

Wykład 1-część druga

Wstęp do inżynierii oprogramowania.

Cykle rozwoju oprogramowania

(część biznesowa aplikacji)

iteracyjno-rozwojowy cykl oprogramowania 1

Java

język programowania

- obiektowo zorientowany
- wysokiego poziomu

platforma Javy

- z maszyny wirtualnej VM
- API (interfejs programowania aplikacji).

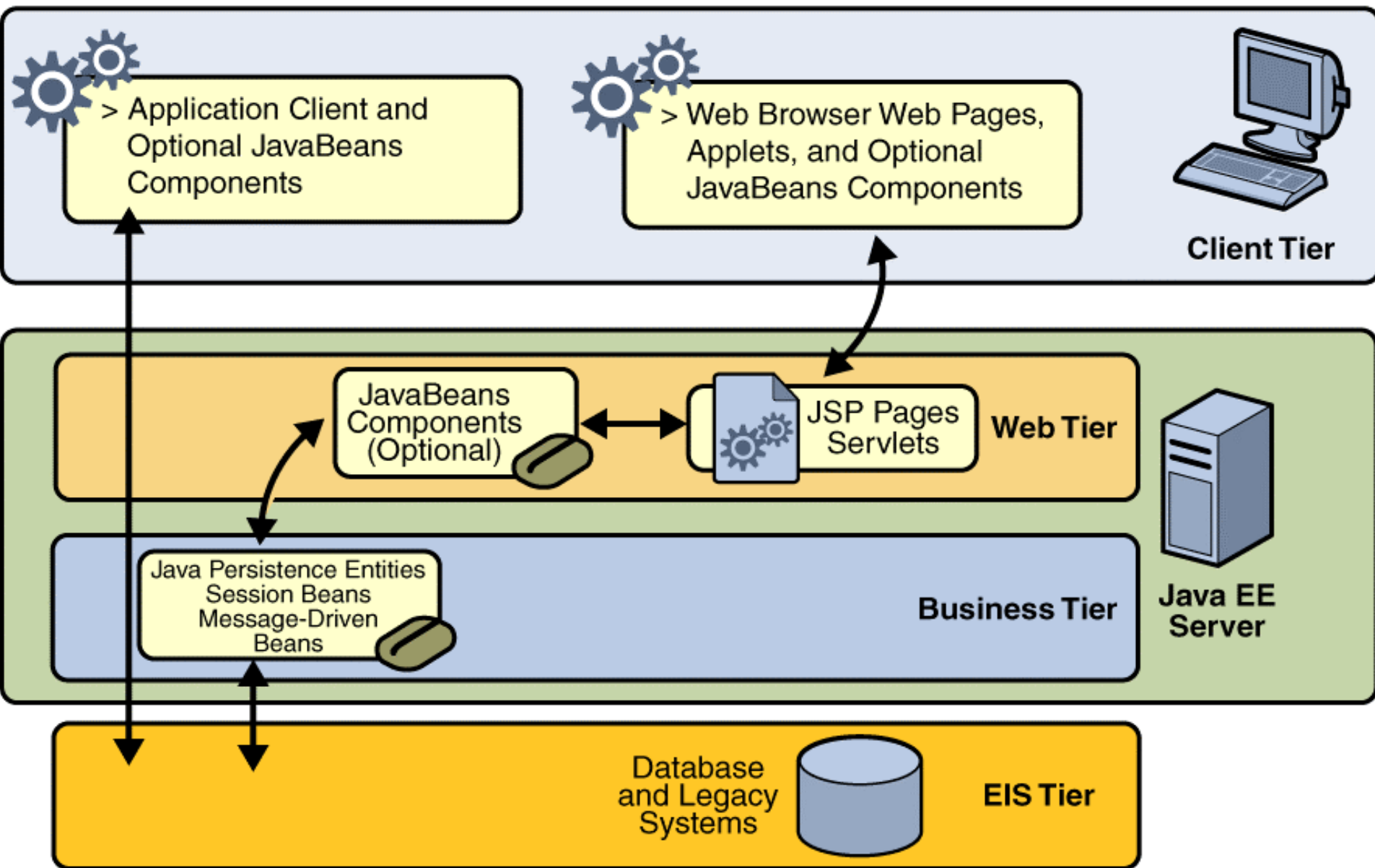
Rezultat

- niezależność od platformy,
- duże możliwości,
- stabilność,
- łatwość rozwoju,
- bezpieczeństwo

Rodzaje platform Javy:

- ◆ Java Platform, Standard Edition (Java SE)
- ◆ Java Platform, Enterprise Edition (Java EE)
- ◆ Java Platform, Micro Edition (Java ME)
- ◆ Java Platform CARD

Warstwy aplikacji (Java EE)



1) Wykonanie projektu typu aplikacja Javy

The image shows the NetBeans IDE 5.5 interface. The main window displays the "Welcome" screen with several sections:

- GETTING STARTED**: Includes links for "Getting Started", "What's New", and "Introduction to Java EE 5".
- ARTICLES & NEWS**: Contains a "Cannot connect to internet." message and a "Proxy Configuration..." button.
- SAMPLE PROJECTS**: Lists "Visual Web", "General", "Identity Blueprints", "Web", and "Enterprise".
- BLOGS**: Contains a "Cannot connect to internet." message and a "Proxy Configuration..." button.

The "Dynamic Help" panel on the right shows "Finding Information About the". The "Output" window at the bottom displays several tabs for "AplikacjaInternetowa1 (run)" and "Sun Java System Application Server 9".

The "File" menu is open, showing options such as "New Project...", "New File...", "Open Project...", "Save", "Print...", and "Exit".

The Windows taskbar at the bottom shows the Start button and several open applications: NetBeans IDE 5.5, Windows Commander..., Microsoft PowerPoint..., and UML Modeling: Devel... The system tray on the right shows the date and time as 22:39.



Steps

1. Choose Project

2. ...

