

# *Środowisko NetBeans*

Paweł Boguszewski



# *Plan prezentacji*

- Wprowadzenie
- Historia
- Funkcjonalność
- Kierunek Rozwoju
- Wtyczki
- Platforma NetBeans
- Podsumowanie



# *Wprowadzenie*

- NetBeans.org
  - projekt OpenSource zapoczątkowany w 2000 r.
  - rozwijany pod skrzydłami SUNa
  - ponad 3.6 miliona ściągnięć produktów
  - Flagowe produkty
    - NetBeans IDE
    - Platforma NetBeans

# Wprowadzenie

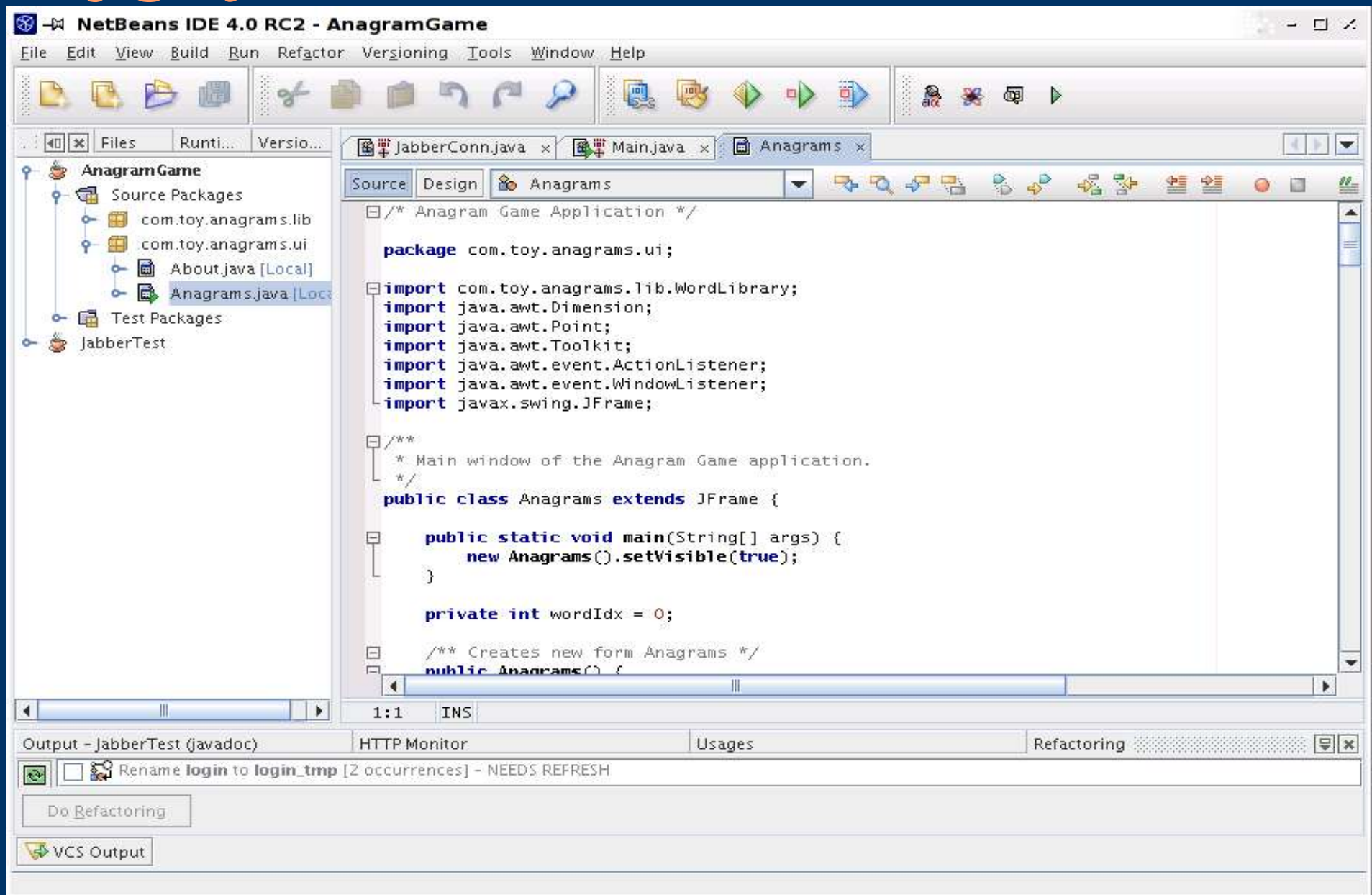
- NetBeans IDE
  - W pełni modularne środowisko posiadające:
    - Edytor z kontrolą składni kodu
    - Debugger
    - System kontroli wersji
    - Narzędzia do refaktoryzacji i testowania programów
  - Możliwość rozszerzenia funkcjonalności poprzez instalację dodatkowych wtyczek



## *Historia*

- 1996 – powstanie projektu Xelfi
  - Zmiana nazwy na NetBeans
  - Modularyzacja środowiska
  - Udostępnienie OpenAPI (wersja 2.x)
  - 1999 – przejęcie przez Suna (wersja 3.0)
  - 2000 – ogłoszenie NetBeans projektem OpenSource
  - CollabNet przygotowuje platformę NetBeans dla środowiska OpenSource
  - Wydanie przez Suna JavaForte zbudowanego na bazie NetBeans
  - NetBeans flagowym IDE Suna
- 
-

# Wygląd



# *Funkcjonalność*

- Interfejs użytkownika
    - Możliwość dostosowania wyglądu środowiska do własnych potrzeb
    - Graficzne przedstawienie informacji o klasach i pakietach
    - Łatwy sposób zarządzania plikami
    - Kreatory przyśpieszające programowanie
    - Możliwość pracowania nad kilkoma projektami jednocześnie
    - Zautomatyzowany mechanizm aktualizacji środowiska
- 
-

# *Edytor*

- Podświetlanie składni dla plików źródłowych Javy, XMLa, HTMLa, CSS i wielu innych
- Parsowanie kodu “w locie”
- Autouzupełnianie oraz autoformatowanie kodu
- Importowanie klas Javy





