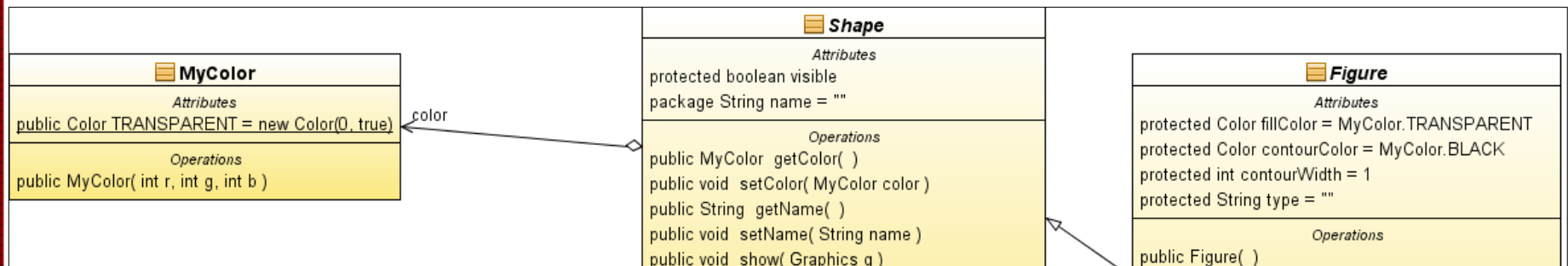


# Разработка на приложения с графичен потребителски интерфейс в езика Java™



Траян Илиев

IPT – Intellectual Products & Technologies

e-mail: [tiliev@iproduct.org](mailto:tiliev@iproduct.org)

web: <http://www.iproduct.org>

Oracle®, Java™ and EJB™ are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Oracle®, Java™ и EJB™ са търговски марки на Oracle и/или негови подразделения. Всички други търговски марки са собственост на техните притежатели.

## Съдържание

1. Графичен потребителски интерфейс (GUI)
2. Прозорци и аплети
3. Стартиране на аплети
4. Swing библиотека на езика Java™ за реализиране на графичен потребителски интерфейс
5. Бутони, етикети, тестови полета
6. Model-View-Controller архитектура
7. Събитиен модел на Swing – типове събития и слушатели
8. Контролиране на разположението – layout класове

## Съдържание II

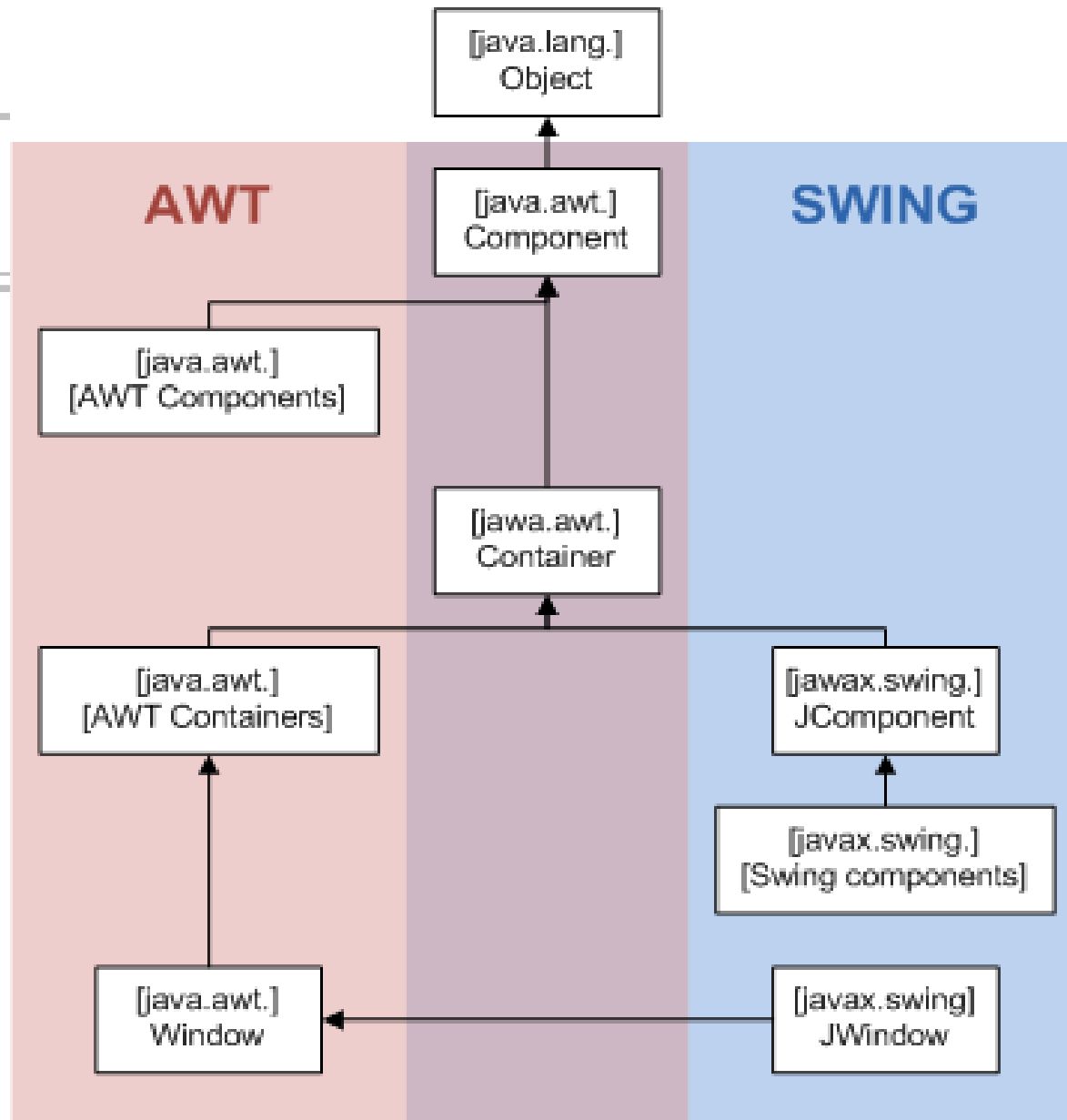
9. Новости при аpletите и JNLP в Java 6 и 7 – Draggable Applets, Nimbus Look and Feel, JLayer
10. Визуално програмиране с Java Beans™
11. Изпълнение на времеемки задачи. Конкурентност в Swing
12. Графично изграждане на desktop приложения с Eclipse WindowBuilder плъгин