Choose Project

Categories:

- General
- UML
- Web
- Enterprise
- Service Oriented Architecture
- NetBeans Plug-in Modules
- Samples

Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Project with Existing Ant Script

Description:

Creates a new J2SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

< Back Next > Finish Cancel Help

New Java Application [Close]

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Set as Main Project

Create Main Class



Projects | **Files** | **Runtime**

- Katalog1
 - Source Packages
 - katalog1
 - Uchwyty.java
 - Test Packages
 - Libraries
 - Test Libraries

Outline | **Navigator - Uc...**

Members View

- Uchwyty()
- main(String[] args)

Filters:

```

Welcome x Uchwyty.java x
/*
 * Uchwyty.java
 *
 * Created on 2 marzec 2007, 22:44
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package katalog1;

/**
 *
 * @author Kruczkiewicz
 */
public class Uchwyty {

    /** Creates a new instance of Uchwyty */
    public Uchwyty() {
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
1:1 INS
  
```

Uchwyty.java - Properties

Properties

| | |
|-----------|----------------|
| Name | Uchwyty |
| All Files | C:\Document... |
| File Size | 515 |

Text

Encoding

Classpaths

| | |
|-------------------|----------------|
| Compile Classpath | |
| Runtime Classpath | C:\Document... |
| Boot Classpath | C:\Program ... |

Uchwyty.java

Dynamic Help

- About the Source Editor
- See Also

Output

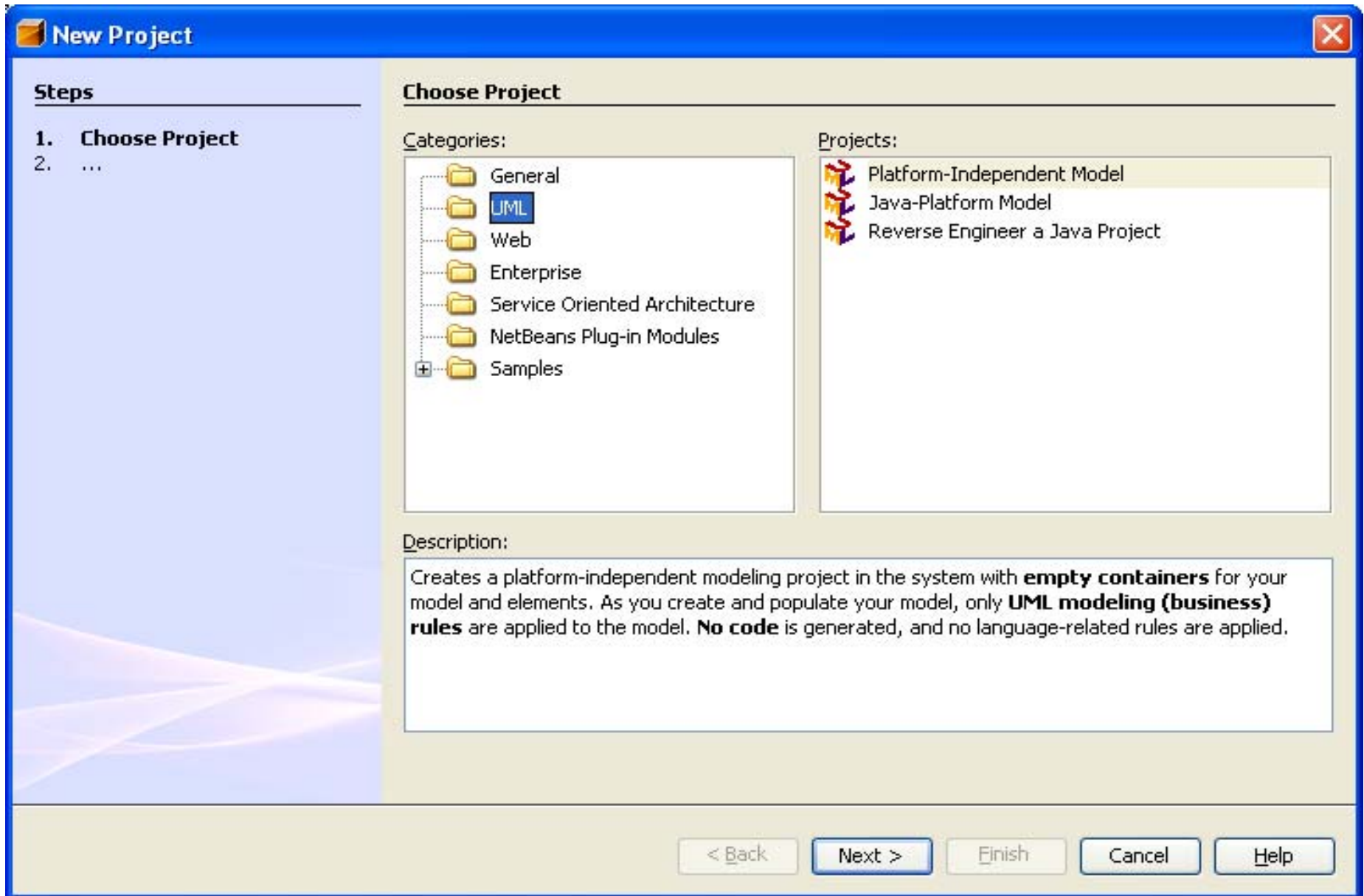
AplikacjaInternetowa1 (run) x Java DB Database Process x Sun Java System Application Server 9 x

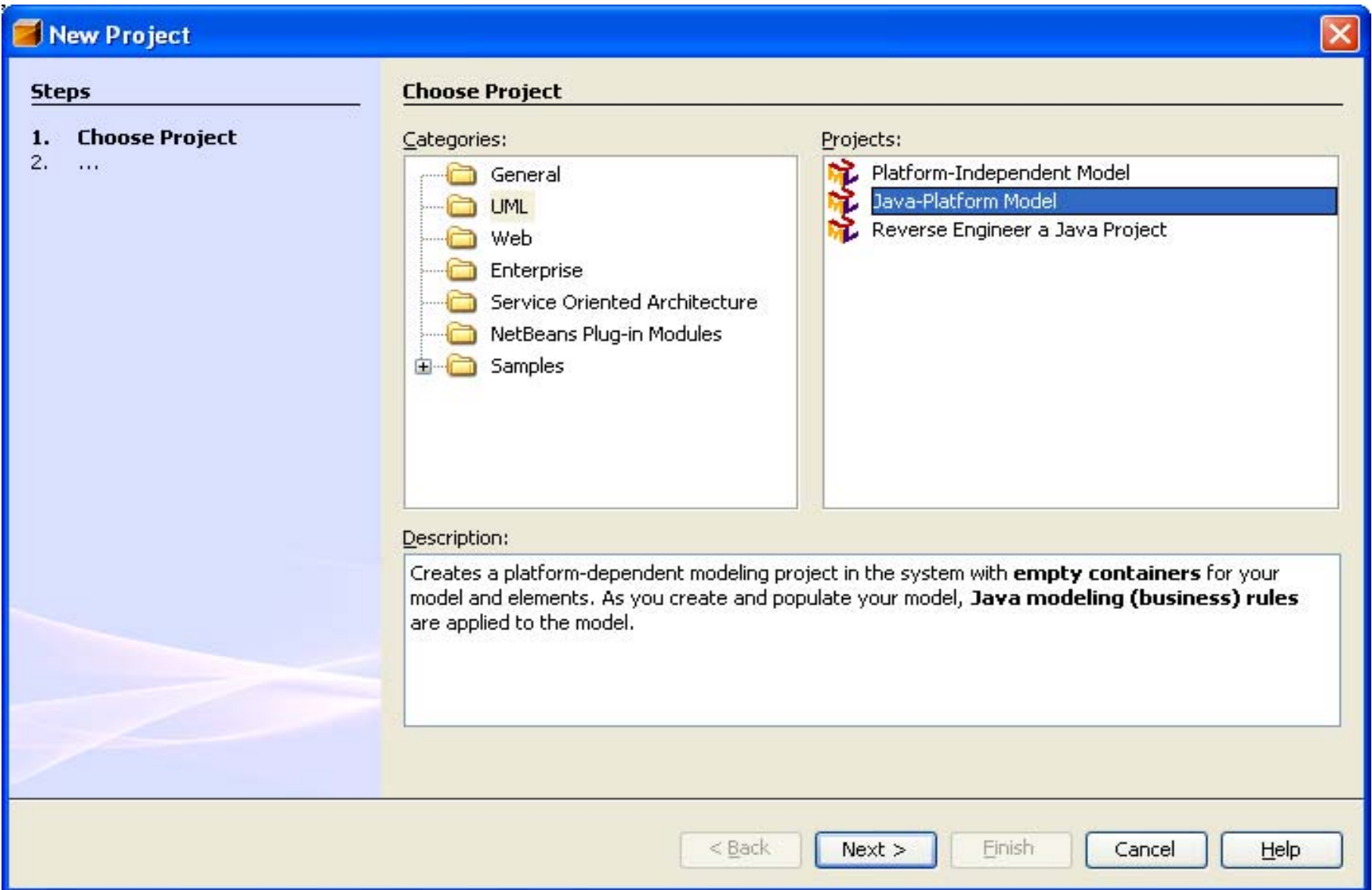
AplikacjaInternetowa1 (run) x AplikacjaInternetowa1 (run) x AplikacjaInternetowa1 (run) x AplikacjaInternetowa1 (run) x AplikacjaInternetowa1 (run) x