# Edytor

The screenshot shows the Edytor IDE interface. The main editor displays the following Java code:

```
import org.jivesoftware.smack.*;
import org.jivesoftware.smack.packet.*;

public class JabberConn {
    XMPPConnection connection;

    public void login(String user, String passwd) throws XMPPException {
        connection.login(user, passwd);
    }

    public void connect() {
        System.out.println("Connecting...");
    }

    public void ...
}
```

A tooltip is visible over the `System.out` reference in the `connect` method. The tooltip lists the following members of `java.lang.System`:

- Class `class`
- PrintStream `err`
- InputStream `in`
- PrintStream `out`
- `void arraycopy(Object src, int srcPos, Object dest, int destPos, int l...`

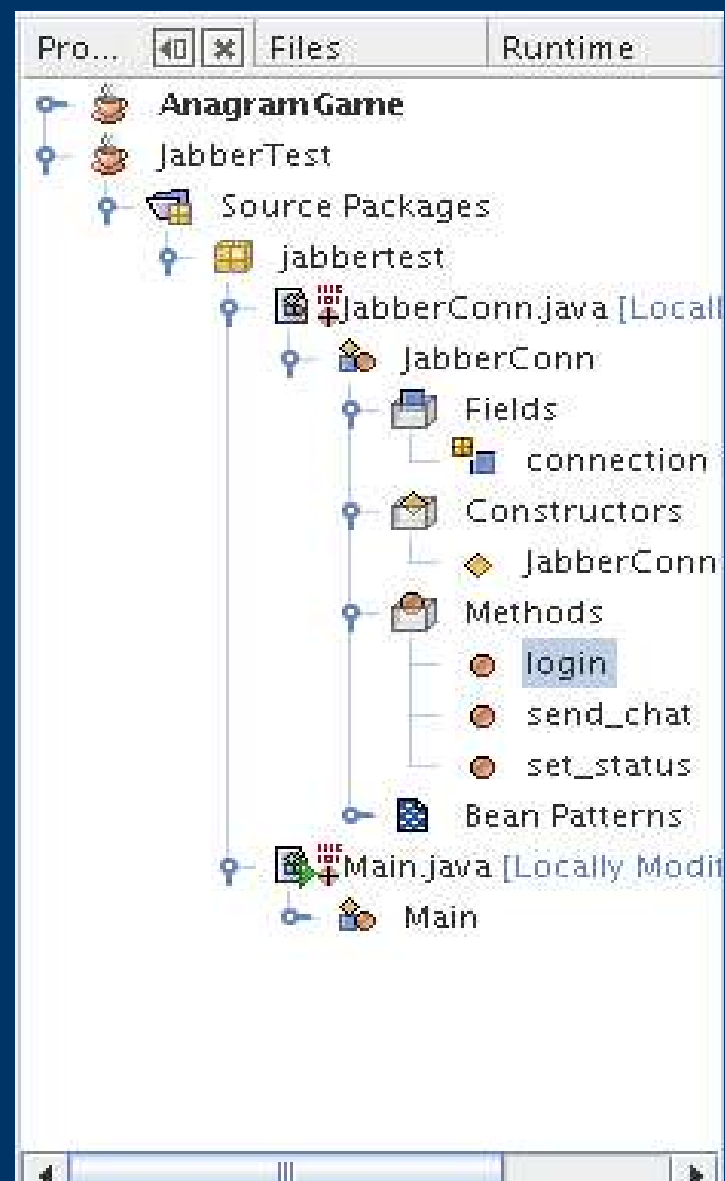
Below the tooltip, a detailed view of `java.lang.System` is shown, including the declaration of the `out` field:

```
public static final PrintStream out
```

The description for `out` reads: "The "standard" output stream. This stream is already open and ready to accept output data. Typically this stream corresponds to display output or

# Edytor

- Hierarchiczny podgląd projektu
- Szybki podgląd klas i obiektów
- Możliwość pracy z wieloma projektami naraz



# Edytor

- Wsparcie dla Javadoca
  - Narzędzia ułatwiające opisywanie klas i metod
  - Łatwe wyszukiwanie i przeglądanie
  - Generacja dokumentacji projektu

# Edytor

The screenshot displays the Edytor IDE interface. At the top, the window title bar shows several tabs: "...ut", "Anagram s x", "JabberConn.java \* x", "Main.java \* x", and "Auto Comment Tool - main x".

The main editor area is split into two panes. The left pane shows a project tree with a folder icon and a red 'X' icon next to "Main", and a file icon with a yellow warning triangle next to "main(java.lang.String[])". To the right of this tree are four buttons: "Auto Correct", "View Source", "Refresh", and "Help". Below the tree, a text field labeled "Declaring Class:" contains the text "Main". At the bottom of this pane, there is a scrollable text area containing the following text:

```
The tag for the exception org.jivesoftware.smack.XM
The tag for the exception java.io.IOException is mis:
```

The right pane is titled "Javadoc Comment Text:" and contains a text area with the text "test". Below this is a section titled "Tags:" containing a text area with the text "@param args argumenty programu". To the right of the tags list are four buttons: "New...", "Delete", "Move Up", and "Move Down". Below the tags section, the text "No tag selected." is displayed. At the bottom of the right pane, there is a row of six buttons: "<B>", "<I>", "<U>", "<C...>", "<P...>", and "{@|I...>".