## Технологии за реализация на графичен потребителски интерфейс с езика Java™

- Съществуват различни опции за реализиране на приложения с графичен потребителски интерфейс (GUI) с Java™:
  - Abstract Window Toolkit (AWT) + Swing – предимства: част от стандартния Java SE API, платформена независимост, разширяемост, конфигурируемост, ефикасност (lightweight UI), Loosely-Coupled & MVC
  - Eclipse Standard Widget Toolkit (SWT) и Rich Client Platform (RCP) – бързина, ефикасност, OS native look & feel, Reusable Design Patterns & MVC, стандартни OSGi компоненти, цялостна платформа за изграждане на GUI приложения

## Реализация на графичен потребителски интерфейс с JFC/Swing

- Основни библиотеки за изграждане на графичен потребителски интерфейс (Graphical User Interface – GUI):
  - Abstract Window Toolkit (AWT)
  - Java™ Foundation Classes (JFC)
  - Swing
- Програмен модел на Swing
- Различни стилове на оформление на интерфейса („pluggable look and feel“)



## Прозорци и аплети. Стартиране на аплети

- Аплети в езика Java™ - особености, предимства и ограничения
- Жизнен цикъл на аplet
  - `init()`
  - `start()`
  - `stop()`
  - `destroy()`
- Стартиране на аплети:
  - с `AppletViewer`
  - в Уeб – браузър

## Swing библиотека за реализиране на графичен потребителски интерфейс

- Рамка на прозорец – базов клас JFrame

```
myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
myFrame.setSize(width, height);
```

```
myFrame.setBounds(Rectangle r);
```

```
myFrame.setLocationRelativeTo(null);
```

```
myFrame.setVisible(true);
```

- Графични компоненти – бутони, етикети, текстови полета и области
- Реализиране на интерактивност (реагиране на събития свързани с компонента)



## Контролиране на разположението – layout класове

- Метод `setLayout( )` на класа `Container`
- Мениджъри на разположението на компонентите:
  - `null` = абсолютно позициониране
  - `BorderLayout`
  - `FlowLayout`
  - `GridLayout`
  - `GridBagLayout`
  - `BoxLayout`
  - `GroupLayout` – **GUI Builders** (например **Eclipse WindowBuilder**)

