



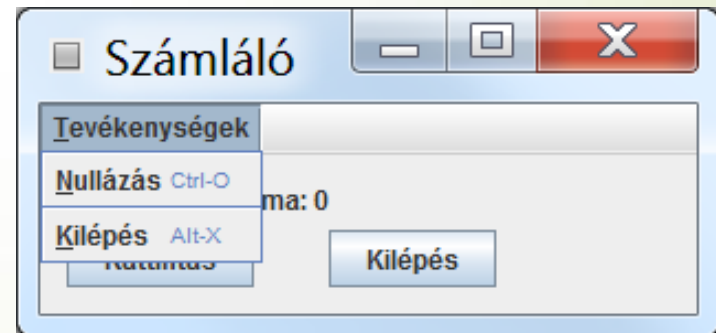
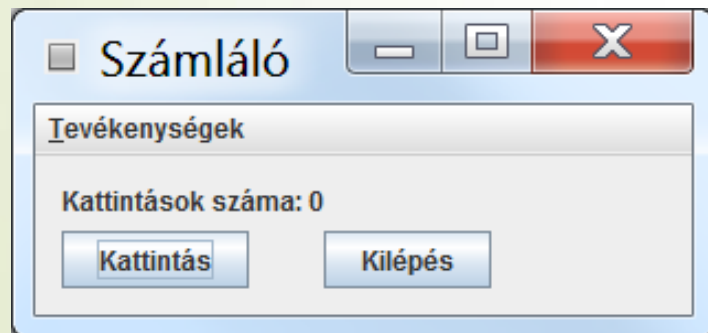
Programozási technológia

Grafikus felhasználói felület –
Eseménykezelés

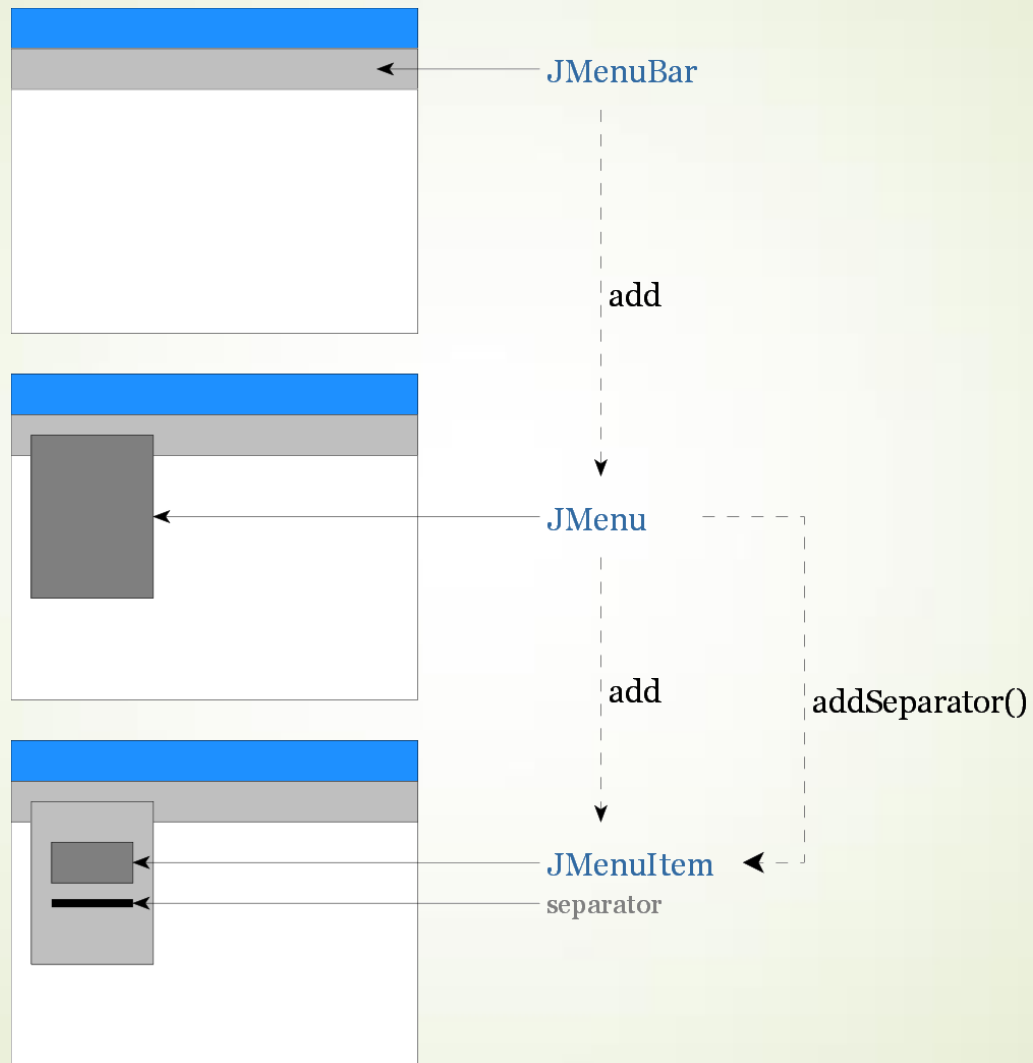
Dr. Szendrei Rudolf
ELTE Informatikai Kar
2018.