# *Edytor*

- Szybki import klas Javy
- Panel z możliwością umieszczenia najczęściej używanych opcji

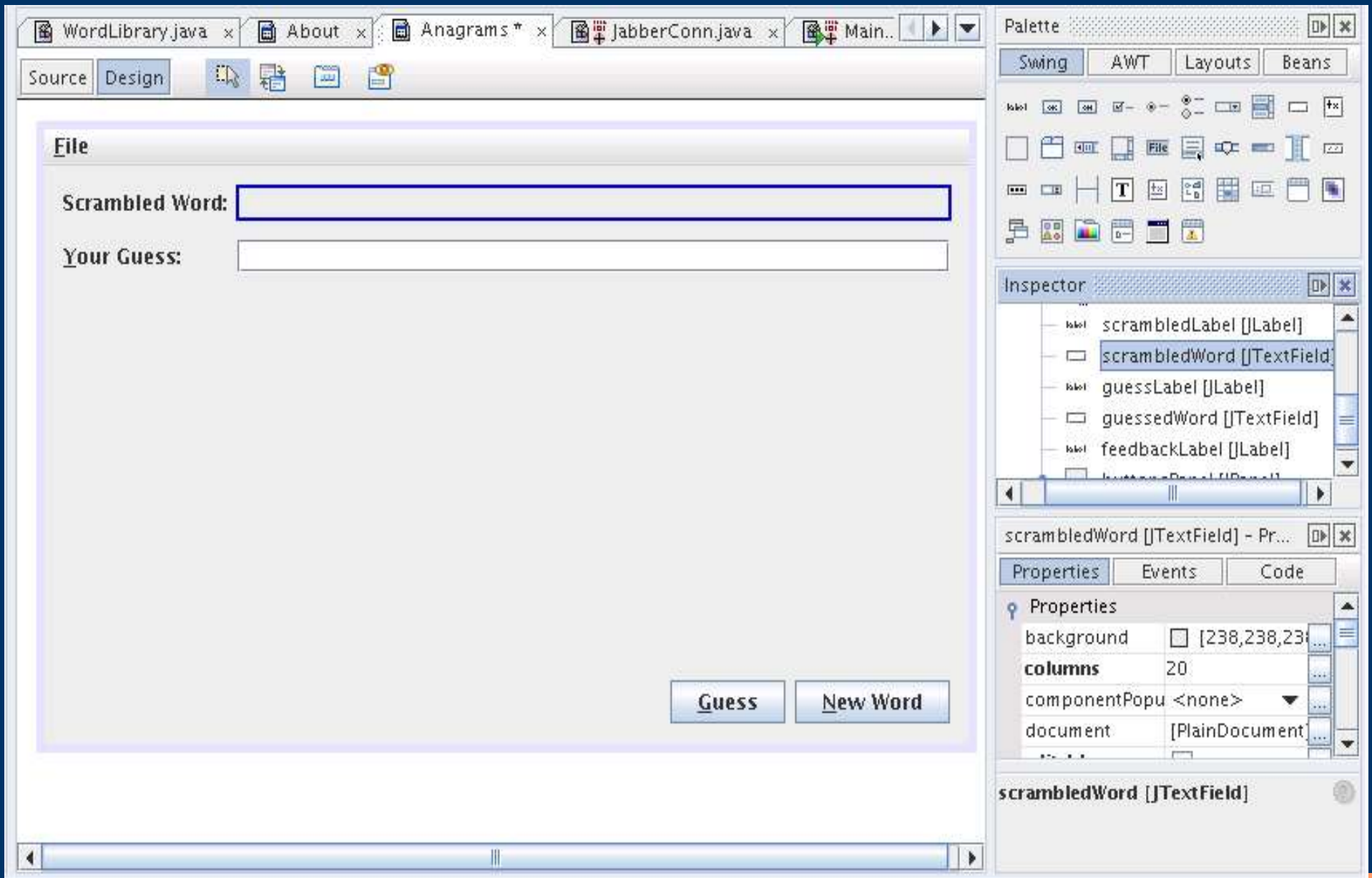


# *Edytor GUI*

- Edytor oparty na WYSIWYG
- Paleta komponentów Swing i AWT
- Możliwość pełnej kontroli generowanego kodu
- Wsparcie JavaBeans