Searching server sun java system application server 9



2) Utworzenie projektu UML







Steps

1. Choose Project
2. **Name and Location**

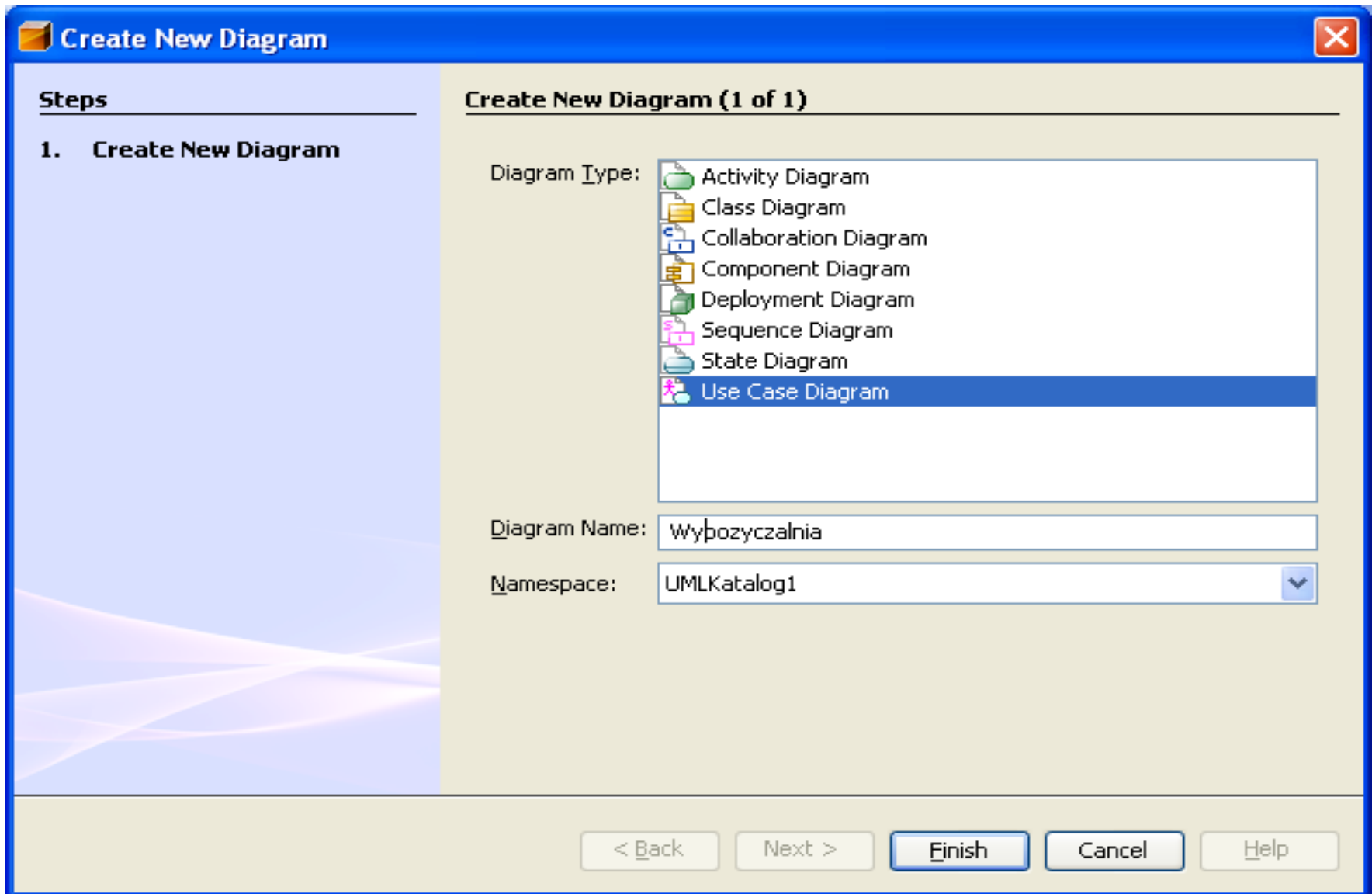
Name and Location

Project Name:

Project Location:

