



**MORO  
SYSTEMS**

# SWT & MigLayout

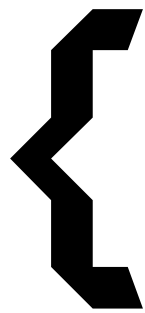
Alternativy Java GUI v praxi

Pavel Janečka & Tomáš Chlouba



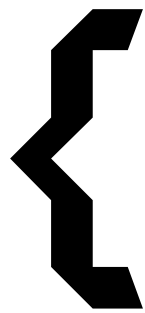
Univerzita Hradec Králové  
Fakulta informatiky a managementu

červen 2011



# SWT & MigLayout

- SWT
  - Historie
  - Komponenty a události
  - Vývoj aplikací a ukázky
- MigLayout
  - Java Layout Managery
  - Ukázka použití a vlastnosti MigLayoutu
  - Pokročilý layout pomocí SWT & MigLayout

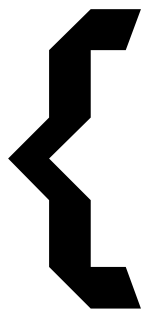


# Standard Widget Toolkit

- Open-source wrapper nativních knihoven operačního systému
- Definuje společné rozhraní pro všechny podporované platformy s využitím nativních komponent
- Alternativa k lightweight knihovně Swing

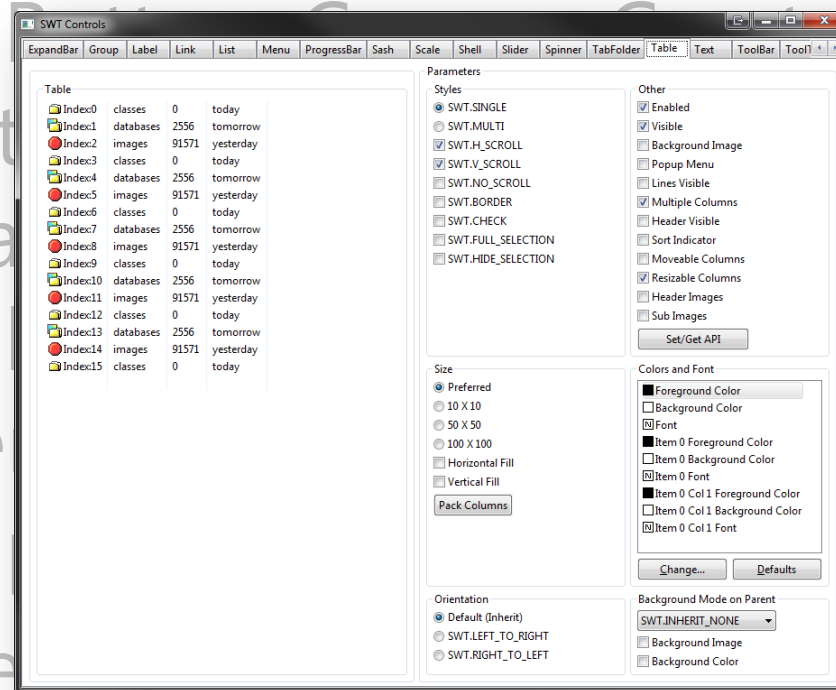
# { SWT ~ Historie

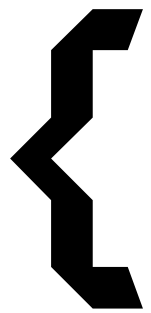
- JDK 1.0 – AWT (Abstract Window Toolkit)
  - 1995, heavyweight komponenty
- JDK 1.2 - Swing
  - 1997, vykreslování kompletně v jazyce Java
- SWT – původně pro jazyk Smalltalk
  - 1993, otevření vývoje IDE Eclipse (dříve VisualAge)
  - IBM vyžadovala native look and feel



# SWT ~ Komponenty a události

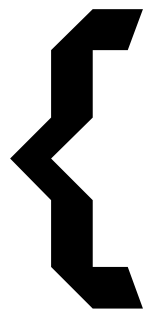
- Browser, Composite, ExpandBar, Group, Label, Link, List, Menu, ProgressBar, Sash, Scale, Shell, Slider, Spinner, TabFolder, Table, Text, ToolBar, Tree, Combo, ExpandBar, ProgressBar, Spinner, Tree
- Activate, DefaultSelection, Detect, ...
- Widgets, Snippets, Examples, Docs, FAQ, ...