# Edytor GUI



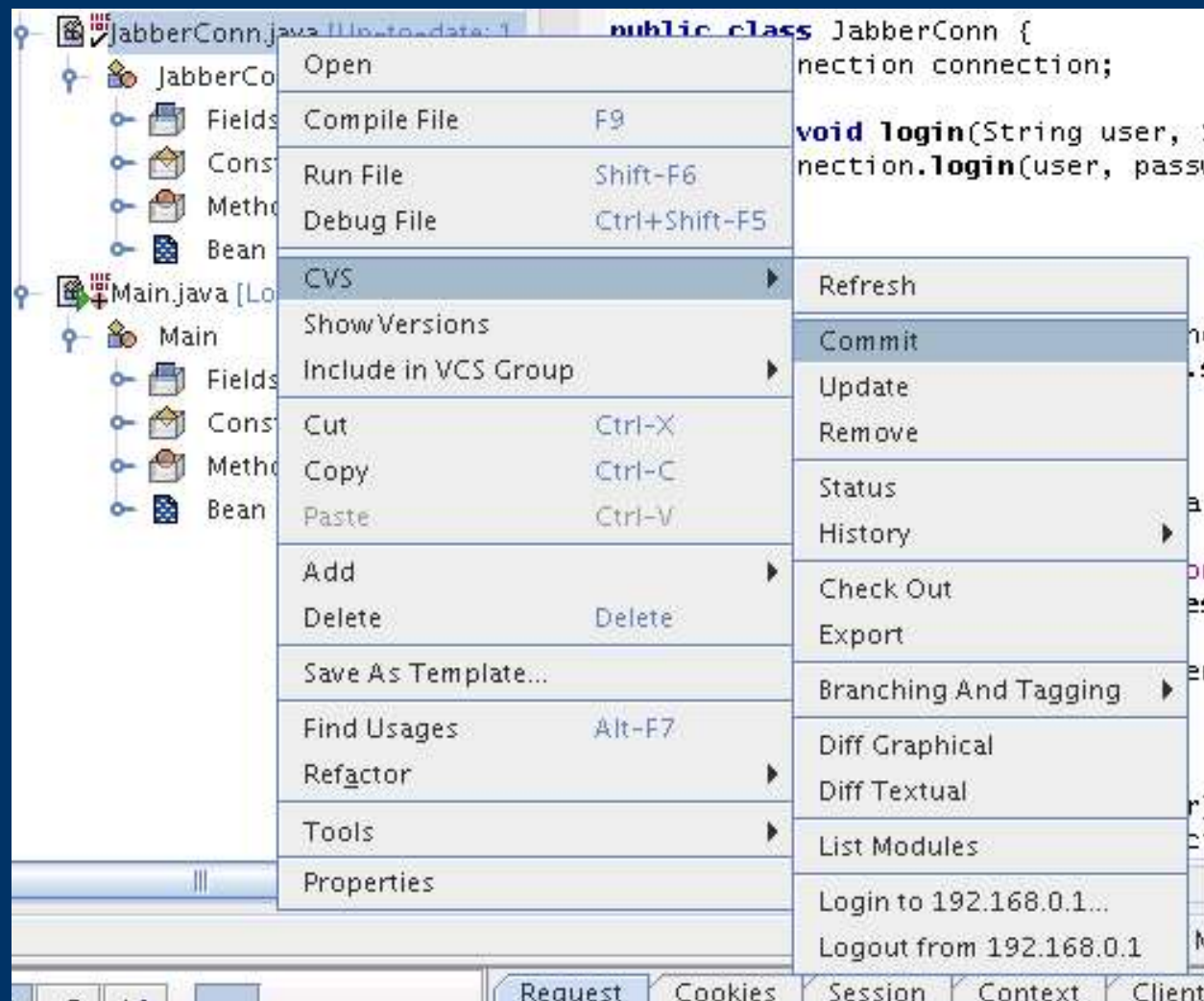
# *Kontrola wersji oprogramowania*

- Wsparcie dla takich systemów kontroli wersji jak CVS i PVCS
- Niezależna od platformy implementacja CVSa
- Intuicyjne GUI
- Przejrzysty graficzny „diff”
- Szybki podgląd na aktualnie zmienione pliki
- Proste zarządzanie projektem w systemie kontroli wersji



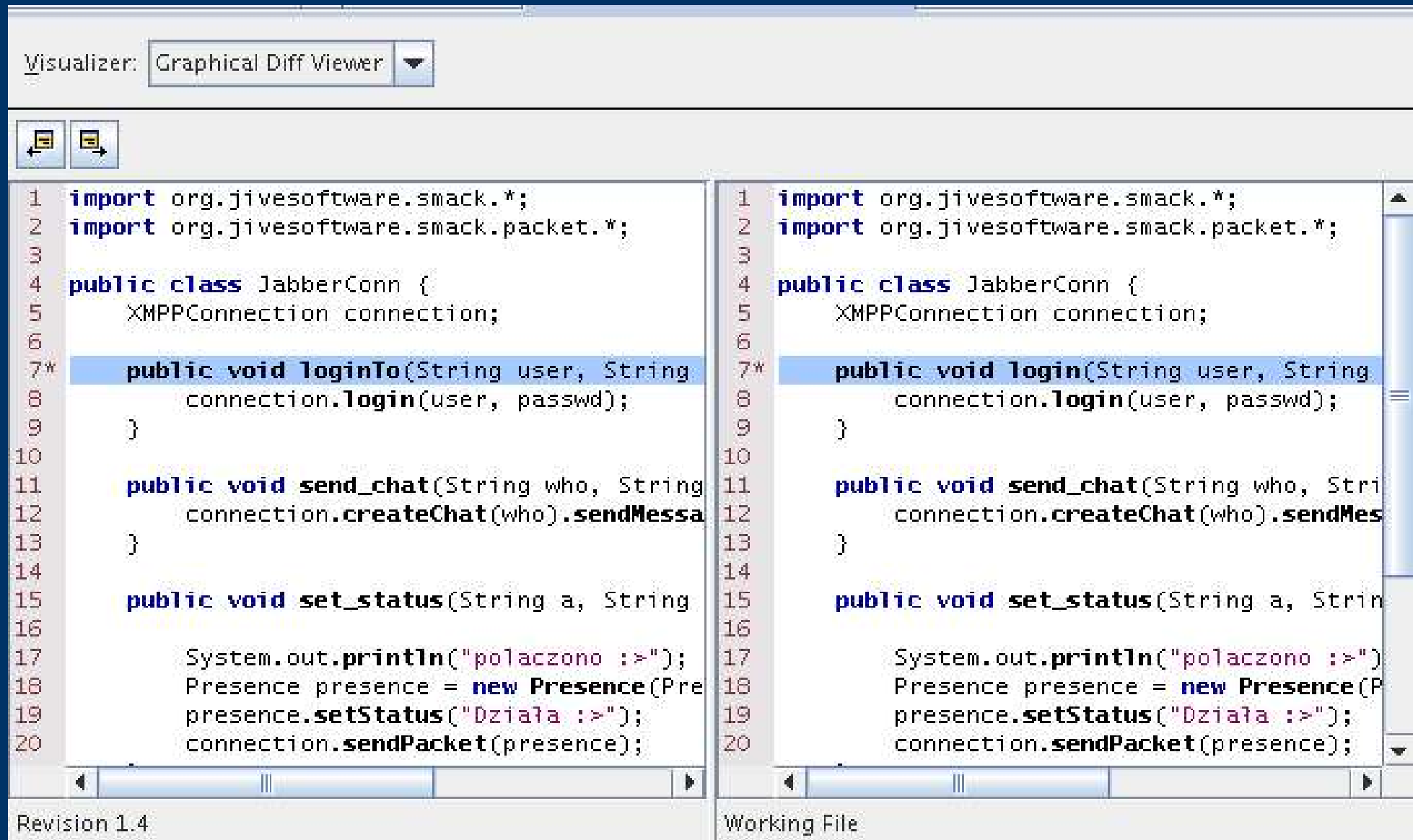


# Kontrola wersji oprogramowania



# Kontrola wersji oprogramowania

Visualizer: Graphical Diff Viewer



```
1 import org.jivesoftware.smack.*;
2 import org.jivesoftware.smack.packet.*;
3
4 public class JabberConn {
5     XMPPConnection connection;
6
7*  public void loginTo(String user, String
8     connection.login(user, passwd);
9 }
10
11 public void send_chat(String who, String
12     connection.createChat(who).sendMessa
13 }
14
15 public void set_status(String a, String
16
17     System.out.println("polaczono :>");
18     Presence presence = new Presence(Pre
19     presence.setStatus("Działa :>");
20     connection.sendPacket(presence);
```