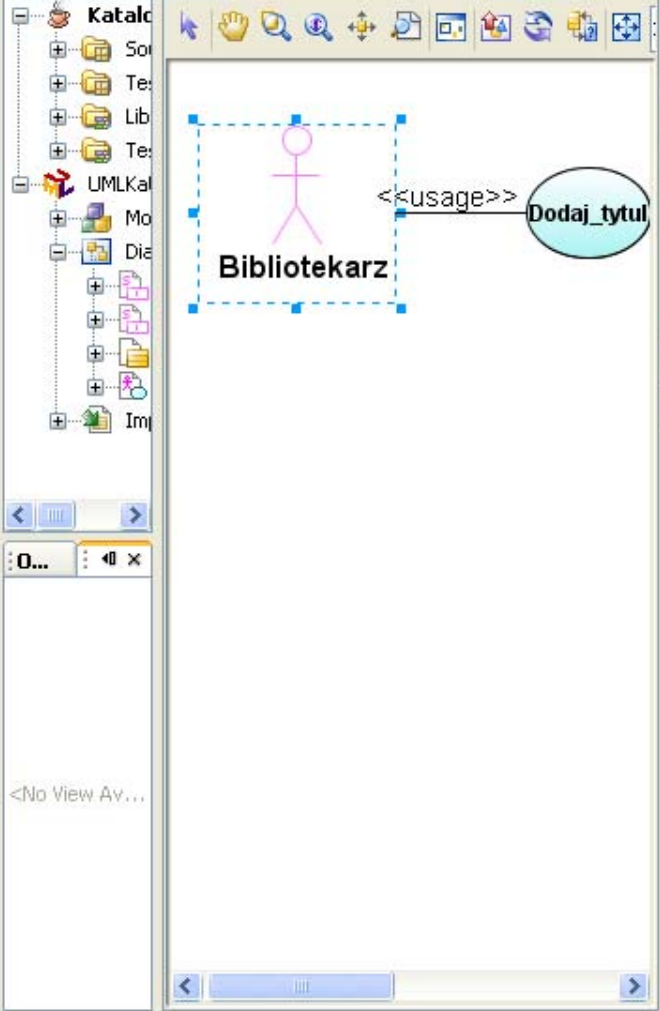
Project Folder:

3) Utworzenie diagramu przypadków użycia (use cases)





...me Wypożyczalnia * x



Palette

- Basic**
 - Use Case
 - Package
 - Actor
 - Include
 - Extend
 - Generalization
 - Association
- Dependencies**
 - Dependency
 - Realize
 - Usage
 - Permission
 - Abstraction
- Comments**
 - Comment
 - Link Comment

UML Documentation

b i u

1. System powinien wspierać wypożyczanie książek
2. Biblioteka wypożycza podane książki i czasopisma osobom zarejestrowanym, o ile je posiada
3. Biblioteka dokonuje zakupu nowych książek, przy czym popularne książki kupuje w kilku egzemplarzach. Usuwa zniszczone książki i czasopisma.
4. Bibliotekarz jest pracownikiem biblioteki, komunikuje się z wypożyczającym. Jego praca jest wspierana za pomocą systemu
5. Wypożyczający może zarezerwować książkę lub czasopismo, które nie jest dostępne w danej chwili. W momencie, kiedy zamówione rzeczy są dostępne- albo po zwrocie lub dzięki zakupowi, można je wypożyczyć i usunąć rezerwację. Rezerwację można usunąć niezależnie.
6. Biblioteka może łatwo utworzyć, zmienić i usunąć informację o tytułach, wypożyczających, wypożyczeniach i rezerwacjach
7. System powinien pracować w popularnych systemach (UNIX, Windows, OS/2) i powinien mieć nowoczesny graficzny interfejs użytkownika
8. System powinien się rozwijać np. wprowadzenie możliwości zawiadamiania rezerwującego książkę o jej dostępności lub dłużnika o przekroczeniu terminu wypożyczenia

Output

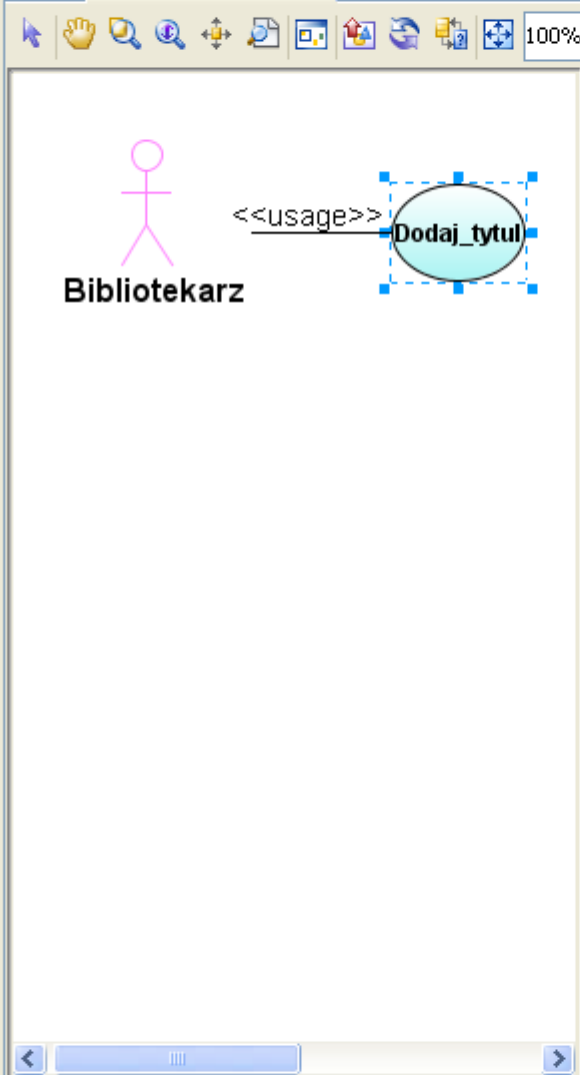


Files Runtime

Katalog1

- Source Packages
- Test Packages
- Libraries
- Test Libraries
- UMLKatalog1
- Model
- Diagrams
- Imported Elements

Wypożyczalnia * x



Palette

Basic

Use Case Package Actor Include Extend
 Generalization Association

Dependencies

Dependency Realize Usage Permission Abstraction

Comments

Comment Link Comment

Dodaj_tytul - UML Documentation

b *i* u

Przypadek użycia - Dodaj_dane_tytulu

1. Podaj dane tytułu książki: tytuł, nazwisko i imię autora, ISBN, wydawnictwo, rok wydania
2. Sprawdź, czy taki tytuł nie istnieje wśród tytułów książek w Wypożyczalni
3. Dodaj nowy tytuł książki, jeśli nie istnieje w Wypożyczalni

Dodaj_tytul - Properties

Use Case

| | |
|------------|-------------|
| Name | Dodaj_tytul |
| Alias | Dodaj_tytul |
| Visibility | public |