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.



Menüsor felépítése



Menüsor komponensei

➤ `JMenuBar`

- Általában a program ablakának tetején helyezkedik el, elemei a menük, beállítani egy `JFrame`-en belül a `setJMenuBar()`-al lehet

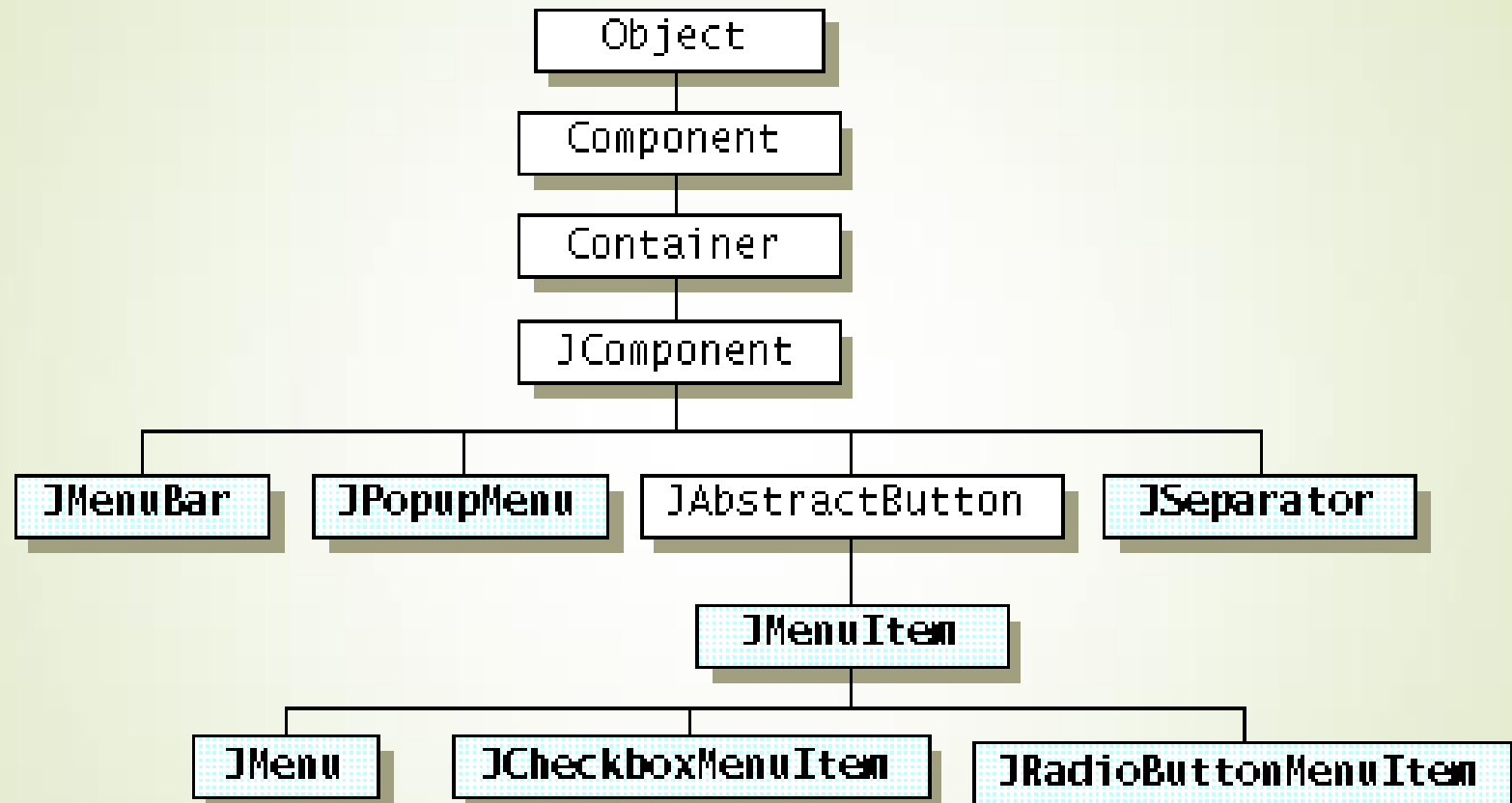
➤ `JMenu`

- 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 komponensek hierarchiája



Menü megvalósítása

```
JMenuBar menuBar = new JMenuBar();
JMenu menuFile = new JMenu("Fájl");
menuFile.setMnemonic('F');
JMenuItem menuExit = new JMenuItem(new AbstractAction() {
    @Override
    public void actionPerformed(ActionEvent e){ ... }
});
menuExit.setText("Kilépés");
menuExit.setMnemonic('K');
menuExit.setAccelerator(
    KeyStroke.getKeyStroke(KeyEvent.VK_X, KeyEvent.ALT_MASK));
menuFile.add(menuExit);
menuBar.add(menuFile);
setJMenuBar(menuBar);
```

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)

Egér események 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

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.
- ▶ Görgetésnél `MouseWheelEvent` objektumot kapunk, amelytől lekérdehető pl. a görgetés iránya (minden egyes forgatásnál meghívódik, azaz 5 „kattanásos” görgetésnél 5x hívódik meg).

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

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
- Objektorientált programozási nyelvben objektumok és függvényeik segítségével valósul meg

Eseménykezelés

Eseményosztály, eseményobjektum

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

Eseménykezelés

Eseménykezelő osztály

- Az eseménykezelő 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.
- Ezek a kapott eseményobjektum és annak forrása alapján a feladatuknak megfelelően elvégzik az esemény kezelését (maguk is válthatnak ki természetesen további eseményeket).
- A hasonló típusú eseménykezelő objektumok számára természetesen létrehozható egy megfelelő eseménykezelő osztály.
- Javában az eseménykezelők a `java.util.EventListener` interfészből származnak, (tipikusan csak interfészek, hiszen az eseménykezelő függvények működése általában nem ismert)

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.