Revision 1.4

```
1 import org.jivesoftware.smack.*;
2 import org.jivesoftware.smack.packet.*;
3
4 public class JabberConn {
5     XMPPConnection connection;
6
7*  public void login(String user, String
8     connection.login(user, passwd);
9 }
10
11 public void send_chat(String who, Stri
12     connection.createChat(who).sendMes
13 }
14
15 public void set_status(String a, Strin
16
17     System.out.println("polaczono :>");
18     Presence presence = new Presence(P
19     presence.setStatus("Działa :>");
20     connection.sendPacket(presence);
```

Working File

# *Debugger*

- Zdalny debugger (remote debugging)
  - “Fix and continue”
  - Obsługa aplikacji wielowątkowych
  - Ukrywanie kodu
  - Obsługa wielu języków programowania w czasie jednej sesji
  - Wsparcie dla komponentów J2EE
- 
-

# *Aplikacje webowe*

- Szablony do szybkiego tworzenia JSP oraz servletów
- Edytor JSP obsługujący podświetlanie kodu, a także parsowanie w czasie edycji
- Możliwość wykonywania aplikacji na wbudowanym serwerze Tomcat
- Wsparcie dla debuggera



# *Refaktoryzacja*

- Move Class
- Rename
- Change Method Signature
- Encapsulate Field



# Refaktoryzacja

- Plany:
  - Extract Interface
  - Pull Up, Push Down Method
  - Override Methods
  - Inline Method
  - Extract Method
  - Move Method
  - Convert Nested to Top-Level Class
  - Extract/Inline Local Variable
  - Hierarchy Browser

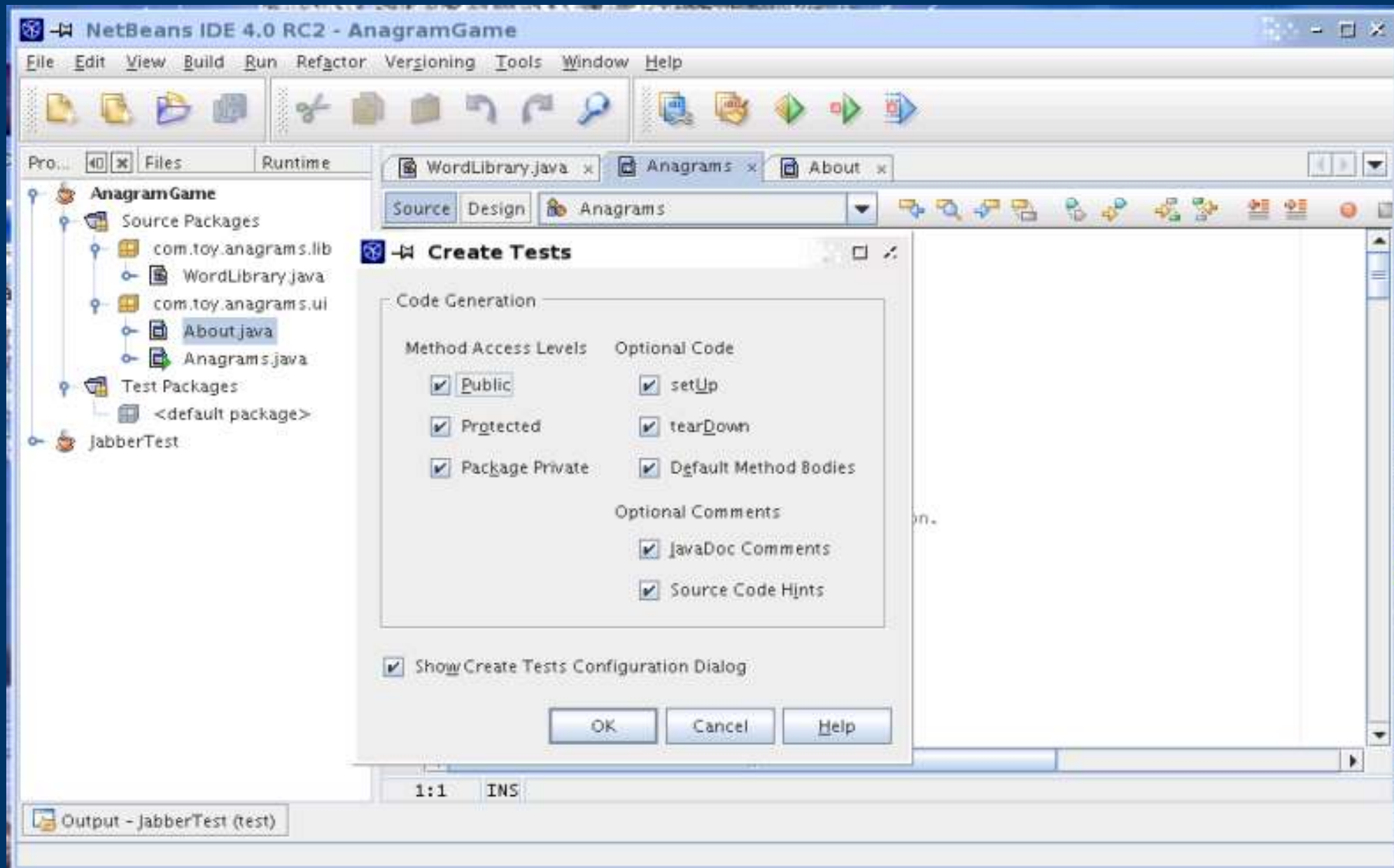


# Testy

- Testowanie programu oparte na JUnit
- Możliwość testowania całego projektu jak i pojedynczej klasy
- Narzędzie wspomagające generację reguł