Dodaj_tytul

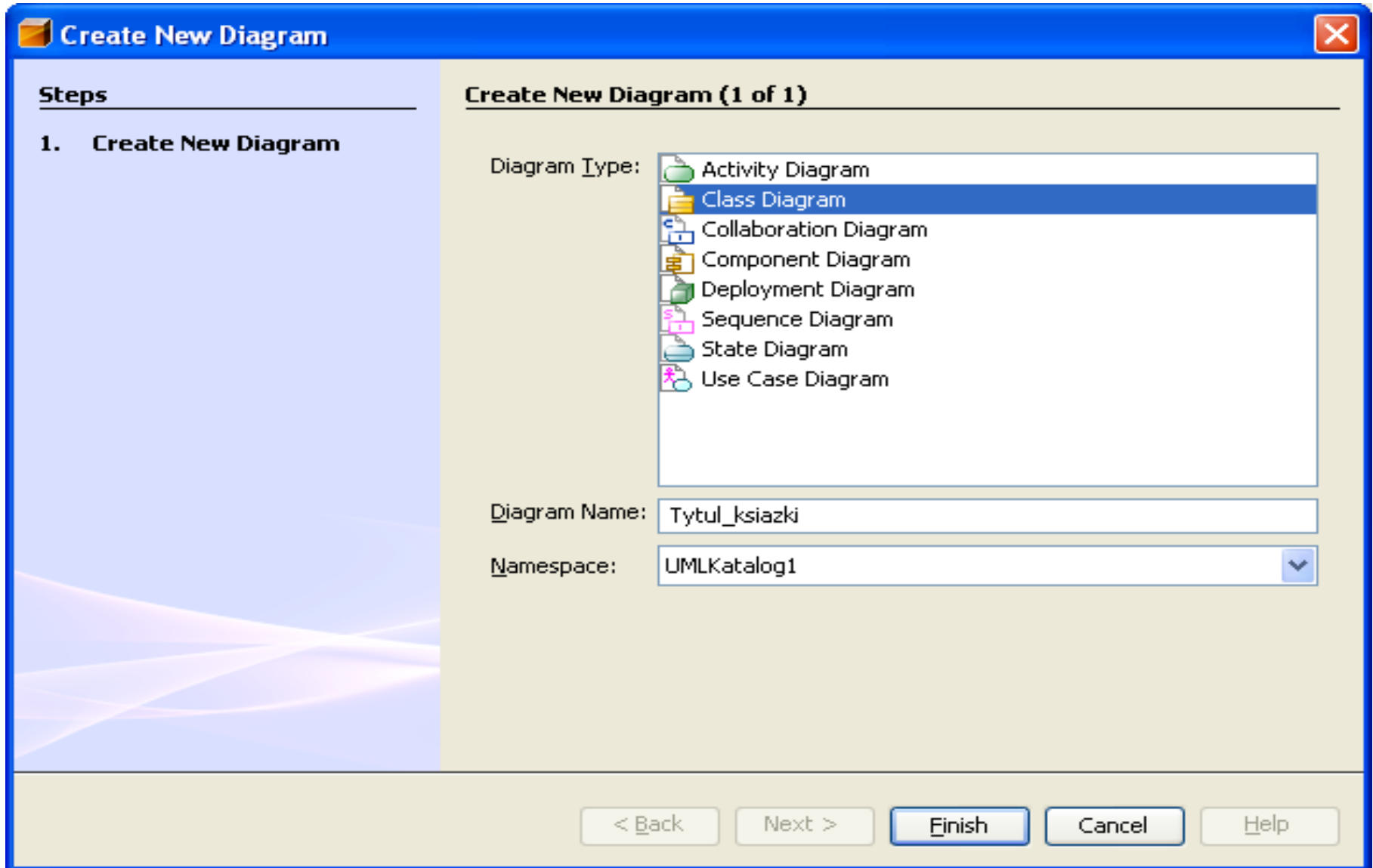
Dynamic Help

Documentation Window (UML Modeling)
 See Also

Navigator

<No View Available>

4) Utworzenie diagramu klas



5) Wstawienie nowej klasy

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a UML class diagram editor with a single class box labeled "Unnamed". The class box is divided into three sections: "Attributes", "Operations", and "Operations". The "Outline" and "Navigator" views at the bottom left show the project structure, including a package named "UMLKatalog1" containing a "Model" package with several classes and diagrams. The "Palette" view on the right shows the "Basic" palette with "Class" and "Interface" options. The "Dynamic Help" view at the bottom right shows "Finding Information About the". The "Output" view at the bottom shows several running processes, including "ApplikacjaInternetowa1 (run)", "Java DB Database Process", and "Sun Java System Application Server 9".

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Projects Files Runtime

Source Packages

- katalog1
 - Uchwyt.java
- Test Packages
- Libraries
- Test Libraries

UMLKatalog1

- Model
 - Tytul_książki
 - Wypożyczalnia
 - Unnamed
 - Bibliotekarz
 - Doda_tytul
- Diagrams
 - Tytul_książki
 - Wypożyczalnia

Outline Navigator

<No View Available>

Output

ApplikacjaInternetowa1 (run) x Java DB Database Process x Sun Java System Application Server 9 x ApplikacjaInternetowa1 (run) x

ApplikacjaInternetowa1 (run) x ApplikacjaInternetowa1 (run) x ApplikacjaInternetowa1 (run) x ApplikacjaInternetowa1 (run) x

Save All finished.

6) Wstawienie atrybutów

The screenshot displays the NetBeans IDE 5.5 interface. The central workspace shows a UML class diagram for the class `Tytul_książki`. The class is divided into three sections: **Attributes**, **Operations**, and **Associations**. The **Attributes** section contains the attribute `private String wydawnictwo`, which is currently selected. The **Operations** section contains the following methods:

```
public Tytul_książki( )  
public String getWydawnictwo( )  
public void setWydawnictwo( String val )
```

The **Palette** on the right side of the IDE shows the **Class** element selected. Below the palette, the **Dynamic Help** section is visible, displaying the text "Finding Information About the".

The **Projects** and **Files** view on the left shows the project structure, including the `UMLKatalog1` project and the `Model` package containing the `Tytul_książki` class.

The **Output** window at the bottom shows several running processes, including `AplikacjaInternetowa1 (run)`, `Java DB Database Process`, and `Sun Java System Application Server 9`.

A status bar at the bottom left indicates "Tytul_książki saved."

7) Wstawienie nowego atrybutu - menu

The screenshot displays the NetBeans IDE 5.5 interface. The main window shows a UML class diagram for the class `Tytul_książki`. The class has two compartments: **Attributes** and **Operations**. The **Attributes** compartment contains the attribute `private String wydawnictwo`, which is currently selected. The **Operations** compartment contains the methods `public Tytul_książki ()`, `public String getWydawnictwo()`, and `public void setWydawnictwo(String)`. A context menu is open over the selected attribute, listing the following options: **Insert Attribute** (Alt+Shift+A), **Delete Attribute**, **Edit**, **Transform**, **Compartment**, **Reset Edges**, **Resize Element to Contents**, **Synchronize Element with Data**, **Hide**, **Show**, **Class**, **Apply Design Pattern...**, **Associate With...**, **Select in Model** (Ctrl+Shift+S), and **Properties**.

The left sidebar shows the **Projects** and **Files** view, with the **UMLKatalog1** project selected. The **UMLKatalog1** project structure includes a **Model** package containing the `Tytul_książki` class, `Wypożyczalnia` class, and `int`, `String`, `void` types. The **Dynamic Help** panel at the bottom right shows the text "Finding Information About the".

Tytul_książki saved.

