

# My Calendar

## Dokumentáció

Készítette: Szilágyi Borbála, 2018

### Rövid leírás


A program egy ablakban megjeleníti a naptárat. A naptár havi nézetű, kiírja az éppen aktuálisan megjelenített hónap nevét, és az évet, ami hozzá tartozik.

A hónapok között két nyíl segítségével tud a felhasználó váltani. Értelmszerűen a jobbra mutató nyíllal előre tudunk lépni a következő hónapra, a balra mutató nyíllal pedig az előző hónap tekinthető meg.

A naptár fejléce tartalmazza a hét napjainak neveit. Ez alatt táblázat-szerűen szerepelnek az adott hónap napja annak megfelelően, hogy egy adott nap a hét mely napjára esik.

A felhasználónak lehetősége van események naptárba való felvételére, valamint események törlésére a naptárból. Ezek a funkciók menüpontok segítségével érhetőek el. A naptár fájlba való mentése automatikusan történik, így, ha a felhasználó bezárást követően újra elindítja a programot, felvett eseményei nem vesznek el.

### Grafikai megjelenés

 My Calendar

Manage events

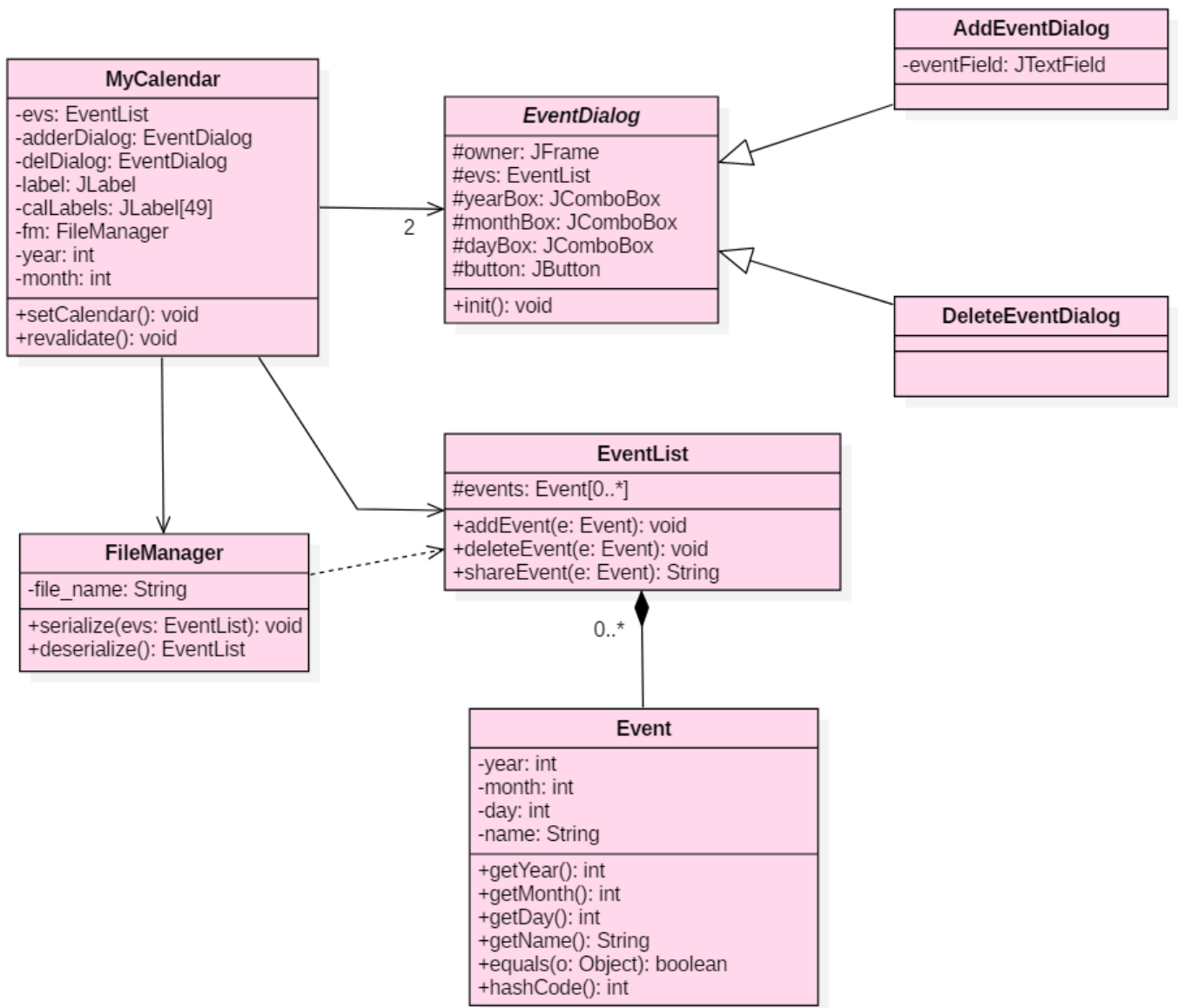
<

november 2018

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1 yoga	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

