

7. Gyakorlat

legendi@inf.elte.hu

2010. március 23.

Kiegészítés

FileNotFoundException Ha nem tudod, hol keresi az IDE, vagy a Java a megadott file-t, alkalmazhatod ezt a cselt:

```
File file = new File("test.txt");
System.out.println( file.getAbsolutePath() );
```

JOptionPane Ha super != null, akkor modális, pl.:

```
JFrame frame = ...;
...
JOptionPane.showMessageDialog( frame, "Modális üzenet");
```

MVC Model-View-Controller. Egyes GUI komponensek nem direkt módon tárolják az adatokat (többféle reprezentáció), ilyenkor modelleket kell használni (pl. JProgressBar, JNumberSpinner, JTable, stb.). Példa:

```
int val = progressBar.getModel().getValue();
progressBar.getModel().setValue(++val);
```

Linux OpenJDK van rajta - használjatok Sun JVM-et, ha lehet.

static Osztály szintű, pl. adott osztály példányainak számolása:

```
class Foo {
    static int ctr = 0;
    final int idx;

    public Foo() {
        ...
        idx = ctr++;
    }
}
```

BorderLayout Használatra példa:

```
JPanel panel = new JPanel( new BorderLayout() );

panel.add( new JLabel("North"), BorderLayout.NORTH );
panel.add( new JLabel("South"), BorderLayout.SOUTH );
panel.add( new JLabel("East"), BorderLayout.EAST );
panel.add( new JLabel("West"), BorderLayout.WEST );
panel.add( new JLabel("Center"), BorderLayout.CENTER );
```

CardLayout Példa:

```
final static String APANEL = "Panel 1";
final static String BPANEL = "Panel 2";

JPanel card1 = new JPanel();
JPanel card2 = new JPanel();

CardLayout layout = new CardLayout();
cards = new JPanel( layout );
cards.add(card1, APANEL);
cards.add(card2, BPANEL);

layout.show(cards, APANEL );
layout.show(cards, BPANEL );

layout.first(cards);
layout.last(cards);
```

JRadioButton ButtonGroup használata kell, ha szeretnénk, hogy egyszerre csak választható ki közülül:

```
JRadioButton button1 = new JRadioButton("A");
JRadioButton button2 = new JRadioButton("B");

ButtonGroup group = new ButtonGroup();
group.add( button1 );
group.add( button2 );
```

JFileChooser Használatnál figyeljünk a visszatérési értékre!

```
final JFileChooser fc = new JFileChooser();
...
```

```

int ret = fc.showOpenDialog( frame );
if ( JFileChooser.APPROVE_OPTION == ret ) {
    ...
}

```

Feladat

Készítsünk egy egyszerű kvíz-alkalmazást, amely az alábbi típusú kérdéseket tartalmazhatja:

- Eldöntendő kérdés, pl.:

Hány fia volt Ádámnak a Biblia szerint?

- ☐ 2
- ☐ 3
- ☒ 33

- Kiválasztós, pl.:

Mely országok határosak Magyarországgal?

- ☒ Szerbia
- ☒ Szlovénia
- ☐ Japán

Minden kérdésben 3-3 lehetőség van. Ezeket fájlból olvassuk, inicializálás során. A formátuma a következő:

```

C:Hany versszakbol all a himnusz?:4:6:8:false:false:true
S:Mely szamok primek?:1:2:7:false:true:true
...

```

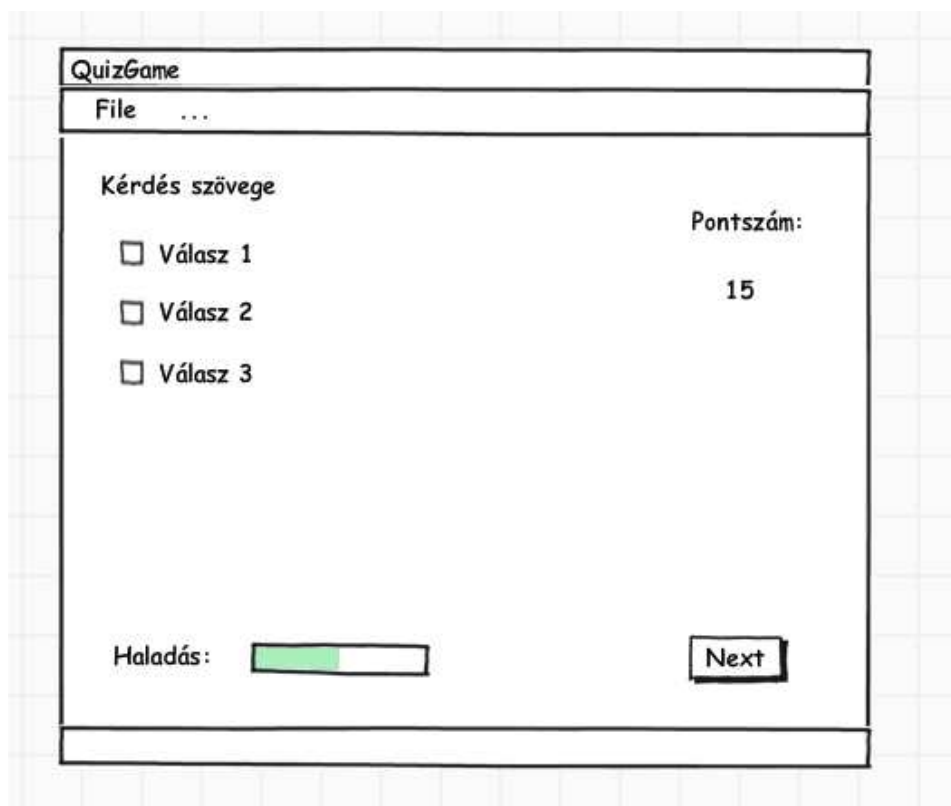
Az inputban : karakterrel elválasztva 8 mező található, ezek jelentései a következők:

- Az első karakter megmondja, hogy milyen típusú kérdés adatait tartalmazza az adott sor. C = eldöntendő, S = kiválasztós kérdés
- A kérdés szövege
- A lehetséges 3 válasz
- A válaszok helyessége

A játék képernyőterve az 1 ábrán található.

A program a következő funkciókat biztosítsa:

- Egyszerre 5 kérdést adjon



1. ábra. Képernyőterv

- Lehesse menüből kilépni, valamint új játékot kezdeni (ezekre kérjen megerősítést a program!)
- A program jelezze a teszt kitöltöttségét egy JProgressBar segítségével
- Pontozás: minden jól megválaszolt eldöntendő kérdés 10 pont. Minden eltalált kiválasztós kérdés 5 pont.
- A játék végén írjon ki egy gratuláló üzenetet, amely tartalmazza, hány pontot ért el a felhasználó

Példakérdések: <http://people.inf.elte.hu/legendi/java/res/questions.txt>

További lehetőségek

1. Játékot lehessen elmenteni, betölteni! (memento)
2. Egyszer lehessen segítséget kérni egy '?' gombbal! Ha használtuk, tilt-suk le a gombot.

3. A végén lehessen megnézni az eredményeket. Itt színes labellel jelezzük, hogy sikerült az adott kérdést teljesíteni (zöld - OK, piros - Fail, sárga - skip).