



Programozási technológia I.

Párbeszédablakok, vezérlő elemek

Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúsztató

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Dr. Szendrei Rudolf
Informatikai Kar
Eötvös Loránd Tudományegyetem



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Tartalom

1 Párbeszédablakok, vezérlő elemek

OK és Mégsem gombot tartalmazó párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

2 Az alkalmazás kerete



Párbeszédablakok, vezérlő elemek

Bevezető

- Cél: párbeszédablakok használata, vezérlőelemek megismerése
- Az elkészítendő programban színekkel dolgozunk majd:
 - Nyilvántartjuk a színt és annak nevét is.
 - A konstruktorral hozhatjuk létre az adott nevű színt.
 - Lekérdezhetjük a szín nevét, magát a színt, és megadjuk a kiírási formátumot is (`toString`), ami a szín neve.
 - A `toString` művelet akkor lehet hasznos, ha listákat, vagy comboboxokat akarunk ilyen elemekkel feltölteni, és nem adunk meg megjelenítési előírást. Ilyenkor ezt a műveletet használja a rendszer a megjelenítendő információ meghatározásához.



Bevezető

- Az egyes vezérlő elemek használatát külön-külön párbeszédablakokban vizsgáljuk meg.
- Modális párbeszédablakokkal foglalkozunk.
- A legtöbb párbeszédablak tartalmazza az OK és a Mégsem nyomógombokat.
- Ennek megfelelően célszerűnek tűnik egy olyan osztály létrehozása, amely ezt a folyamatot támogatja a gombok létrehozásával és az alapvető eseménykezelés elvégzésével.



Párbeszédablakok – OKCancelDialog

OK és Mégsem gombot tartalmazó párbeszédablakok

Funkcionalitás:

- Az **OK** gomb megnyomása esetén ellenőrizni kell (illetve lehet), hogy az ablak tartalma megfelelő-e, és ha a válasz igen, akkor le kell zárni az ablakot.
- A **Mégsem** gomb megnyomásakor esetlegesen vissza kell állítani a kezdeti értékeket, és be kell zárni az ablakot.

(Ezek az eseménykezelők feladatai.)



Párbeszédablakok – OKCancelDialog

OK és Mégsem gombot tartalmazó párbeszédablakok

- Az eseménykezelés miatt be kell vezetnünk két absztrakt műveletet:
 - Az **OK** gomb megnyomásakor szükséges ellenőrzések elvégzése és az ellenőrzés eredményének megadása.
 - A **Mégsem** gomb aktivizálásakor szükséges teendők végrehajtása.
- További művelet:
 - A bezárást okozó gomb azonosítása.
- Az azonosításhoz szükséges két konstans, amelyek megfelelnek az **OK**, illetve a **Mégsem** gomboknak.
- Az absztrakt műveletek miatt a létrehozott `OKCancelDialog` osztály absztrakt lesz. Ebből kell származtatni a konkrét esetnek megfelelő párbeszédablakot.



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Párbeszédablakok – OKCancelDialog

Származtatás az OKCancelDialog osztályból

- Meg kell határozni (implementálni) a két absztrakt műveletet.
- A párbeszédablakot fel kell tölteni a megfelelő vezérlő elemekkel.
- A vezérlő elemekhez hozzá kell venni a két nyomógombot, pontosabban el kell helyeznünk a gombokat tartalmazó panelt.



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Párbeszédablakok – OKCancelDialog

OKCancelDialog osztály

- Az absztrakt osztályt a `JDialog` osztályból származtatjuk.
- Az osztály tartalmazza a kívülről használható osztályszintű konstansokat, illetve a származtatott osztályokban használható panelt, és gombokat.
- Események kezelése:
 - az absztrakt műveletek meghívása,
 - gombkód beállítása,
 - eltüntetés.



OKCancelDialog osztály – folyt.

- Konstruktor:
 - Minden esetben megadjuk a keretet, ahonnan a párbeszédablakot aktivizáltuk, és az ablak címét. (Ezek segítségével lehet egy modális párbeszédablakot létrehozni.)
 - A konstruktorban létrehozzuk a paramétereknek megfelelő modális párbeszédablakot, és beállítjuk az alapértelmezett bezárási műveletet. (Az alapértelmezett az ablak elrejtése, ezért ez most tulajdonképpen felesleges.)
 - Ezután létrehozzuk a megfelelően paraméterezett gombokat, amelyeket egy panelban helyezünk el.
 - Az `OK` gombot állítjuk be alapértelmezett nyomógombként. (`Enter` billentyűvel aktivizálható.)
 - A `Mégsem` gombhoz hozzárendeljük az `Escape` billentyűt.



Csúszka

- A `SliderDlg` egy csúszkát tartalmazó párbeszédablak.
- A csúszka aktuális értéke is jelenjen meg a csúszka fölött.
- A párbeszédablak megadásakor beállítható:
 - a csúszka intervalluma,
 - az aktuális érték,
 - és a fő értékek (a fő értékeket az értékkel és hosszabb vonallal jelöli a csúszka).
- A normál érték minden esetben 1.



