



Programozási technológia I.

Grafikus felhasználói felületek, eseménykezelés

Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Dr. Szendrei Rudolf
Informatikai Kar
Eötvös Loránd Tudományegyetem



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Tartalom

1 Menü

2 Egér

3 Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum, eseményosztály

Eseménykezelő objektum, -osztály

Eseménykezelő függvény

Eseménykezelő kiváltása és kezelése

Események és eseménykezelők összekapcsolása



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

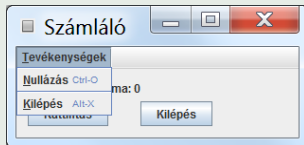
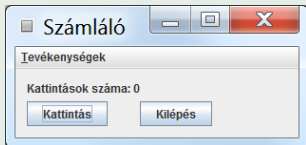
Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Menü

Menü

Egészítsük ki a már korábban elkészített Számláló alkalmazásunkat úgy, hogy az tartalmazzon egy menüt, amiben nullázhatjuk a számláló értékét, illetve kiléphetünk a programból, valamint külön címkén jelenítsük meg a kattintások számát.





Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

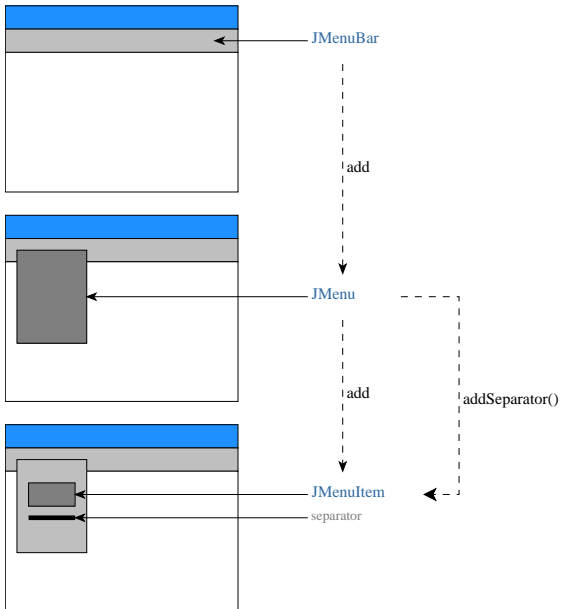
Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása





Menü

Menüsor komponensek műveletei

- `JMenuBar`
 - Általában a program ablakának tetején helyezkedik el, elemei a menük, beállítani egy `JFrame`-en belül a `setJMenuBar()`-al lehet
- `JMenu`
 - konstruktor: paraméter a név
 - `setMnemonic`: paraméter a kiválasztó billentyű
- `JMenuItem`
 - konstruktor: paramétere egy `AbstractAction`-ből származtatott objektum (vagy név, ikon; de ekkor eseménykezelőt hozzá kell venni)
 - `setMnemonic`: paraméter a menün belüli kiválasztó billentyű
 - `setAccelerator`: a gyorsbillentyű kódja



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

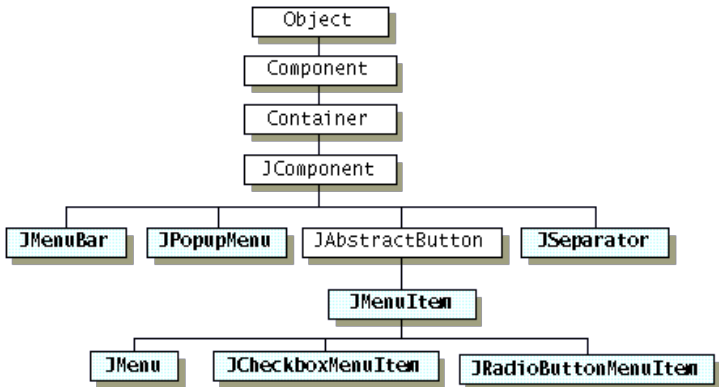
Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása





Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Menü - Megvalósítás

```
JMenuBar menuBar = new JMenuBar();
```

```
JMenu menuTevékenység = new JMenu("Tevékenységek");
menuTevékenység.setMnemonic('T');
```

```
JMenuItem menuNullázás = new JMenuItem(new AbstractAction()
{
```

```
    @Override
    public void actionPerformed(ActionEvent e) { ... }
```

```
});
```

```
menuNullázás.setText("Nullázás");
```

```
menuNullázás.setMnemonic('N');
```

```
menuNullázás.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
                                                    KeyEvent.CTRL_MASK));
```

```
JMenuItem menuKilépés = new JMenuItem(new AbstractAction()
{
```

```
    @Override
    public void actionPerformed(ActionEvent e) { ... }
```

```
});
```

```
menuKilépés.setText("Kilépés");
```

```
menuKilépés.setMnemonic('K');
```

```
menuKilépés.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X,
                                                    KeyEvent.ALT_MASK));
```

```
menuTevékenység.add(menuNullázás);
```

```
menuTevékenység.addSeparator();
```

```
menuTevékenység.add(menuKilépés);
```

```
menuBar.add(menuTevékenység);
```

```
setJMenuBar(menuBar);
```

```
...
```

```
setVisible(true);
```



Menü

Menüsor - Opcionális menüelem

- Változtassuk meg a programot úgy, hogy csak akkor lehessen a menüből nullázni, ha a kattintások száma nem nulla!
- Ehhez a `JMenuItem.setEnabled()` műveletét használhatjuk
- Engedélyezés:
 - Kezdetben letiltjuk a menüpontot (a létrehozásakor)
 - Kattintás esetén engedélyezzük a menüpontot (kattintás eseménykezelőjében)
 - Nullázás esetén letiltjuk a menüpontot (nullázás eseménykezelőjében)



Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Egér

Egér kezelése

A felület elemei közül több komponens bizonyos formában kezeli az egeret (pl. gombok, listák a kattintást, görgetést a görgetősávok). Ha eltérő kezelés, illetve új esemény figyelése szükséges, akkor egéresemény figyelőt kell a komponenshez rendelni:

- `addMouseListener`
paramétere egy `MouseListener` interfészt megvalósító objektum, amely kezeli a az egérgomb (lenyomás, felengedés, kattintás) és területet váltó (belépés, elhagyás) eseményeket
- `addMouseMotionListener`
paramétere egy `MouseMotionListener` interfészt megvalósító objektum, amely kezeli az egér húzását
- `addMouseWheelListener`
paramétere egy `MouseWheelListener` interfészt megvalósító objektum, amely kezeli az egér görgető gombjának eseményeit



Egér

MouseAdapter osztály

- Mindhárom interfészt megvalósítja (`MouseListener`, `MouseMotionListener`, `MouseWheelListener`)
- Ha nem akarunk mindenhez megvalósítást írni, akkor célszerű ebből származtatni, illetve ha több interfész műveleteire van szükségünk
- Görgetés esetén egy `MouseWheelEvent` objektumot kapunk, amelytől lekérdehető például a görgetés iránya (+/- 1-et kapunk). Ez a művelet minden egyes elforgatás hatására meghívódik, azaz egy 5 „kattanásos” lefele görgetésnél 5x hívódik meg +1 értékkel!

```
public class ÜresKeret extends JFrame{
    private final MouseAdapter egérAdapter = new MouseAdapter() {...}
    public ÜresKeret () {
        ...
        addMouseListener(egérAdapter);
        addMouseMotionListener(egérAdapter);
        addMouseWheelListener(egérAdapter);
        ...
    }
}
```



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseménykezelés alapjai

- Az eseménykezeléssel működő programok alapötlete az, hogy a program futása közben események hatására mennek végbe bizonyos folyamatok
- Események váltódnak ki, melyekre az eseménykezelők reagálnak
- Egy esemény a program bármely részében kiváltódhat, az erre reagáló eseménykezelő pedig tipikusan a program egy teljesen más részében fut le
- Objektumorientált programozási nyelvben objektumok és függvényeik segítségével valósul meg



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseményobjektum, eseményosztály

- Minden kiváltott eseményhez tartozik egy eseményobjektum
- Az eseményobjektum hordozza az esemény tulajdonságait – az objektum saját változói segítségével
- Rendelkezik egy referenciával arról is, hogy hol váltódott ki az esemény – másnéven forrás (source)
- A hasonló típusú eseményekhez természetesen létrehozható egy megfelelő eseményosztály
- Java-ban az események őszosztálya a `java.util.EventObject` osztály, tartalmazza a forrás kezelését



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseménykezelő objektum, -osztály