8) Edycja atrybutu

The screenshot displays the NetBeans IDE 5.5 interface. The central workspace shows a UML class diagram for the class **Tytul_książki**. The class is divided into three sections: **Attributes**, **Operations**, and **Associations**. The **Attributes** section contains a single attribute: `private int Unnamed`. The **Operations** section contains three methods: `public Tytul_książki()`, `public String getWydawnictwo()`, and `public void setWydawnictwo(String val)`. The **Associations** section is currently empty. The **Palette** on the right side of the IDE lists various UML elements: Class, Interface, Package, Collaboration Lifeline, and Enumeration. Below the palette, the **Tytul_książki - UML Doc...** window is visible, showing formatting options for the class diagram. The **Tytul_książki - Properties** window shows the class's name, alias, and visibility (public). The **Dynamic Help** window is also visible, showing the text "Finding Information About the". The **Output** window at the bottom shows several instances of "AplikacjaInternetowa1 (run)" and "Java DB Database Process". The **Navigator** window shows the project structure, including the "UMLKatalog1" project and the "Tytul_książki" class.

Tytul_książki saved.

9) Właściwości nowego atrybutu

The screenshot displays the NetBeans IDE 5.5 interface. The central workspace shows a UML class diagram for the class **Tytul_książki**. The class is divided into two sections: *Attributes* and *Operations*. The *Attributes* section lists the following attributes:

- private String wydawnictwo
- private String ISBN
- private String tytul
- private String nazwisko
- private String imie** (highlighted in blue)

The *Operations* section lists the following methods:

- public Tytul_książki()
- public String getWydawnictwo()
- public void setWydawnictwo(String val)
- public String getISBN()
- public void setISBN(String val)
- public String getTytul()
- public void setTytul(String val)
- public String getNazwisko()
- public void setNazwisko(String val)
- public String getImie()
- public void setImie(String val)

On the right side of the IDE, the **Palette** shows the class hierarchy (Class, Interface, Package, Collaboration Lifeline, Enumeration). Below it, the **imie - Properties** window shows the following details for the attribute:

| imie - Properties | |
|-------------------|--------|
| Attribute | |
| Name | imie |
| Alias | imie |
| Type | String |

At the bottom of the IDE, the **Dynamic Help** window is visible, showing links for "Using the IDE Help System" and "See Also". The taskbar at the bottom shows several instances of "AplikacjaInternetowa1 (run)" and other applications like "Windows Commander...", "Microsoft PowerPoint...", and "UML Modeling: Devel...".

Tytul_książki saved.

10) Generowanie kodu klasy **Tytul_książki** do projektu **Katalog1**

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help



Projects Files Runtime

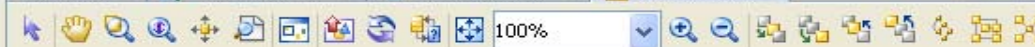
Katalog1

- Source Packages
 - katalog1
 - Uchwyty.java
- Test Packages
- Libraries
- Test Libraries
- UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - Tytul_książki
 - int
 - String
 - void
 - Bibliotekarz
 - Doda_tytul
 - Diagrams
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements

Outline Navigator

<No View Available>

Welcome x Uchwyty.java x Wypożyczalnia x Tytul_książki x



- Edit
- Transform
- Compartment
- Reset Edges
- Resize Element to Contents
- Synchronize Element with Data
- Hide
- Show
- Class
- Redefine Operations...
- Generate Dependency Diagram
- Apply Design Pattern...
- Navigate To Source
- Generate Code...**
- Associate With...
- Select in Model Ctrl+Shift+5
- Properties

```
public String getNazwisko( )
public void setNazwisko( String val )
public String getImie( )
public void setImie( String val )
```

Palette

- Class
- Interface
- Package
- Collaboration Lifeline
- Enumeration
- Node

Tytul_książki - UML Doc...



Tytul_książki - Properties

| Class | |
|-------------|---------------|
| Name | Tytul_książki |
| Alias | Tytul_książki |
| Visibility | public |
| Stereotypes | |

Tytul_książki

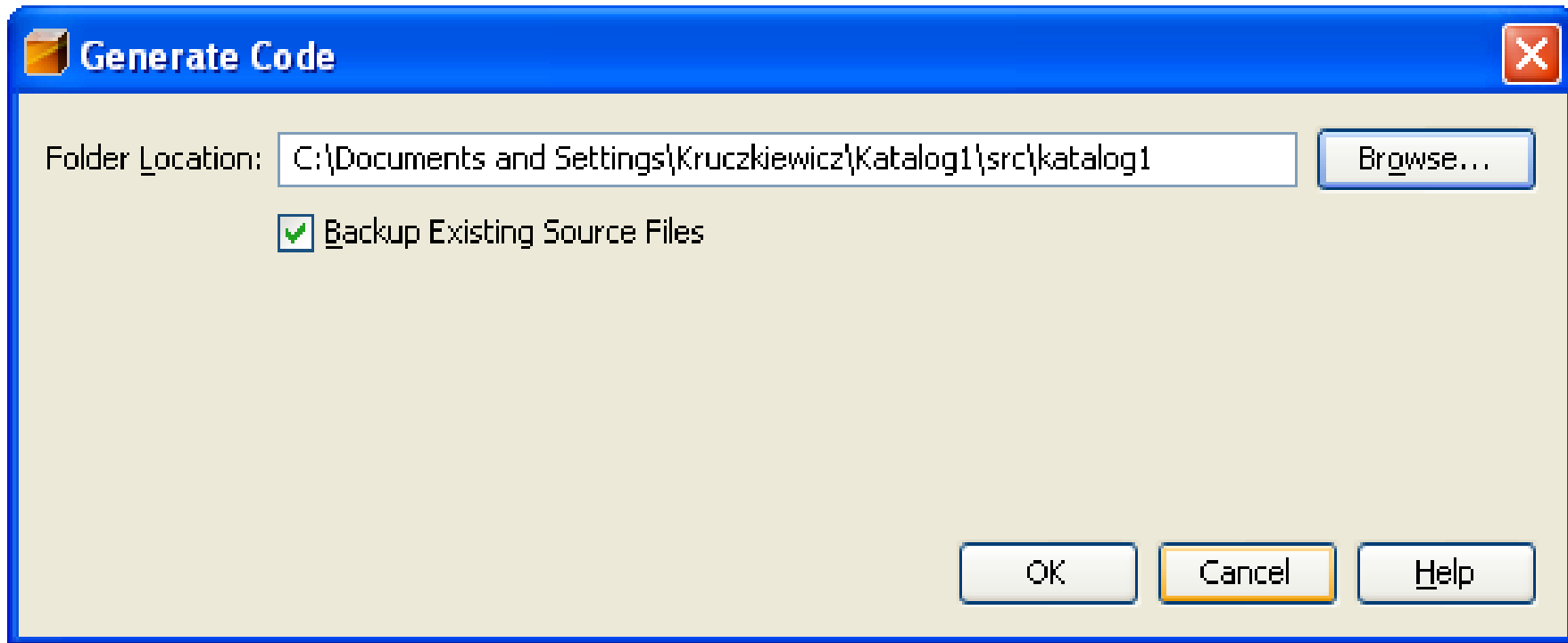
Dynamic Help

Finding Information About the

Output

- AplikacjaInternetowa1 (run)
- Java DB Database Process
- Sun Java System Application Server 9
- AplikacjaInternetowa1 (run)
- AplikacjaInternetowa1 (run)
- AplikacjaInternetowa1 (run)
- AplikacjaInternetowa1 (run)
- Generate Code Log