## Osztálydiagram



## Osztályok leírása

A program 8 modulból, ennek megfelelően 8 főosztályból áll. Ezeken kívül egyes osztályokban privát, belső osztályokat definiáltam a forráskód megfelelőbb tagolása érdekében.

### 1. Event

Az egyes események tárolására alkalmas osztály, megvalósítja a `Serializable` interfészt. Minden eseménynek van időpontja (év, hónap, nap), valamint neve. Az objektumorientált szemléletnek megfelelően az adattagok privát láthatóságúak, elérésüket getter függvények szolgálják.

```
public Event(int y, int m, int d, String n)
```

Négy paraméteres konstruktor, létrehozza a megfelelő eseményt.

```
public Event(int y, int m, int d)
```

Három paraméteres konstruktor, a névnek null pointert állít be.

```
public int getYear()
```

Visszaadja az esemény évét.

```
public int getMonth()
```

Visszaadja az esemény hónapját.

```
public int getDay()
```

Visszaadja az esemény napját.

```
public String getName()
```

Visszaadja az esemény nevét.

```
public boolean equals(Object o)
```

Felüldefiniálja az Object osztály egyező nevű függvényét. Szerepe az események összehasonlítása. Igaz értékkel tér vissza, amennyiben a két esemény dátuma megegyezik.

```
public int hashCode()
```

Felüldefiniálja az Object osztály egyező nevű függvényét, az equal függvény felüldefiniálása miatt volt szükségszerű az implementációja.

## 2. **EventList**

Az események listája, melyeket a program aktuálisan tárol. Lehetőséget biztosít események felvételére, valamint a listából való törlésére.

```
public EventList()
```

Konstruktor, létrehozza az események egy üres listáját.

```
public void addEvent(Event e)
```

Felveszi a megadott paraméterekkel rendelkező eseményt a listába. Ha már van esemény tárolva az adott dátummal, akkor ezt először törli.

```
public void deleteEvent(Event e)
```

Törli a megadott paraméterekkel rendelkező eseményt a listából.

```
public String shareEvent(Event e)
```

Megkeresi a megadott paraméterekkel rendelkező eseményt a listában. Visszatérési értéke az esemény neve. Ha nincsen találat, akkor null pointer.

## 3. **EventDialog**

Absztrakt osztály, a javax.swing.JDialog leszármazottja. Szerepe az, hogy az események felvételéért és törléséért felelős JDialog-ok megvalósítása során minél inkább elkerülhető legyen a duplikáció.

```
public AddEventDialog(JFrame o, EventList e1)
```

Konstruktor, amely paraméterként átveszi a tulajdonos ablakot, valamint a kezelendő események listáját. Létrehoz egy dialógusablakot, valamint beállítja a paramétereit.

```
public void init()
```

Ennek a függvénynek a segítségével történik az adatok hozzárendelése a JComboBox-okhoz.

#### **4. AddEventDialog**

Az EventDialog osztály leszármazottja, az események felvételéért felelős.

```
public AddEventDialog(JFrame o, EventList el)
```

Konstruktor, létrehozza a dialógusablakot, amely segítségével az új események felvétele történik. Az évet, hónapot és napot egy-egy JComboBox segítségével lehet kiválasztani, az esemény nevét pedig JTextField használatával lehet megadni. A kívánt esemény egy „ADD” feliratú JButton megnyomása után mentődik el.

```
private class ButtonListener implements ActionListener
```

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy az „ADD” gomb lenyomása után a megadott esemény felvétele megtörténjen az események listájába, és a felhasználói nézet frissüljön – így az újonnan felvett esemény is megjelenjen a naptárban.

#### **5. DeleteEventDialog**

Az EventDialog osztály leszármazottja, az események törléséért felelős.

```
public DeleteEventDialog(JFrame o, EventList el)
```

Konstruktor, létrehozza a dialógusablakot, amely segítségével az új események törlése történik. Paraméterként átveszi a tulajdonos ablakot, valamint a kezelendő események listáját. Az évet, hónapot és napot egy-egy JComboBox segítségével lehet kiválasztani. A megadott naptárról az esemény egy „DELETE” feliratú JButton megnyomása után törlődik ki.

```
private class ButtonListener implements ActionListener
```

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy a „DELETE” gomb lenyomása után a megadott esemény törlődjön az események listájába, és a felhasználói nézet frissüljön – így a törölt esemény már ne jelenjen meg a naptárban.