## Model-View-Controller архитектура

- Многократно използвани шаблони за проектиране – **Reusable Design Patterns**
- Примери – **Decorator, Observer**
- Трислойна **Model-View-Controller** архитектура
- Модел на данните - **Model**
- Потребителски интерфейс – **View**
- Междинен слой (бизнес логика на приложението) – **Controller**

## Събитиен модел на Swing – типове събития и слушатели на събития

- Типове събития и слушатели
- Добавяне и премахване на слушатели за събитие към/от компонент
- Пример:  
**ActionEvent**  
**ActionListener**  
**addActionListener( )**  
**removeActionListener( )**
- Методи на основните слушателски интерфейси
- Адаптерни класове за отделните интерфейси

## Swing компоненти

- икони
- рамки
- панели
- области за текстово редактиране
- радио-бутони
- комбинирани полета (combobox)
- списъчни полета
- прозорци със съобщения
- плъзгачи и индикатори за прогрес
- менюта
- контекстни менюта
- изчертаване
- диалогови прозорци

## Swing – по-сложни елементи

- Визуализация и управление на дървета и таблици
- Избор на изглед (**Look & Feel**)
- Новостите при графичната визуализация със Swing в Java™ 7 – **Nimbus Look and Feel**, полупрозрачни и не-правоъгълни прозорци, **JLayer** компонент позволяващ създаване и управление на интерактивни полупрозрачни слоеве в компонентите
- Пакетиране на аplet в **.jar** файл
- Подписани **jar** файлове и аплети
- **Java™ Web Start** и **JNLP**

## Интерактивни, полу-прозрачни слоеве с JLayer

- **JLayer** = шаблон „декоратор“, който позволява да рисуваме върху произволни компоненти и произволни техни области, както и да **прихващаме събития** на тези компоненти, без да ги модифицираме директно
- Наследява **JxLayer** проекта и е част от **swing** в **Java 7**
- Състои се от два класа – **JLayer** (final class) и **LayerUI** (предназначен да бъде наследяван – дефинира методи **paint()**, **installUI()**, **uninstallUI()**, **processXXXEvent()** и др.):

```
JPanel panel = createMyBasePanel();  
LayerUI<JPanel> myLayerUI = new myDecoratorLayerUI();  
JLayer<JPanel> myLayer = new JLayer<JPanel>(panel, myLayerUI);  
myJFrame.add (myLayer);
```

## Изпълнение на времеемки задачи. Конкурентност в Swing

- Основен принцип: времеемките задачи се изпълняват в отделни нишки от тази за обработка на събития в Swing
- Видове нишки:
  - Initial threads – нишките на нашата програма
  - Event dispatch thread – нишка на Swing, в която се изпълняват слушателите на събития
  - Worker (background) threads – отделни нишки за изпълняване на времеемки изчисления
- `SwingUtilities.invokeLater(Runnable doRun)` и `SwingUtilities.invokeAndWait(Runnable doRun)`

## Подписване на jar файлове и аплети

- Основи на мрежовата сигурност. Сертификати
- Подписаният аplet (пакетиран като .jar файл) удостоверява на потребителя, че кода който предстои да зареди и стартира наистина е създаден от притежателя на съответния сертификат и не е модифициран.
- Подписаните аплети имат повече права – например да достъпват файлове на потребителския компютър
- Създаване на тестови (Self Signed) сертификати
- Процес на подписване – команди **keytool** и **jarsigner**



## Подписване на jar файлове - детайли

- Хранилища на сертификати

```
keytool -list -keystore "C:\Program  
Files\Java\jdk1.6.0_25\jre\lib\security\cacerts"
```

- Генериране на сертификат

```
keytool -genkey -alias <keyname> -keystore <url>
```

- Подписване на jar файл

```
jarsigner -keystore <url> <jarfile> <keyname>
```

## Вграждане на подписан аplet

```
<APPLET  
  CODE=mypackage.MyClass.class  
  ARCHIVE = myjar.jar  
  WIDTH=600  
  HEIGHT=400>  
</APPLET>
```

## Новости при аpletите в Java 6 update 10

- Повишена надеждност чрез изолиране на аплета от браузъра
- Java™ Web Start и JNLP (Java™ Network Launch Protocol)
- Индивидуализирани настройки на JVM за всеки отделен аplet
- Едновременна поддръжка на множество JRE версии
- Подобрена Java/JavaScript комуникация
- Подобрено зареждане на аpletите във фонов режим
- Разширен размер на *heap* -а
- “Draggable” аплети, които могат да се извличат и поставят извън браузъра