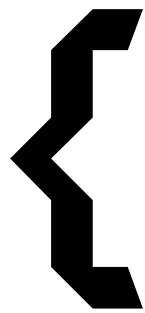
# SWT ~ Design a look and feel

- Cíle SWT
  - Vysoký výkon
  - Nativní vzhled
  - Vysoká integrace do systému
- „Menší“ možnosti než Swing
- SWT neimplementuje MVC
  - JFace projekt (Eclipse)
- DWT, SwingWT, SWTSwing



# SWT ~ Vývoj aplikací

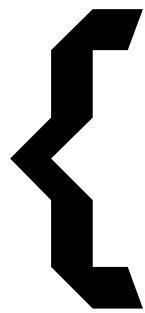
- Z důvodu vazby na nativní knihovny nedovoluje SWT dědění komponent
  - FAQ - Why can't I subclass SWT widgets like Button and Table?
- SWT neodesílá programově vyvolané události
- Single-threaded UI model
  - FAQ - Why do I get the error "org.eclipse.swt.SWTException: Invalid thread access"?
- Nastavení komponent pomocí bitového součtu



# SWT ~ Vývoj aplikací

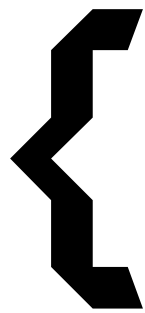
- Nutnost ruční dealokace
  - Metoda `.dispose()`
  - Hlavně z důvodu využívání nativních objektů
  - V praxi pouze obrázky, barvy a fonty
- Složitější tvorba vlastních komponent
  - Potomci objektů Canvas či Composite
    - Metoda `dispose`
    - `PaintListener`, `ControlListener`, `EventListener`





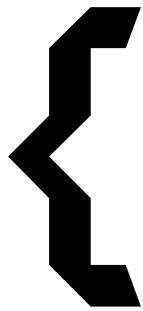
# SWT ~ Vývoj aplikací

- Komponenty neobsahují standardně přednastavený layout manager
  - Každé komponentě představující kontejner musí být explicitně nastaven vybraný manager
  - Komponenty se jinak vůbec nezobrazují
- FillLayout, RowLayout, GridLayout, FormLayout
  - Understanding Layouts in SWT  
<http://www.eclipse.org/articles/article.php?file=Article-Understanding-Layouts/index.html>



# SWT ~ Vývoj aplikací

- Při vytváření instance komponenty nutná znalost parenta
- Obsluha layout manažerů
  - `void setLayoutData(Object layoutData)`
- Podporované platformy
  - Windows (Win32, WPF), Linux, Solaris 10, HP-UX, AIX (GTK 2), Mac OS X (Cocoa), Pocket PC (Windows CE)