Eseménykezelés

Esemény kiváltása és kezelése

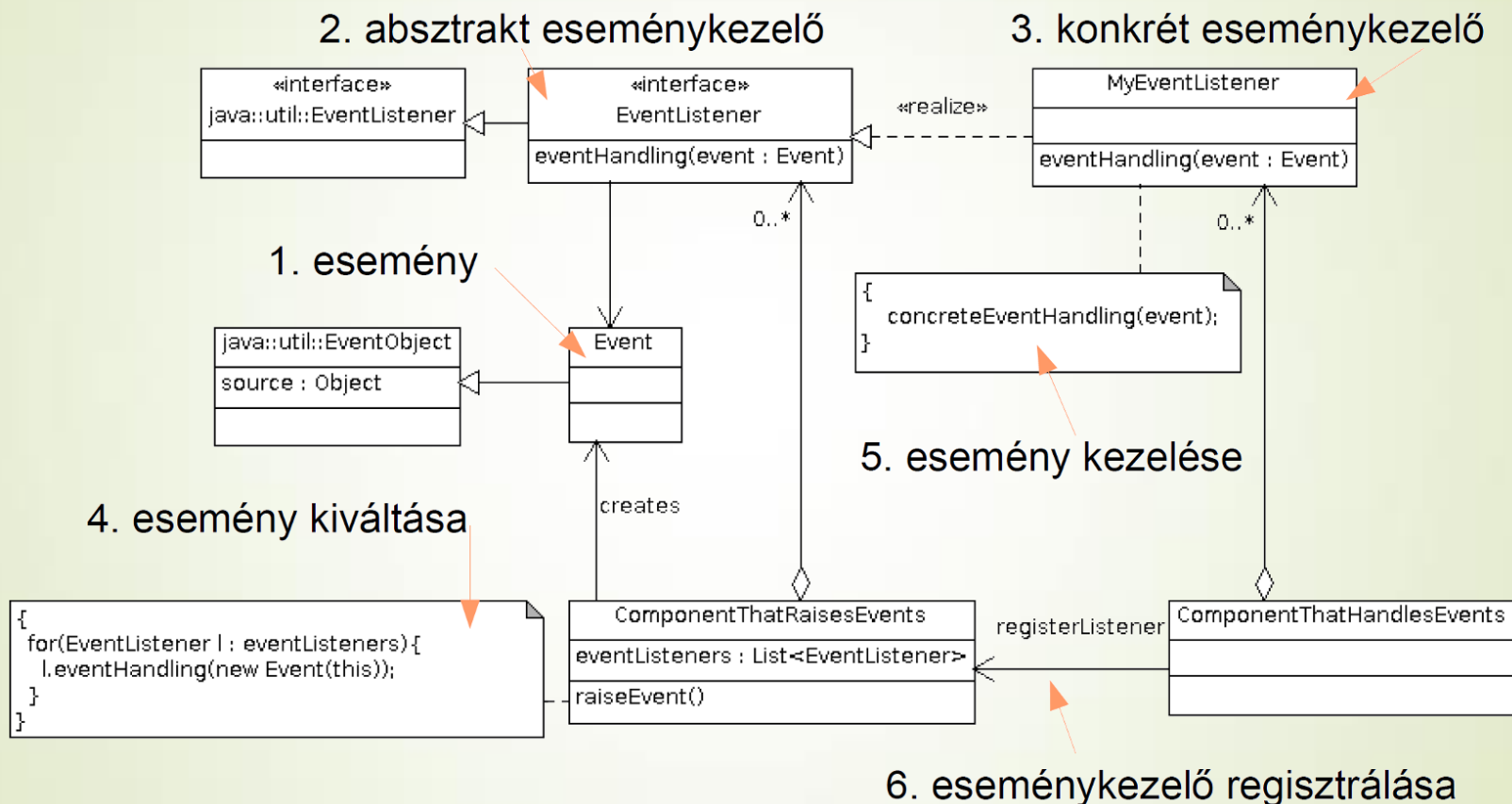
- ▶ Egy esemény kiváltásakor ismerni kell az eseményt kezelő objektumokat.
- ▶ Az esemény kiváltásának első lépése az eseményobjektum létrehozása (forrása általában a `this` referencia).
- ▶ Második lépésként az eseménykezelő objektumok eseménykezelő függvényeinek meghívása történik az eseményobjektum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

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

Eseménykezelés



Eseménykezelés

- 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

Eseménykezelés felhasználása

- Az eseménykezelés felhasználásának egyik módja a felhasználói interakciók feldolgozása.
- Alkalmas emellett a blokkoló függvény hívások kiváltására. Helyettük úgynevezett aszinkron hívásokat valósíthatunk meg az eseménykezelést felhasználva.
- 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 élelciklusa miatt.