11) Wybór projektu Javy jako miejsca do generowania kodu



NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Projects: Katalog1
 Files: katalog1, Tytul_książki.java, Uchwyt.java
 Runtime: Welcome, Uchwyt.java, Wypożyczalnia, Tytul_książki

UML Diagram: Tytul_książki

```

classDiagram
    class Tytul_książki {
        private String wydawnictwo
        private String ISBN
        private String tytul
        private String nazwisko
        private String imie
        Tytul_książki()
        String getWydawnictwo()
        void setWydawnictwo(String val)
        String getISBN()
        void setISBN(String val)
        String getTytul()
        void setTytul(String val)
        String getNazwisko()
        void setNazwisko(String val)
        String getImie()
        void setImie(String val)
    }
  
```

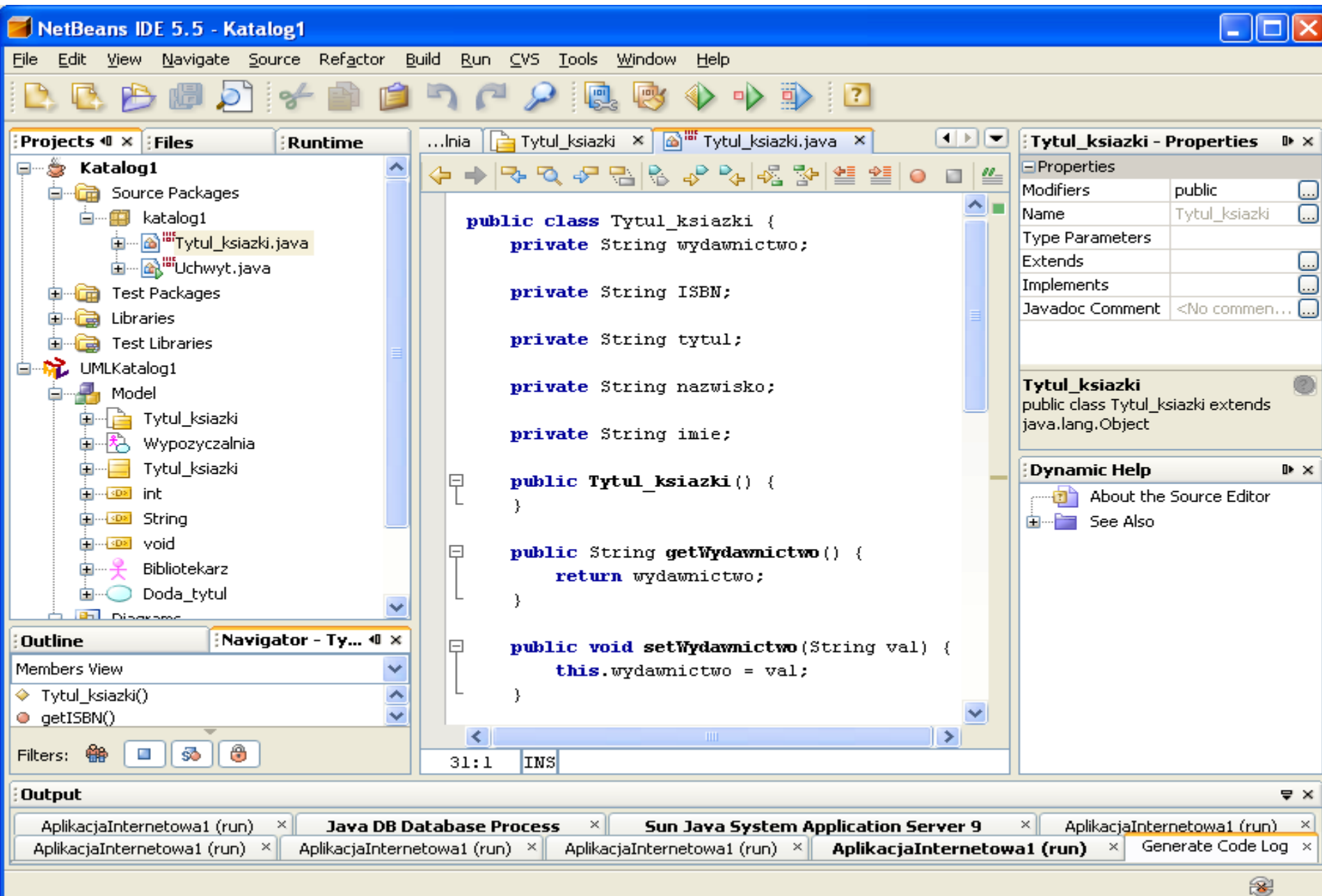
Palette: Class, Interface, Package, Collaboration Lifeline, Enumeration, Node

Tytul_książki - UML Doc...
 Tytul_książki - Properties
 Class: Tytul_książki
 Name: Tytul_książki
 Alias: Tytul_książki
 Visibility: public
 Stereotypes:

Dynamic Help: Finding Information About the

Output: AplikacjaInternetowa1 (run), Java DB Database Process, Sun Java System Application Server 9, AplikacjaInternetowa1 (run), Generate Code Log

12) Wygenerowany kod



The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code for the class `Tytul_książki`. The code is as follows:

```
public class Tytul_książki {  
    private String wydawnictwo;  
  
    private String ISBN;  
  
    private String tytul;  
  
    private String nazwisko;  
  
    private String imie;  
  
    public Tytul_książki() {  
    }  
  
    public String getWydawnictwo() {  
        return wydawnictwo;  
    }  
  
    public void setWydawnictwo(String val) {  
        this.wydawnictwo = val;  
    }  
}
```

The IDE interface includes a Project Explorer on the left showing the project structure, an Outline window at the bottom left, and a Properties window on the right showing the class name and its superclass (`java.lang.Object`). The status bar at the bottom indicates the current cursor position at line 31, column 1.

```

public class Tytul_ksiazki {
    private String wydawnictwo;
    private String ISBN;
    private String tytul;
    private String nazwisko;
    private String imie;
    public Tytul_ksiazki()           {   }
    public String getWydawnictwo()  { return wydawnictwo;   }
    public void setWydawnictwo(String val) { this.wydawnictwo = val; }
    public String getISBN()         { return ISBN;       }
    public void setISBN(String val)  { this.ISBN = val;   }
    public String getTytul()        { return tytul;     }
    public void setTytul(String val) { this.tytul = val;  }
    public String getNazwisko()     { return nazwisko;   }
    public void setNazwisko(String val) { this.nazwisko = val; }
    public String getImie()        { return imie;       }
    public void setImie(String val)  { this.imie = val;   }
}

```

13) Wstawienie klasy fasadowej **Uchwy** – będzie zastosowana do obsługi wywołań przypadków użycia.