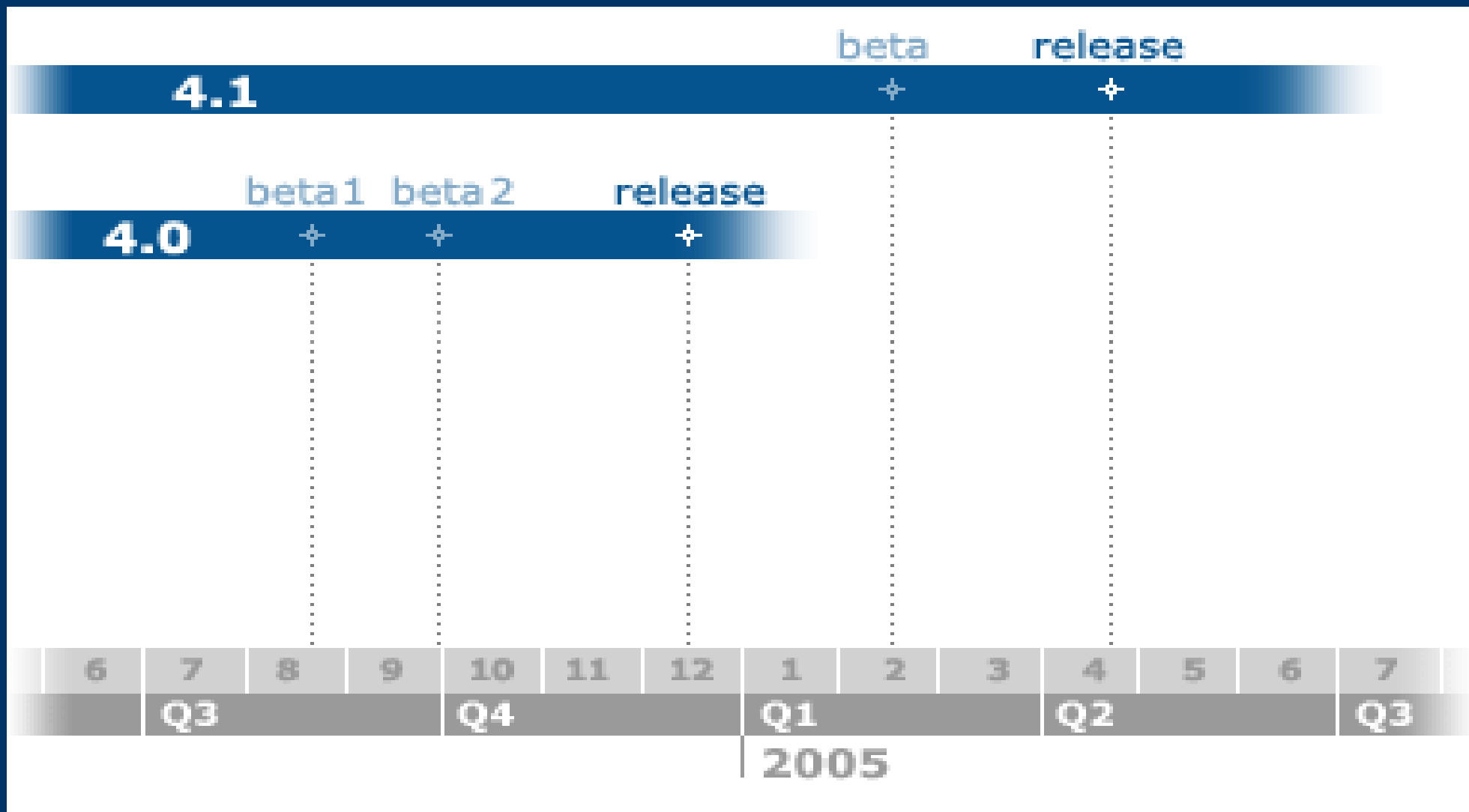


# Testy





# Kierunek Rozwoju



# *Kierunek Rozwoju*

- Rozszerzenie środowiska o obsługę jak największej liczby języków programowania oraz technologii znajdujących się na rynku
  - Utrzymywanie bardzo liberalnej licencji pozwalającej firmom na czerpanie korzyści bez obawy o zmianę warunków dystrybucji NetBeans
  - Rozwój społeczności związanej z projektem
  - Dostosowywanie platformy NetBeans do potrzeb rynku nie tracąc stabilności i zgodności wstecz
- 
-

# *Moduły*

- 25 kategorii modułów
- Łącznie ponad 80 wtyczek
- Rozwijane przez społeczność związaną z ruchem OpenSource