Párbeszédablakok – SliderDlg

Csúszka

Műveletek:

- Az `OK` gomb mindig elfogadható (`processOK`).
- A `Mégsem` esetén semmit nem kell tenni (`processCancel`).
- Új műveletek:
 - A csúszka értékének lekérdezése a párbeszédablak bezárása után (`getValue`).
 - A csúszka értékének állítása (`setValue`) (megjelenítés előtt).



Csúszka

Állapotváltozás:

- Az érték megjelenítése miatt figyelni kell, hogy a csúszka értéke megváltozik-e.
- Ezért az osztály megvalósítja a `ChangeListener` interfészt.
- Ehhez a `stateChanged` műveletet kell implementálnunk.
 - a csúszka értékének lekérdezése,
 - a címke feliratának módosítása.
- Ahhoz, hogy a művelet meghívásra kerüljön, az osztályt fel kell vennünk a csúszka állapotváltozásaira figyelők közé (a csúszka `addChangeListener` művelete).



Párbeszédablakok – ButtonDlg

Gombok

- A `ButtonDlg` párbeszédablak checkbox és rádiógomb típusú gombokat tartalmaz.
- A párbeszédablak segítségével határozhatjuk meg a szövegmező háttérszínét.
 - Ha a checkbox-ot bekapcsoljuk a háttér a szövegszín inverze.
 - Ha a checkbox-ot kikapcsoljuk, akkor a hátteret a kiválasztott rádiógomb határozza meg.
- A rádiógombok megjelenítése speciális:
 - a szöveg az adott szín neve,
 - az ikon a színnel kitöltött kör, amelyben fekete pont jelzi a kiválasztottságot (fekete szín esetén a pont fehér).



Rádiógombok speciális megjelenítése

- A megjelenítési módot külön meg kell adnunk a gomb létrehozásakor az ikon, illetve a kiválasztottsági ikon specifikálásával.
- Erre szolgál a `gomb` függvény.
 - Ebben létrehozunk két képet (`BufferedImage`), amelyekben megrajzoljuk a kívánt ábrát, és a képekből létrehozuk az ikonokat.
 - A rajzoláshoz szükséges a kép grafikus eszközkapcsolat leírója, ami a `getGraphics` függvény ad meg.
 - A rajolás ezután értelemszerű.



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Párbeszédablakok – ButtonDlg

Rádiógombok kezelése

- A létrehozott rádiógombokat egy csoportba kell foglalnunk.
- A rádiógombokat egy kerettel ellátott panelra kell helyoznünk.
- A gombokat nyilvántartjuk (*színgombok*) azért, hogy le tudjuk majd kérdezni a kiválasztottságukat.



Műveletek

- Az **OK** gomb lenyomása akkor fogadható el, ha van kiválasztott háttérszín.
 - Ez lehet a szövegszín inverze, vagy ha ez nincs kijelölve, akkor
 - egy színt kellett kiválasztani.
- A **Mégsem** gomb lenyomásakor nincs teendő.
- Szükséges az értékek lekérdezése
 - inverzmód,
 - szín
- Továbbá szükséges a checkbox értékének megadása (`setValue`)



Párbeszédablakok – ListDlg

Lista

- A `ListDlg` párbeszédablak egy listában tartalmazza a választható szövegszíneket.
- A listában nem csak a szín nevét, hanem egy színnel kitöltött téglalapot is megjelenítünk.
 - Egy speciális megjelenítési forma szükséges (`elemforma`), amit a `setCellRenderer` művelettel rendelhetünk a listához.
 - (Ha csak szöveget akarunk a listában, erre nincs szükség, az elemek `toString` művelete szerinti szöveg jelenik meg.)
- Ha azt akarjuk, hogy a lista elemén duplán kattintva az elemet válasszuk ki és zárjuk be az ablakot, akkor a listához fel kell vennünk egy egér esemény figyelőt, amelyben a dupla kattintást figyeljük.
(Az esemény megegyezik az OK gomb megnyomásával.)
Az OK gomb megnyomása elfogadható, ha van kiválasztott elem a listában.



Párbeszédablakok – ComboDlg

Combobox

A `ComboDlg` párbeszédablak egy szöveget adhatunk meg combobox segítségével.

- A combobox előre definiált értékei a színek nevei.
- A neveket most a szerkeszthetőség (pontosabban a kiválaszthatóság miatt) `String` objektumokként helyezzük el a comboboxban az `addItem` művelettel.
- A combobox alapértelmezésben nem engedi az elemek szerkesztését. Ezt állíthatjuk a `setEditable` művelettel.
- A szerkesztett elemet a sorszerkesztő `getItem` műveletével kaphatjuk meg.
- A sorszerkesztőt a `getEditor` függvény adja meg.



Párbeszédablakok – ComboListDlg

Kombó lista

- A combobox alapértelmezett viselkedésekor egy listából választhatunk elemet.
- A `ComboListDlg` dialógus legördülő listájában a színeket jelentjük meg, a nevük mellett a megfelelő színű körlemezzel.
 - Az ikonok megjelenítése a listához hasonlóan érhető el.
 - Most a `setRenderer` művelettel lehet a megfelelő megjelenítőt a kombó listához rendelni.
 - (Ha csak szöveg kell, akkor erre nincs szükség, az elemek `toString` művelete adja a szöveges információt.)
- A kiválasztott elemet a `getSelected` függvénnyel kaphatjuk meg.
- Alapértelmezettként a lista első (nulla indexű) eleme kiválasztott. Így mindig lesz kiválasztott elem, azaz az OK gomb mindig elfogadható.