# SWT ~ Vývoj aplikací

Hello world

```
public SWTHelloWorldApp() {
```

```
    Display display =
```

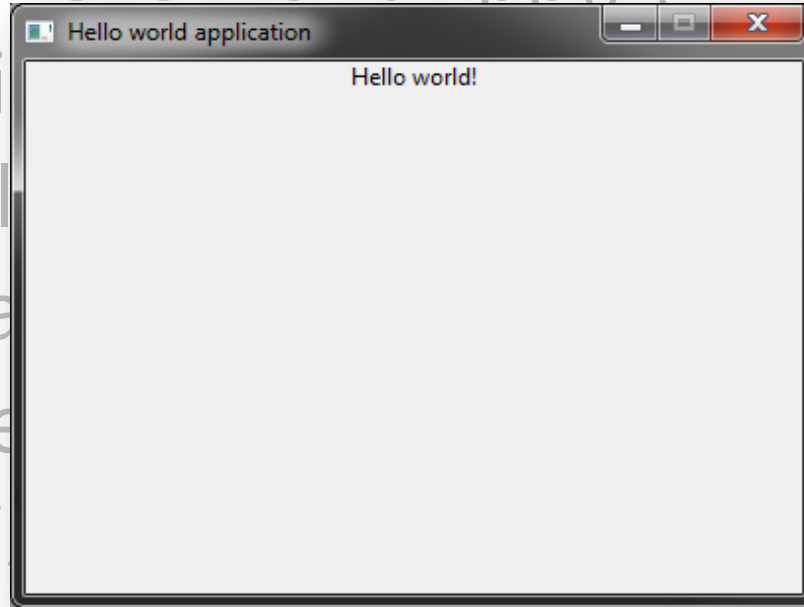
```
    new Display("Hello world");
```

```
    Shell shell =
```

```
    new Shell(shell, SWT.CLOSE
```

```
    | SWT.VERTIC
```

```
    | SWT.CEN
```

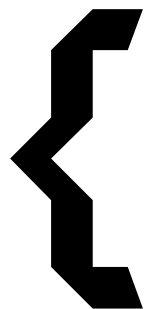


```
SWT.CLOSE
```

```
SWT.VERTIC
```

```
ation");
```

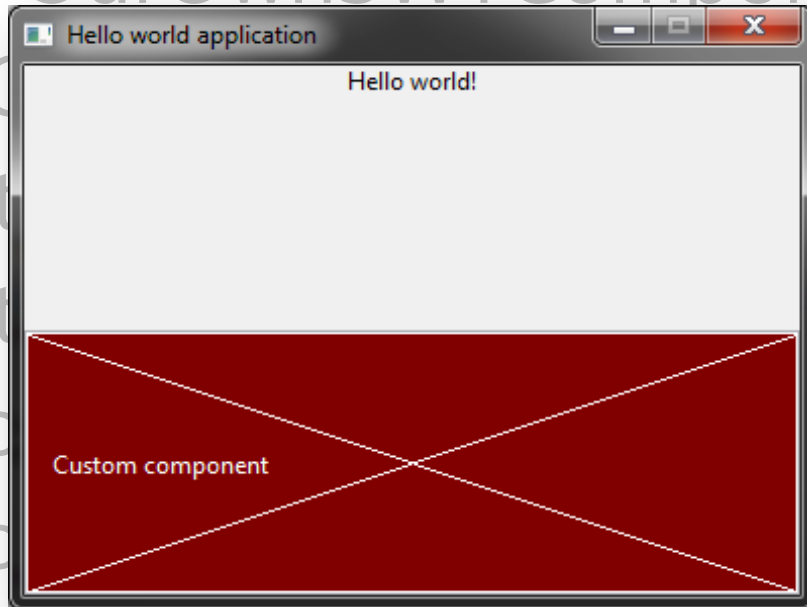
```
Label lblHello = new Label(shell, SWT.CEN
```

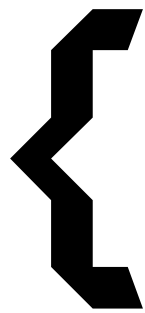


# SWT ~ Vývoj aplikací

## Tvorba vlastní komponenty

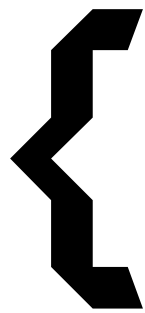
```
public class OurOwnSWTComponent extend  
private GC  
private int  
private int  
private Co  
private Co  
private Font fntText = new Font(this.getDis
```





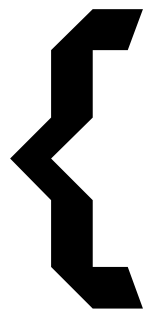
# Standard Widget Toolkit

- Java wrapper knihoven uživatelského rozhraní různých operačních systémů
- Výhody
  - Rychlost a paměťová nenáročnost!
  - Nativní vzhled aplikací
  - Bitový součin pro nastavení komponent
  - Licence (Eclipse Public Licence)
  - Komunita



