPerformed(java.awt.event.ActionEvent evt) {
    int ossz = 0;
    for (int i = 0; i < rendeltitalModel.size(); i++) {
        ossz += rendeltitalModel.get(i).rendelesAr();
    }
    lblFizetendo.setText("Fizetendő: " + ossz + " Ft");
    rendeltitalModel.clear();
}

```

Egy lehetséges teszt:

```
public class ItalTeszt {  
  
    public ItalTeszt() {  
    }  
  
    @Before  
    public void setUp() {  
    }  
  
    @Test  
    public void italRendeles() {  
        Ital ital = new Ital("fajta", "vonalkod", 1000, 2);  
        assertTrue(ital.defaultAdagAr() == 200);  
        assertEquals(ital.getOsszBevetel(), 0);  
  
        ital.rendeles();  
        assertEquals(ital.rendelesAr(), 200);  
  
        ital.rendeles();  
        assertEquals(ital.rendelesAr(), 400);  
  
        assertEquals(ital.getOsszBevetel(), 0);  
  
        ital.fizetve();  
        assertEquals(ital.getOsszBevetel(), 400);  
        assertEquals(ital.rendelesAr(), 0);  
  
        ital.rendeles();  
        ital.fizetve();  
        assertEquals(ital.getOsszBevetel(), 600);  
    }  
}
```