Párbeszédablakok – EditDlg

Sorszerkesztő

- Egysoros szöveg bevitelét támogatja az `EditDlg` párbeszédablak.
- Ebben egy `JTextField` objektummal valósítjuk meg a szerkesztőt.
 - A szerkesztett szöveget a `getText` művelettel kérdezhetjük le.
 - A kezdeti szöveget a `setText` művelettel lehet beállítani, mi ezt nem használjuk, így üres szöveg lesz kezdetben a tartalom, a későbbiekben pedig a párbeszédablak megelőző bezárásakor tartalmazott szöveg.
 - A sorszerkesztőt szokásos módon használhatjuk szerkesztésre, értelmezettek a vágólapot használó műveletek: kivágás, másolás, beillesztés.



Párbeszédablakok – SpinnerDlg

Spinner

- A `SpinnerDlg` párbeszédablakban egy spinner segítségével változtathatjuk meg ugyanazt az értéket, mint a csúszkával.
 - A spinner létrehozásakor megadhatunk egy modellt, ami leírja a spinner értéktartományát, aktuális értékét, és a rákövetkezés módját.
 - Ha nem adunk meg modellt, egész számokat enged meg a tartomány korlátozása nélkül.
 - Meg lehet adni, egy listát, illetve tömböt (`SpinnerListModel`), ekkor az értékek értelemszerűek.
 - Választható számokat tartalmazó modell (`SpinnerNumberModel`) a megfelelő beállításokkal.
- Nekünk most ez utóbbira van szükségünk.
- Ha a szerkesztőbe érvénytelen értéket írunk, akkor az `Enter` billentyű lenyomásakor nem fogadja azt el. Ha az `OK` gombot nyomjuk meg, az utolsó érvényes értéket veszi figyelembe!



Párbeszédablakok – TáblázatDlg

Táblázat

- A `TáblázatDlg` párbeszédablakban egy táblázatban jelenítjük meg a színek jellemzőit: a nevet, és a komponens (rgb) értékeket.
 - Egy táblázat megjelenítésére szolgál a `JTable` osztály.
 - A táblázat tartalmát egy modell segítségével adhatjuk meg, ami az adatokat szolgáltatja.
 - Ezt legegyszerűbben az `AbstractTableModel` absztrakt osztály felhasználásával tehetjük meg.
 - A táblázat oszlopai egyenlő szélességűek. Ha ettől el szeretnénk térni, akkor a táblázat oszlop modelljét kell módosítanunk egy `DefaultTableColumnModel` osztályból származtatott osztály példányával (mi most ettől eltekintünk).



Táblázat modell

- Az `AbstractTableModel` példányosítása (táblamodell) során meg kell adnunk a következő műveletek jelentését:
 - `getRowCount` a táblázat sorainak száma,
 - `getColumnCount` a táblázat oszlopainak száma,
 - `getValueAt` a táblázat adott sorának adott oszlopában lévő értéke.



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Párbeszédablakok – TáblázatDlg

Táblázatok tulajdonságai

- Ha az oszlopok neveit is meg akarjuk adni, akkor a `getColumnName` függvényt kell felüldefiniálnunk.
- Ha görgetni akarjuk a táblázatot, akkor azt egy `JScrollPane` objektumba kell ágyaznunk. (Ha ezt nem tesszük meg, akkor külön gondoskodni kell az oszlopcímek kiíratásáról.)
- A táblázatok tartalma változtatható (mi most csak megjelenítést használtunk).
 - (`isCellEditable`, `setValueAt`, `fireTableDataChanged` ... műveletek a szerkesztéshez.)
- A táblázat elemeinek megjelenítése is szabályozható, ekkor az oszlop `setCellRenderer` műveletével állítható be egy `DefaultTableCellRenderer` típusú objektum.

Az utolsó 3 lehetőségre (az oszlopok szélességének szabályozásával együtt) később visszatérünk.



Párbeszédablakok,
vezérlő elemek

OK és Mégsem gombot
tartalmazó
párbeszédablakok

Csúszka

Lista

Combobox

Combo lista

Sorszerkesztő

Spinner

Táblázat

Az alkalmazás kerete

Párbeszédablakok

Az alkalmazás kerete

- A különféle párbeszédablakokat egy keretből fogjuk vezérelni.
- Erre szolgál a `Dialogusok` osztály.
- Ebbe kerülnek be a különböző dialógusok, és a kezelésükhöz szükséges elemek, továbbá a menüpontok eseménykezelői.
- A tesztelést menü segítségével végezzük majd.
- A színeket a szövegszerkesztő mező háttér-, illetve betűszínében használjuk.
- Ebben a mezőben fogjuk naplózni a párbeszédablakok használatát.