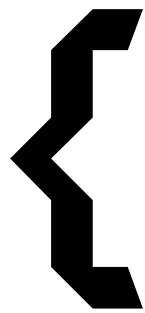
# Standard Widget Toolkit

- Nevýhody
  - Dodatečné knihovny
    - Různé platformy, x64 problém
  - Platformě závislé buggy
    - Tabulka, browser (Windows)
- Jiný přístup
- Nelze přímo stáhnout JavaDoc
  - <http://paul.luminos.nl/update/485>



# Layout Managery v Javě

- Slouží k tvorbě GUI
- Ruční tvorba náročná a nepřehledná
- Složité GUI realizovatelné pouze pomocí vnořování komponent s různými LM
- Malá změna v kódu má velký vliv na vzhled
- Co se stane, když změním velikost okna?



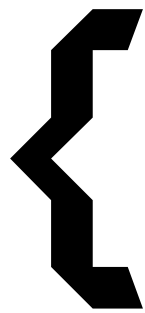
# Layout Managery v Javě

- Swing Layout Managery
  - BorderLayout, FlowLayout
  - GridLayout, GridBagLayout
  - BoxLayout, SpringLayout



# MigLayout

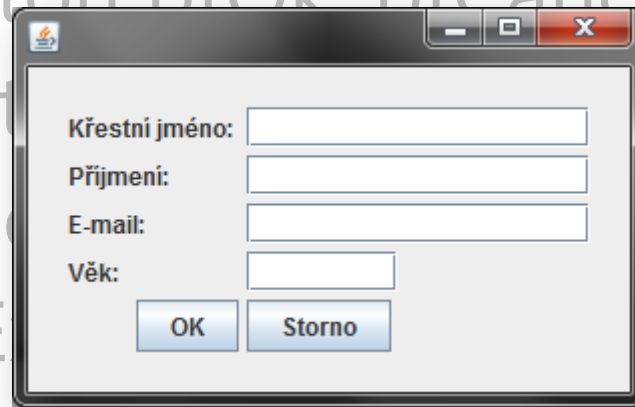
- Autorem Mikael Grev
- Alternativa klasických layout managerů bez nutnosti používat vnořování komponent
- Proč MigLayout?
- [miglayout.com](http://miglayout.com)
  - Quick Start, tutoriály

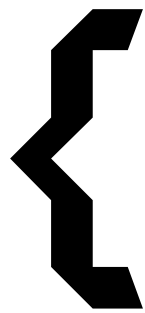


# MigLayout ~ ukázka

Jednoduchý layout v MigLayoutu x Swing

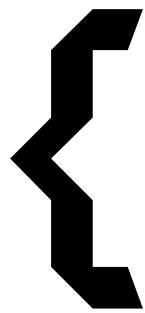
```
public class MiGExample {  
    private JButton btOk, btCancel;  
    private JText... e, tfLastName,  
    private JLabel... LastName, lb  
    public MiGE  
    JFrame frmWindow = new JFrame();  
    JPanel pnWin = new JPanel(new MigLayout
```





# MigLayout ~ constraints

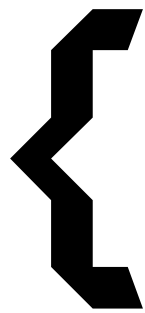
- Constraints pro komponentu nastavujeme při přidávání komponenty
  - wrap – zalomení řádku
  - span – slučování buněk
  - push – doplnění mezery
  - split – „rozdělení“ buněk
  - cell – absolutní pozicování



# MigLayout ~ constraints

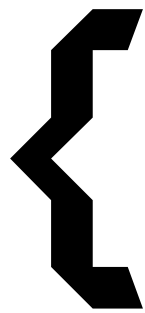
- Constraints můžeme nastavit pro celý layout, jednotlivé řádky nebo sloupce
- Zalamování řádků, mezery, rozměry, zarovnání, insets

```
MigLayout layout = new MigLayout(
    "", // layout
    "[[]10[]", // sloupce
    "[[]10[]"); // řádky
```