## “Draggable” аплети

```
<APPLET code="DragExample" codebase="../bin"  
http://javafx.com/docs/articles/ShouldDrag/im/applet_dragged2.jpg  
width="600" height="600">
```

```
<PARAM name="jnlp_href " value="applet_dragged2.jnlp">
```

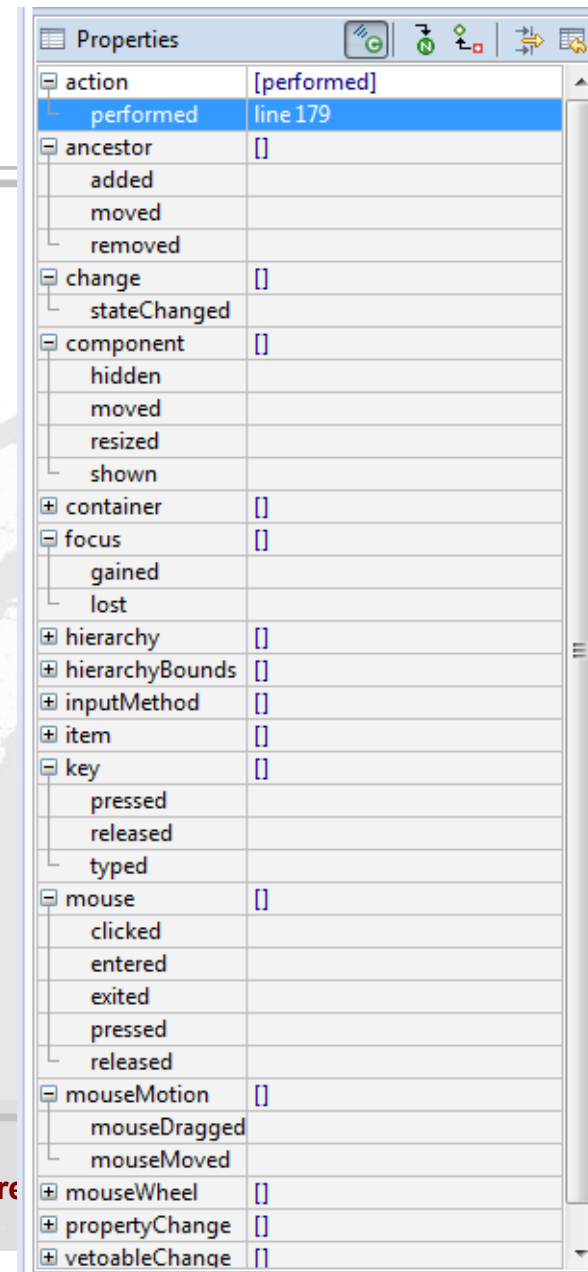
```
<PARAM name="draggable" value="true">
```

```
</APPLET>
```

