

Java Swing labor

Készítette: Hartung István, BME IIT, 2015.

A feladatok megoldásához felhasználandó osztályok leírásait az alábbi URL-en találja meg:

<http://download.oracle.com/javase/8/docs/api/>

1 Caesar-kódolás

Írjon egy olyan függvényt, amely Caesar kódolást valósít meg az angol abc nagybetűivel (amennyiben a bemenet nem nagybetűs, alakítsa azzá):

<http://hu.wikipedia.org/wiki/Caesar-rejtjel>

A függvény fejléce:

```
private String caesarCode(String input, char offset)
```

A függvény egyik bemeneti paramétere legyen a kódolni kívánt String, a másik pedig az eltolódás mértéke karakterben megadva. Pl. 'K' esetén A->K kódolást csináljon. Figyeljen arra, hogy a 'Z' karakter után az 'A' következzen!

A függvény működését tesztelje is!

Segítség:

- 'Z' karakterkódja: 90
- Az angol abc 26 betűs

2 JFrame

Készítse el első ablakát (az osztály neve legyen *CaesarFrame*). Ehhez a JFrame osztályból való leszármazás szükséges. Olvassa át ehhez az osztály API-ját!

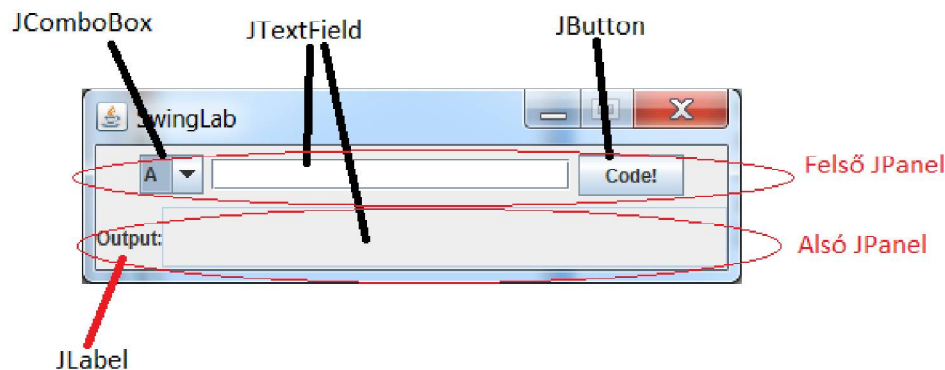
<http://download.oracle.com/javase/6/docs/api/javax/swing/JFrame.html>

- Az osztály konstruktorában gondoskodjon arról, hogy ha a felhasználó az ablak sarkában levő X-re kattint, akkor az alkalmazás be is záródjon, ne maradjon futva a háttérben! Ehhez használja a **setDefaultCloseOperation(int)** függvényt!
- Keresse meg azt a függvényt, amivel az ablak címsorában levő szöveget lehet beállítani, ezt állítsa a „SwingLab” karaktersorozatra!
- Állítsa be az ablak méretét 400x110 pixelesre és állítsa be, hogy ne lehessen átméretezni!
- Az ablak megjelenítéséhez a **main** függvényben példányosítsa az osztályt, és hívja meg azt a metódust, amely láthatóvá teszi az ablakot!

3 Elrendezés

Vizsgálja meg a Java-ban található főbb elrendezéseket (**Layout**)! Különös tekintettel az egyszerűbb **BorderLayout**, **FlowLayout** és **GridLayout** osztályokra.

<http://download.oracle.com/javase/tutorial/uiswing/layout/visual.html>



Hozza létre a fenti elrendezésű ablakot!

- A következő osztályokból érdemes az osztályban privát láthatóságú tagváltozókat létrehozni (későbbi feladatok miatt):
 - **JTextField** (a felső és alsó mező, 20 karakter szélesek), **JButton** és **JComboBox**
 - A többi elem elegendő a konstruktorban létrehozni
- A **JComboBox** létrehozása előtt egy **Object** tömböt kell feltöltenie az angol ABC nagybetűivel, és célszerű ezt átadni a **JComboBox** konstruktorának
- Emellett szükséges lehet több **JPanel** hozzáadására is az elrendezés miatt. Figyeljen az ezeken alapértelmezésben beállított **Layout**okra, valamint a **JFrame Layout**jára is!
- Állítsa be, hogy az alsó mezőt ne lehessen szerkeszteni!
- Tegye az ablakot átméretezhetővé, és próbálja is ki ezt a funkciót!

4 Listenerek

Definiáljon a *CaesarFrame* osztályon belül egy (belső) osztályt (**OkButtonActionListener**), hogy elérje a befogadó osztály attribútumait is. Ez a listener implementálja az **ActionListener** interfészt úgy, hogy a bekövetkezett esemény hatására a fölül levő szövegmezőbe beírt szöveget Caesar kódolja a **JComboBox**ban beállított karakter eltolással, és a kódolt szöveget jelenítse meg az alsó szövegmezőben!

A feladat megoldásához használja a korábban elkészített függvényt!

Segítség: a **JTextField**nek vannak **getText()** és **setText(String s)** metódusai, a **JComboBox**ban kiválasztott érték lekérdezéséhez a **getSelectedItem()** által visszaadott értéket kell karakterrészre bontani.

Ennek az osztálynak egy példányát állítsa be a **JButton** eseményfigyelőjének az ablak konstruktorában!

5 Adapterek

Valósítsa meg, hogy a felső szövegdobozban történt minden billentyűleütés esetén megtörténjen ez a kódolás!

- Valósítsa meg a feladatot úgy, hogy létrehoz egy **InputFieldKeyListener** nevű osztályt, ami a **KeyAdapter** osztályból származik le, és felülírja a megfelelő eseménykezelő függvényt! Az így létrejövő osztályt adja hozzá a bemeneti mezőhöz **KeyListener**ként!
- Milyen hibát tapasztal? Mi okozhatja ezt?

6 DocumentListener

Javítsa ki az előző feladatot, hogy helyesen működjön!

Ehhez a DocumentListener interfész megvalósítása szükséges.

Tipp: el kell kérni a textfield modelljét (Document)

<http://download.oracle.com/javase/6/docs/api/javax/swing/JTextField.html>

7 Másik irányba kódolás

Az alsó **JTextField**et állítsa át, hogy szerkeszthető legyen, és valósítsa meg, hogy amennyiben a felhasználó oda ír be szöveget, és megnyomja a „Code!” feliratú gombot, akkor dekódolódjon az oda írt szöveg!

Ehhez szükség van arra, hogy

1. tudjuk, mikor melyik textfieldben játunk utoljára,
2. legyen dekódoló függvényünk (a felelősségek szétválasztásának elve megköveteli, hogy a GUI ne tudjon a kódolás módjáról)

Tipp: használjon FocusListener-t!

<http://download.oracle.com/javase/6/docs/api/java/awt/event/FocusListener.html>