#### **6. FileManager**

A szerializálást megvalósító osztály. Privát attribútumként tartalmazza a kezelt fájl nevét.

```
public void serialize(EventList evs)
```

A paraméterként kapott eseménylistát kiírja abba a fájlba, amelynek nevét privát attribútumként tárolja.

```
public EventList deserialize()
```

A privát attribútumként tárolt fájlnevhez tartozó fájlt megnyitja, majd beolvassa a tartalmát. Visszatérési értéke a beolvasott események listája.

## 7. MyCalendar

Maga a naptár, a javax.swing.JFrame leszármazottja.

**public MyCalendar()**

Konstruktor, létrehozza a naptárt tartalmazó ablakot. Havi nézetet jelenít meg, a hónapok között két JButton segítségével lehet váltani. Tartalmaz egy menüt, aminek a segítségével a felhasználó felvehet és törölhet eseményeket. A megjelenítendő eseményeket fájlból olvassa be, a FileManager osztály segítségével.

**private class MyWindowAdapter extends WindowAdapter**

Belső osztály, a WindowAdapter leszármazottja. Felelőssége, hogy a naptár ablak bezárásakor az események fájlba írása megtörténjen. Így a program újbóli elindításakor a bevitt események változatlanul megjelennek.

**private class LeftButtonActionListener implements ActionListener**

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy a balra mutató nyilat tartalmazó gomb lenyomása után a megelőző hónapra váltson a nézet.

**private class RightButtonActionListener implements ActionListener**

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy a jobbra mutató nyilat tartalmazó gomb lenyomása után a következő hónapra váltson a nézet.

**private class AddEventListener implements ActionListener**

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy az „Add event” menüpontjának kiválasztását követően megjelenjen a megfelelő JDialog.

**private class DeleteEventListener implements ActionListener**

Belső osztály, megvalósítja az ActionListener interfészt. Felelőssége, hogy a „Delete event” menüpontjának kiválasztását követően megjelenjen a megfelelő JDialog.

**private void setCalendar()**

Frissíti a napokat, valamint a kijelzett eseményeket az aktuálisan kiválasztott hónapnak megfelelően.

**public void revalidate()**

Felüldefiniálja az ős egyező nevű függvényét. Hívásakor a naptár frissül.

## 8. Main

A főprogramot tartalmazó osztály.

**public static void main(String[] args)**

Létrehoz egy MyCalendar objektumot. Beállítja a láthatóságot és az átméretezhetőséget.

## Felhasználói kézikönyv

A program az elindítást követően megjelenít egy naptárt havi nézetben. Az elindítást követően az éppen aktuális hónap napjai láthatók.

Esemény felvételéhez a menüsorban ki kell választani a „Manage event”, majd azon belül az „Add event” opciót. A megjelenő ablakban ki kell választani a kívánt időpontot, valamint megadni az esemény nevét. A megadás módját az 1. ábra szemlélteti. Az „ADD” gombra kattintva az esemény mentése megtörténik, majd megjelenik a naptárban.

Események dátumának megadására 10 éven belüli intervallum áll lehetőségre, mely mindig az éppen aktuális évnek megfelelően van számolva. A grafikus megjelenítő felület korlátait figyelembe véve egy dátumhoz csak egy esemény felvételére van lehetőség. Amennyiben olyan dátum kerül megadásra, amelyhez már tartozott esemény, akkor a korábbi törlődik, és az új fog megjeleníteni.

1. ábra

Esemény törléséhez a menüsorban ki kell választani a „Manage event”, majd azon belül a „Delete event” opciót. A megjelenő ablakban ki kell választani az időpontot, amelyről az eseményt törölni szeretnénk. A megadás módját az 2. ábra szemlélteti. A „DELETE” gombra az esemény törlésre kerül, többet nem jelenik meg a naptárban.

2. ábra

A hónapok között két nyíl segítségével lehet váltani. A jobbra mutató nyíllal a következő hónapot tudjuk megjeleníteni, a balra mutató nyíllal pedig az előzőt. Ennek módját a 3. ábra szemlélteti.

3. ábra

A felvett események az ablak bezárását követően mentésre kerülnek, így újbóli megnyitás esetén láthatók lesznek – és nem vesznek el.