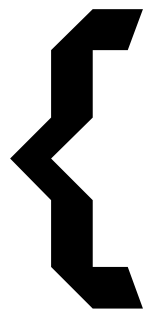
# MigLayout ~ rozměry

- Rozměry komponent můžeme nastavit individuálně pro jednotlivé komponenty
- Možnost nastavit minimum, preferred a maximum velikost pro Swing, preferred pro SWT
  - width (w), height (h), !
  - min, pref, max



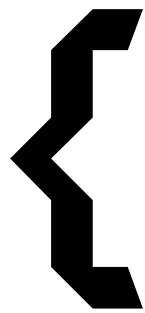
# MigLayout ~ zarovnání

- Komponenty v buňkách je možné zarovnat v rámci buňky, pokud kolem nich je volné místo
  - (align) left, right, center
- Dokovat komponenty je možné na čtyři strany okna, dokované komponenty jsou vyjmuty ze základního layoutu
  - (dock) north, south, east, west



# MigLayout ~ doplnění

- Komponentám je možné nastavit automatické obsazení celé buňky
  - grow, shrink
- Debug mód při zobrazení komponent, díky kterému je přehledně zobrazeno jejich rozložení
  - debug
- NetBeans plug-in



# MigLayout ~ benchmark

- Jednoduchý benchmark SWT & Swing

## Swing

Java Version: 1.6.0\_23  
Look & Feel: The Microsoft Windows Look and Feel  
Time to Show: 449 millis.  
Benchmark Run Time: 14952 millis.  
Average Run Time: 747 millis (20 runs).

## SWT (x64)

Java Version: 1.6.0\_23  
Time to Show: 401 millis.  
Benchmark Run Time: 64391 millis.  
Average Run Time: 3219 millis (20 runs).

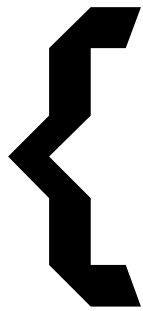
## SWT (x32)

Java Version: 1.6.0\_23  
Time to Show: 406 millis.  
Benchmark Run Time: 57526 millis.  
Average Run Time: 2876 millis (20 runs).

- Další výsledky

- <http://www.javalobby.org/java/forums/t78884.html>



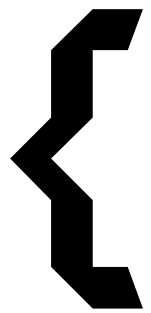


# SWT & MigLayout

The screenshot shows the KASS application window with several red annotations pointing to specific UI elements:

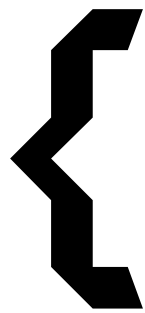
- wrap split | wrap**: Points to the 'Data sources' section.
- split 3, flowy**: Points to the 'List of tests and table of tests to run' section.
- align center**: Points to a button in the 'List of tests' section.
- split 5, flowy, sg btns, center**: Points to the right-hand sidebar containing buttons like 'Serial correlation', 'Histogram', etc.
- wrap**: Points to the 'Configuration of data source' section.
- span 2, wrap, growx**: Points to the table of test results.
- split 3, span**: Points to the 'Run tests', 'Restart tests', and 'View tests results' buttons.
- growx, dock south**: Points to the 'interactive help for KASS application' section at the bottom.

#	Test name	Progress	Result
1	Cochran test	0 %	Not availa...
2	Spearman Order Correlation Coefficie...	0 %	Not availa...
3	Extreme Point Analysis	0 %	Not availa...



# SWT & MigLayout

Jste připraveni vyzkoušet  
alternativy Java GUI ?



# SWT & MigLayout ~ Zdroje

- Eclipse Foundation. SWT Home [online]. 2005 [cit. 2011-04-25]. SWT: The Standard Widget Toolkit. Dostupné z WWW: <http://www.eclipse.org/swt/>
- MiG InfoCom AB. MigLayout [online]. 2007 [cit. 2011-04-25]. MigLayout - The Java Layout Manager for Swing, SWT. Dostupné z WWW: <http://www.miglayout.com/>.