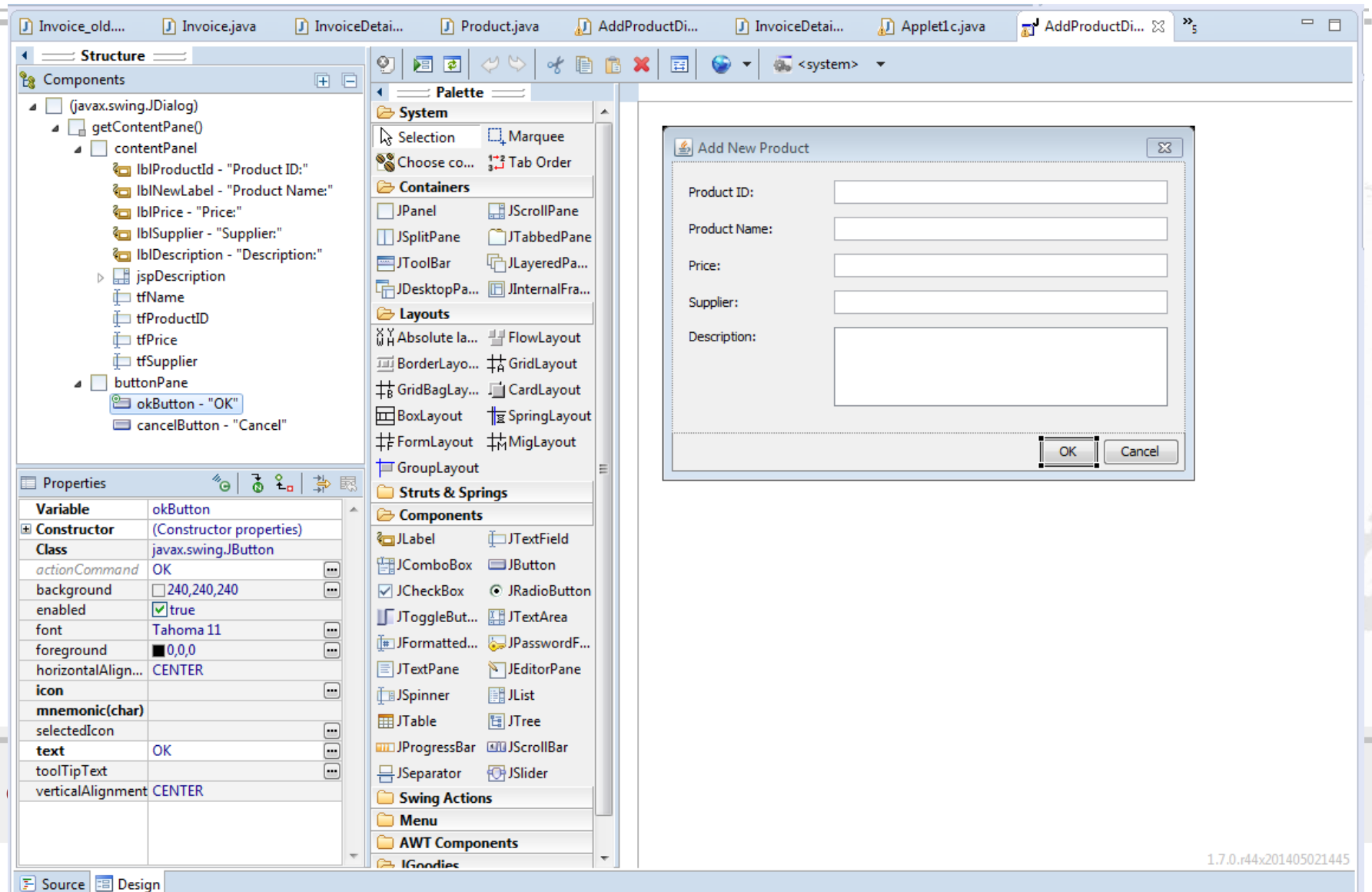
- JNLP описание – XML файл с разширение .JNLP
- Предефиниране на методи на класа **JApplet**:
  - public boolean isAppletDragStart(MouseEvent e);
  - public void appletDragStarted();
  - public void appletDragFinished();
  - public void setAppletCloseListener(ActionListener l);
  - public void appletRestored();

# Графично изграждане на приложения

- Визуално програмиране с Java Beans™
  - свойства (properties) – getXXX() или isXXX() + setXXX()
  - събития (events) – XXXEvent
  - бизнес методи (произволно име)
- Графично изграждане на desktop приложения с NetBeans и Swing Application Framework (JSR 296)
- Графично изграждане на desktop приложения с Eclipse WindowBuilder



# Eclipse WindowBuilder (1)



## Eclipse WindowBuilder (2)

- **WYSIWYG** визуален редактор и **layout** инструменти за създаване на форми, прозорци и менюта; **Java кодът** се генерира автоматично; **контролите** се добавят с **drag-and-drop**, с кликане могат да се прикачат **обработчици** към дефинираните за контрола събития, да се променят **свойства**, да се **интернациоанлизира (i18n)** текста и др.
- Създаден през **2003** WindowBuilder подобрява производителността на GUI дизайна **2x – 10x** – печели начграда за **Best Commercial Eclipse Tool** през 2009.
- Поддържа **Swing, SWT, XWT** и **Google's GWT**
- **Bi-directional code generation (round-trip editing), Visual inheritance, Morphing, Data-binding, Wizards and templates ...**

## Литература и интернет ресурси

- Екел, Б., Да мислим на JAVA. Софтпрес, 2001.
- Oracle® Java™ Technologies webpage – <http://www.oracle.com/technetwork/java/>
- Oracle® Java™ Swing Tutorials – <http://docs.oracle.com/javase/tutorial/uiswing/>
- How to Decorate Components with the JLayer Class – <http://docs.oracle.com/javase/tutorial/uiswing/misc/jlayer.html>
- Eclipse WindowBuilder Plugin – <https://eclipse.org/windowbuilder/>
- Oracle® Java™ Web Start Technology webpage – <http://docs.oracle.com/javase/6/docs/technotes/guides/javaws/>



Благодаря Ви за вниманието!

Въпроси?