- Alkalmas egy adott típusú esemény kezelésére, azaz reagálni arra, ha az esemény kiváltódik
- Rendelkezik egy vagy több eseménykezelő függvénnyel, ahol mind 1-1 esemény kezelésére szolgál
- A függvény(ek) megkapja/megkapják a kiváltott eseményobjektumot
- Az eseményobjektum forrása és tulajdonságai alapján – és persze a saját feladatának megfelelően – végzi el az esemény kezelését
- A hasonló típusú eseménykezelő objektumok számára természetesen létrehozható egy megfelelő eseménykezelő osztály
- Java-ban az eseménykezelők a `java.util.EventListener` interfészből származnak, tipikusan maguk is csak interfészek, hiszen az eseménykezelő függvények konkrét működése általában nem ismert



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseménykezelő függvény

- Egy eseménykezelő függvény egy esemény kezelésére szolgál, tartalmazza azt a tevékenységet, amelyet az adott eseményre reagálásképp végre kell hajtani
- Paraméterként megkapja a kiváltott eseményt, amelyre éppen reagálni kell, azaz amelyet kezelni kell
- Működése közben kiválthat más eseményeket is



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseménykezelő kiváltása és kezelése

- Egy esemény kiváltásakor ismerni kell az eseményt kezelő objektumokat
- Az esemény kiváltásának első lépése az eseményobjektum megkonstruálása (forrásként rendszerint beállítható a `this` referencia)
- Második lépésként az eseménykezelő objektumok eseménykezelő függvényeinek meghívása történik az eseményobjektum átadásával
- Az esemény kezelése az eseménykezelő függvényben történik, tehát ezen függvények törzsét, működését kell megadni



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Események és eseménykezelők összekapcsolása

- Ahhoz, hogy egy esemény kiváltásának pillanatában ismerjük az eseménykezelőket, össze kell kapcsolni azokat egymással
- Tevékenységek:
 - regisztráljuk az eseménykezelőt az eseménynél
 - az eseménykezelő feliratkozik az eseményre
 - az eseménykezelőt hozzárendeljük az eseményhez



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

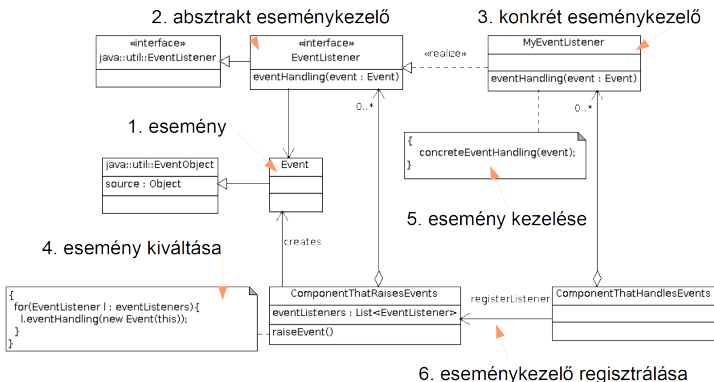
Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény

Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása





Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Események és eseménykezelők összekapcsolása

- Bizonyos fogalomkörökben szokás az eseménykezelő függvényeket **Callback** függvényeknek is nevezni
- Az események logikailag két csoportba sorolhatók:
 - felhasználói interakció eredménye
 - rendszerszintű „értesítés”
- A felhasználói interakciók kezelésére már láttunk példát a kattintások számlálása, ablak bezárása, egér mozgatása stb. során
- Ugyanakkor az ablak bezárását nem csak a felhasználó kezdeményezheti, hanem például rendszer frissítéskor az operációs rendszer is küldhet ilyen értesítést, mielőtt automatikusan újraindul



Menü

Egér

Eseménykezelés

Eseménykezelés alapjai

Eseményobjektum,
eseményosztály

Eseménykezelő objektum,
-osztály

Eseménykezelő függvény
Eseménykezelő kiváltása
és kezelése

Események és
eseménykezelők
összekapcsolása

Eseménykezelés

Eseménykezelés felhasználása

- Az események felhasználásának egyik módja, hogy felhasználói interakciót fogadunk vele
- Az eseménykezelés másik nagy területe a blokkoló függvény hívások helyett úgy nevezett aszinkron hívások megvalósítása eseménykezelővel
- Az aszinkron hívás esetében nem várjuk meg az eredményt, hanem rögtön folytatjuk a munkát
- A hívás paramétereként megadjuk azt az eseménykezelőt, amelyet majd a rendszer meghívhat, ha végzett a kért művelettel
- Ez az aszinkron hatás mindkét irányban működhet, azaz az esemény kiváltója is nyugodtan folytathatja a munkáját, nem kell megvárnia az esemény lekezelését
- A módszer alkalmazásánál körültekintően kell eljárni az objektumok életciklusa miatt