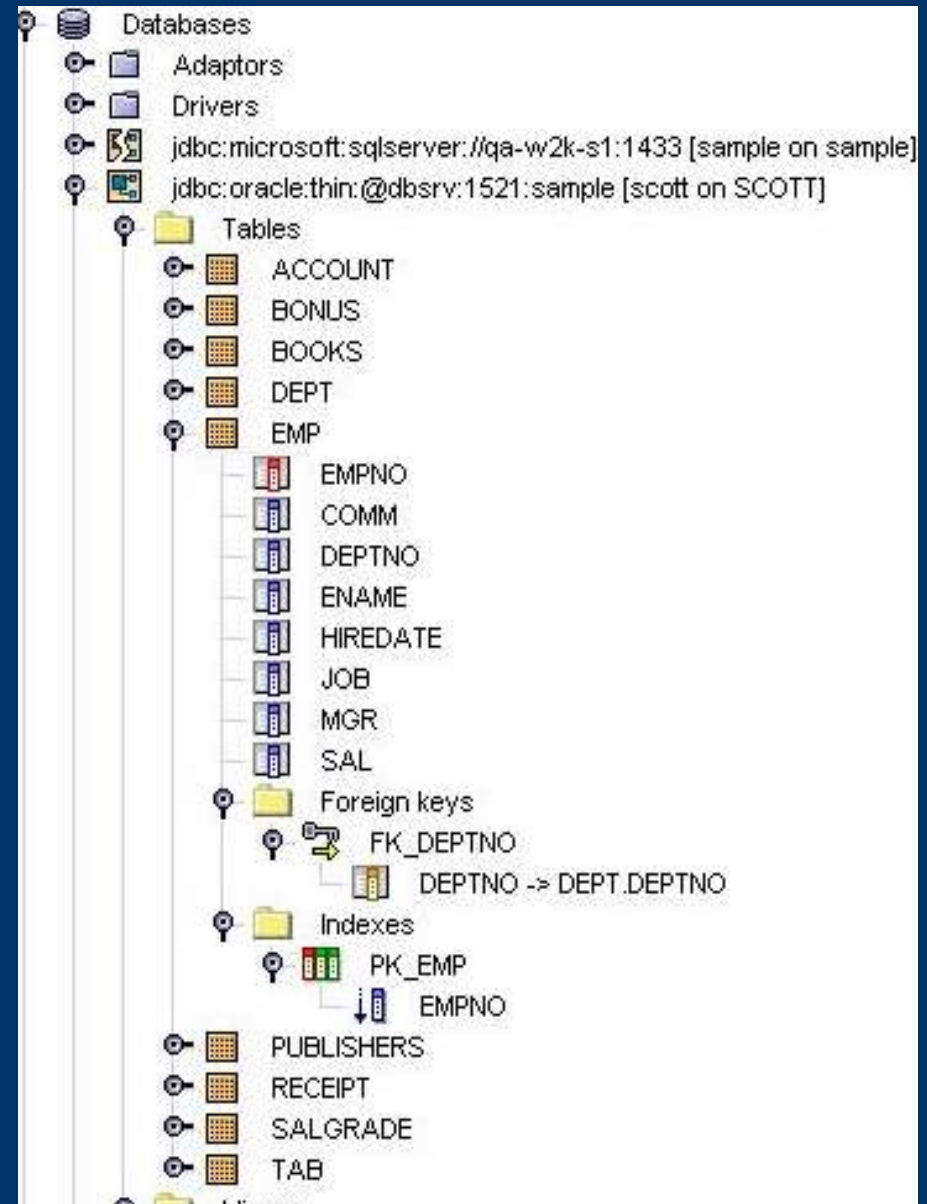


# *Database Explorer*

- Podgląd i edycja schematów bazy danych
  - Możliwość wykonywania poleceń SQL i DDL
  - Dostęp do danych zawartych w tabelach
  - Możliwość przenoszenia tabel pomiędzy bazami danych
  - Połączenie z bazą danych realizowane poprzez JDBC
  - Produkt darmowy
- 
-

# Database Explorer

- Obsługa baz danych:
  - Oracle
  - IBM DB2
  - Microsoft SQL Server
  - PointBase, Sybase
  - Informix, Cloudscape
  - MySQL
  - PostgreSQL
  - InstantDB
  - Interbase
  - Firebird
  - Hypersonic SQL
  - Microsoft Access



# Corba

- Wsparcie dla CORBA IDL polegające na:
  - parsowaniu “w locie”
  - podświetlaniu składni
- Wsparcie dla zewnętrznych kompilatorów
- Generator implementacji
- Interface Repository Browser
- Naming System Browser
- Liczne kreatory wspomagające tworzenie aplikacji wykorzystujących CORBE



# RMI

- Kompilacja oraz uruchamianie
- Przeglądarka rejestru zarówno na lokalnej jak i zdalnej maszynie
- Activation System Browser
- Szablony wspomagające tworzenie nowych obiektów



# RMI

The image shows a screenshot of an IDE interface. On the left is the 'Explorer [Filesystems]' window, and on the right is the 'Properties Window [ATestImpl]'.

**Explorer [Filesystems]**

- Filesystems
  - C:\repository
    - activatable
      - AParam
      - ATest
      - ATestClient
      - ATestImpl**
      - class ATestImpl
      - ATestSetup
      - MarshaledObjectSupport
      - TT

**Properties Window [ATestImpl]**

Arguments

Compiler	External Compilation
Debugger	RMI Debugging
Executor	RMI Executor
Stub compiler	RMI Stub Compiler

Properties Execution RMI Export



# *Inne wtyczki*

- XML
- CPPlite
- UML
- ... i wiele innych



# *Platforma NetBeans*

- Szybkie tworzenie aplikacji poprzez wykorzystanie gotowych składników
- Bardzo liberalna licencja
- Szczegółowa dokumentacja
- Ustandaryzowane API
- Przenośność



# Project XEMO

**XEMO Integrated Composition Environment Prerelease 1.0 (Build 200112062138) [Project Default]**

File Edit Tools View Project Window Help

Explorer (Filesystems)

Filesystems

- Default System
  - Actions
  - Bookmarks
  - Editors
  - Menu Bar
  - Modules
    - org.netbeans-modules-auto
    - org.netbeans-modules-ctrl
    - org.netbeans-modules-hint
    - org.netbeans-modules-lexer
    - org.netbeans-modules-prod
    - org.netbeans-modules-util
    - org-openide-ide
    - org-xemo-notation-viewer
- Filesystems Settings
- Projects
- Services
- Shortcuts
- Starbar
- Templates
- Toolbars
- UI
- Webcode
- Windows

Notation (Notation Viewer)

Web Browser

Location: <http://www.xemo.org/documentation/NotationAPI>

**XEMO NotationAPI Prerelease v1.0**

```
public class NotationPanel
extends javax.swing.JPanel
implements java.util.Observer
```

**All Classes**

- Darkness
- LayoutPoint
- LedgerLines
- Line
- LineSpaceComparator
- MalformedLayout
- MarginConstraint
- NotationApplet
- NotationApplication
- NotationFont
- NotationPanel
- NoteHead
- NoteObject
- PositionComparator
- QuadDemo
- Score
- SelectionController
- ViewObject

NotationPanel is the root view used to integrate a notation application or applet. A NotationPanel encapsulates the layout hierarchy built using layout objects and containers originating from the user interface to the layout hierarchy. state information includes:

- Root model
- Root view
- Root controller
- Global paint parameters

A NotationPanel may be used as any other JPanel as part of a Swing component and a complete interface. NotationPanel itself supports the MVC design pattern via the Observer interface.

