

# Seminář Java

## GUI – JFC/Swing

Radek Kočí

Fakulta informačních technologií VUT

Duben 2009

## Swing

- nové GUI dostupné od verze 1.2.x
- součást JFC (Java Foundation Classes)
- konečná verze GUI pro Javu

## AWT (Abstract Window Toolkit)

- starší varianta dostupná od verze 1.x.x
- od verze 1.1.x událostně řízená
- omezené možnosti

## Balíčky

- `java.awt` – základní komponenty AWT GUI
- `java.awt.event` – události AWT GUI
- + další balíky v `java.awt`
  
- `javax.swing` – základní komponenty Swing GUI
- `javax.swing.event` – události komponenty Swing GUI
- + další balíky v `javax.swing`

## Řízení programu událostmi

- obecnější pojem označující typ asynchronního programování
- základní princip tvorby GUI
- tok programu je řízen událostmi (zpracování událostí určuje běh aplikace)
- událostní aplikace by měly být programovány jako vícevláknové

## Komponenty GUI

- grafické (uživatelské) elementy – tabulka, text, ...
- grafické kontejnery

## Základní přístup

- událostní řízení
- kontejnery obsahují elementy a/nebo jiné kontejnery
- vzhled GUI je dán způsobem poskládání grafických elementů a kontejnerů

<http://java.sun.com/docs/books/tutorial/uiswing/components/>

# Ukázka Swing aplikace

```
public class HelloWorld {
    public static void main(String[] args) {
        // vytvoreni okna aplikace
        JFrame okno = new JFrame("Hello World");

        // vytvoreni textu a vlozeni do okna
        JLabel text = new JLabel("Nazdarek ...");
        okno.getContentPane().add(text);

        // implicitni operace pri zavreni okna
        okno.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);

        // otevreni okna
        okno.pack();
        okno.setVisible(true);
    }
}
```

## Rozmístění komponent

- komponenty se vkládají do grafických kontejnerů
- umístění komponenty není dáno absolutní polohou
- umístění komponenty je vztaženo relativně ke kontejneru, ve kterém je vložena
- způsob umístění je dán *správce umístění*
  - velikost, tvar a rozmístění závisí na typu správce
  - záleží na pořadí vložení (**add()**)

## Nastavení správce rozmístění

- v konstruktoru při vytváření kontejneru

```
converterPanel =  
    new JPanel(new GridLayout(2, 2));
```

- metodou `setLayout()` kontejneru

```
JFrame okno =  
    new JFrame("Hello World application");
```

```
okno.getContentPane().  
    setLayout(new FlowLayout());
```



## FlowLayout

- komponenty jsou ukládány zleva doprava na jeden řádek
- při zaplnění řádku se přechází na nový řádek
- implicitní

## GridLayout

- komponenty jsou ukládány do mřížky (tabulky)
- komponenty jsou ukládány zleva doprava a shora dolů do buněk mřížky
- počet sloupců a řádků se určuje v konstruktoru
- mřížka je rovnoměrná

## GridBagLayout

- nejmocnější (také nejsložitější) správce umístění
- o umístění prvků můžeme rozhodovat naprosto volně
- využíván především při automatickém generování kódu (RAD nástroje)

## BoxLayout

- vychází z **GridLayout**
- umožňuje vodorovné nebo svislé umístění
- umožňuje nastavit rozteče (mechanismus "rozpěry a tmel")

## BorderLayout

- implicitně umísťuje komponenty na střed a roztahuje na celou velikost kontejneru
- lze definovat oblast vložení v přeepsané metodě `add()`
  - `BorderLayout.NORTH`
  - `BorderLayout.SOUTH`
  - `BorderLayout.EAST`
  - `BorderLayout.WEST`
  - `BorderLayout.CENTER` (implicitní)

Bez správce

- `setLayout(null)`

## Řízení programu událostmi

- událost vzniká obvykle uživatelskou akcí (kliknutí, změna polohy myši, ...)
- událost může vzniknout v libovolné komponentě GUI
- každá komponenta má definovaného "posluchače" (**listener**) událostí
- při vyvolání události zašle systém zprávu posluchači – ten událost zpracuje

Události lze rozdělit podle uživatelské akce nad

- oknem – **WindowEvent**
- klávesnicí – **KeyEvent**
- myší (klikání, pohyb) – **MouseEvent**
- fokusem (získání, ztráta) – **FocusEvent**
- GUI (obecná akce, např. stisk tlačítka) – **ActionEvent**
- ... (viz [java.awt.event](#) a [java.swing.event](#))

## Posluchač událostí

- objekt, jehož třída implementuje příslušné rozhraní

## Rozhraní posluchačů událostí

- **ActionListener** (awt)
- **MouseListener** (awt)
- **MouseMotionListener** (awt)
- **MouseListener** (swing)
- ... (viz [java.awt.event](#) a [java.swing.event](#))

## Přidání posluchače události

- komponenta musí registrovat posluchače událostí, aby příslušná událost mohla být ošetřena
- každá komponenta knihovny Swing obsahuje
  - metodu `addXXXListener()`
  - metodu `removeXXXListener()`
  - kde `XXX` reprezentuje název události (`Mouse, ...`)
- např. `addActionListener(ActionListener listener)`

## Implementace posluchače událostí

- anonymní třída
- vnitřní třída
- top-level třída



```
class myActionListener implements ActionListener {  
    public void actionPerformed(ActionEvent event)  
        ...  
    }  
}
```

---

```
...  
convertTemp = new JButton("Convert");  
convertTemp.  
    addActionListener(new MyActionListener());  
...
```

```
java.awt.Component  
    java.awt.Container  
        java.swing.JComponent
```

Třída **JComponent**

- metoda **void paintComponent(Graphics g)**
- metoda **void repaint()**

```
button.addActionListener(new DoSomething());
```

---

```
class DoSomething implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {}
    }
}
```

```
class DoSomething implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        Runnable r = new Runnable() {
            public void run() {
                label.setText("Time consuming operation");
                try {
                    Thread.sleep(10000);
                } catch (InterruptedException e) {}
                label.setText("");
            }
        };
        new Thread(r).start();
    }
}
```

```
public static void main(String[] args) {  
    //Schedule a job for the event-dispatching thread  
    //creating and showing this application's GUI.  
    javax.swing.SwingUtilities.  
        invokeLater(new Runnable()  
    {  
        public void run() {  
            createAndShowGUI();  
        }  
    });  
}
```

`http://java.sun.com/docs/books/tutorial/  
uiswing/learn/`