Since: XEMO NotationAPI 1.0

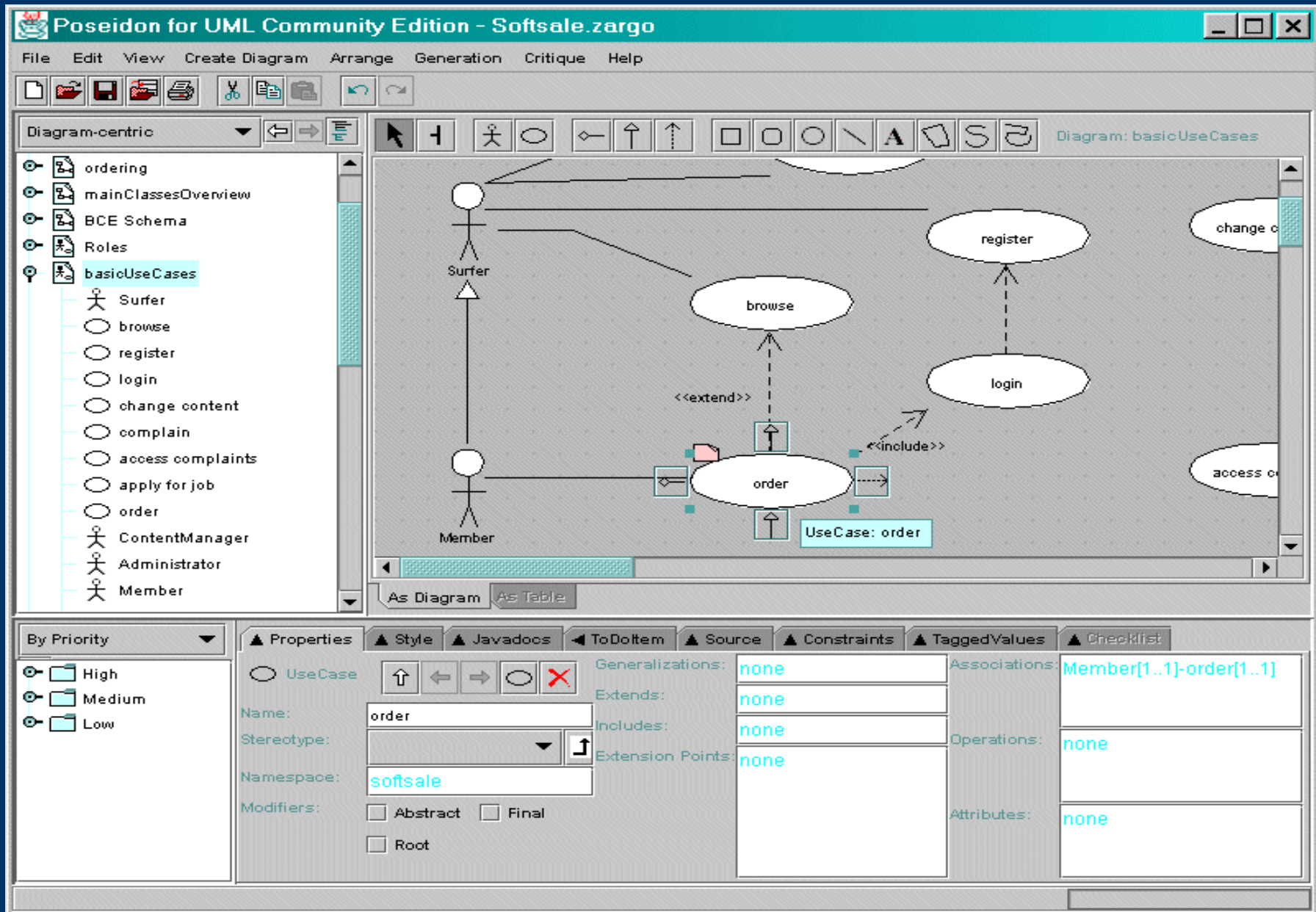
See Also:

© 2000-2001 Kismet Software LLC. All rights reserved.

**PROJECT XEMO™**

www.xemo.org

# Poseidon for UML



# *Konkurencja*

- Eclipse
- Idea
- JCreator
- Borland JBuilder



# Eclipse

Java - TestAsp.aj - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer: AspectGen [192.168.0], RandomTest [192.168.0], Testowa [192.168.0.1]

```
public aspect TestAsp {

    // ===== statystyki wywołań funkcji =====
    Hashtable slw = new Hashtable();

    before() : call(void *(..)) {
        if (!slw.containsKey(thisJoinPoint.getSignature().toString())) {
            slw.put(thisJoinPoint.getSignature().toString(), new Long(1));
        } else {
            long ile = (long)((Long)slw.get(thisJoinPoint.getSignature().toString()).longValue);
            slw.put(thisJoinPoint.getSignature().toString(), new Long(++ile));
        }
    }

    after() : execution (* Testowa.main(..)) {
        System.out.println("Statystyki wywołan metod: ");
        Enumeration e_slw = slw.keys();
        while (e_slw.hasMoreElements()) {
            String klucz = (String)e_slw.nextElement();
        }
    }
}
```

Declaration Javadoc Problems Console

Writable Smart Insert 1:1

# IntelliJ Idea

The screenshot displays the IntelliJ IDEA IDE interface. At the top, the title bar shows the project name "Pet Store.ipr" and the file path. The menu bar includes File, Edit, Search, View, Go To, Code, Refactor, Build, Run, Tools, Window, and Help. Below the menu bar is the Main Toolbar with various icons for file operations and development actions.

The left sidebar contains the Project Structure view, showing a tree of packages and files. The "account" package is expanded, revealing the "ejb" sub-package which contains "AccountEJB", "AccountLocal", and "AccountLocalH".

The main editor window displays the code for "AccountEJB.java". The code includes a catch block for "NamingException" and a "throw new CreateException" statement. A red squiggly line indicates a warning or error at line 96, which is the start of the "ejbPostCreate" method. The code also includes a "setEntityContext" method and an "unsetEntityContext" method.

The bottom tool window shows the "Find - Occurrences of 'main'" search results. It lists several files and their occurrences of the "main" method, including "HttpPostPetStoreProxy.java" and "main" in the "com.sun.j2ee.blueprints.admin.client" package.

The status bar at the bottom shows the current time as 92:10, the current mode as "Insert", and the memory usage as "55M of 59M".

# *Podsumowanie*

- Potężne środowisko dostosowywane do potrzeb programistów
- Wsparcie dla obecnych technologii
- Rozwijane pod skrzydłami SUNa
- Nacisk na modularyzację środowiska