The screenshot shows the NetBeans IDE 5.5 interface. The main window displays two class diagrams side-by-side. The left diagram is for the class **Uchwy**, which has no attributes and one operation: `public Uchwy()`. The right diagram is for the class **Tytuł_książki**, which has four private attributes: `String wydawnictwo`, `String ISBN`, `String tytuł`, and `String nazwisko`. It also has eight public operations: `public Tytuł_książki()`, `public String getWydawnictwo()`, `public void setWydawnictwo(String val)`, `public String getISBN()`, `public void setISBN(String val)`, `public String getTytuł()`, `public void setTytuł(String val)`, `public String getNazwisko()`, `public void setNazwisko(String val)`, `public String getImie()`, and `public void setImie(String val)`.

The interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Build, Run, CVS, Tools, Window, Help), a toolbar, and several panels: Projects, Files, Runtime, Palette, Diagram, Dynamic Help, Outline, Navigator, and Output. The Output panel shows several instances of 'AplicacjaInternetowa1 (run)' and 'Java DB Database Process'.

14) Wstawienie powiązania między klasami

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows two UML class diagrams. The top diagram is for the class **Tytul_książki**, which has the following attributes and operations:

```
classDiagram
    class Tytul_książki {
        private String wydawnictwo
        private String ISBN
        private String tytul
        private String nazwisko
        private String imie
        Tytul_książki()
        String getWydawnictwo()
        void setWydawnictwo(String val)
        String getISBN()
        void setISBN(String val)
        String getTytul()
        void setTytul(String val)
        String getNazwisko()
        void setNazwisko(String val)
        String getImie()
        void setImie(String val)
        String toString()
    }
```

The bottom diagram is for the class **Uchwyty**, which has the following operations:

```
classDiagram
    class Uchwyty {
        Uchwyty()
        void dodaj_tytul(String a, String b, String c, String d, String e)
    }
```

A line connects the **Uchwyty** class to the **Tytul_książki** class, indicating a relationship. The left sidebar shows a project tree with a package named **Tytul_książki**. The right sidebar shows the **Palette** with various UML elements like Class, Interface, Package, etc. The bottom status bar shows the Windows taskbar with the Start button and several open applications.

15) Określenie licznosci związku – 1 do 1..0

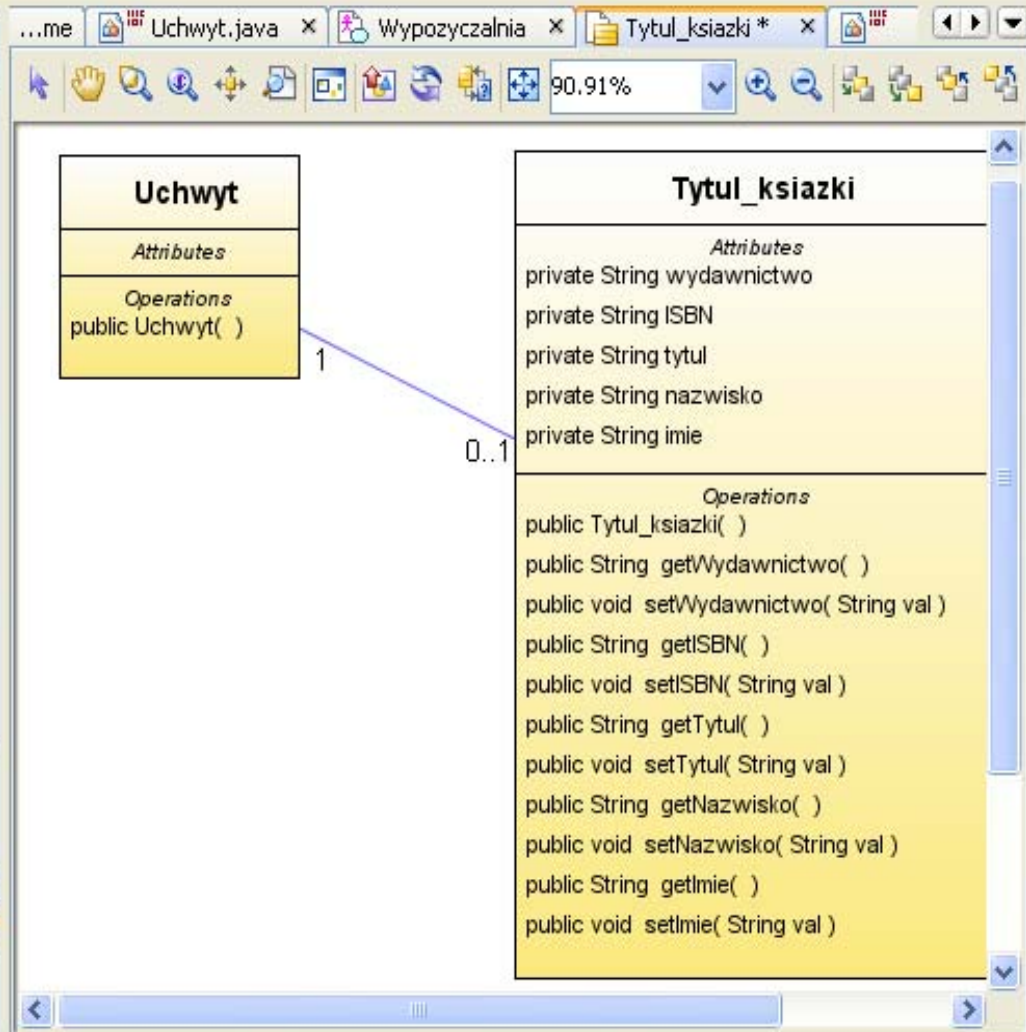
The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a UML class diagram with two classes: **Uchwy** and **Tytul_książki**. The **Uchwy** class has an attribute section and an operation `public Uchwy()`. The **Tytul_książki** class has an attribute section and three operations: `public void setWzawisko(String val)`, `public String getImie()`, and `public void setImie(String val)`. An association line connects the two classes. A context menu is open over the association, with the **Set Multiplicity** option selected. The submenu shows the multiplicity **1** highlighted. The **Palette** on the right shows the **Association** design pattern selected. The **UML Documentation** and **Association - Prop...** panels are also visible.

Save All finished.



Projects **Files** **Runtime**

- Katalog1
 - Source Packages
 - katalog1
 - Tytul_ksiazki.java
 - Tytul_ksiazki
 - Uchwyty.java
 - Uchwyty
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1
 - Model
 - Tytul_ksiazki
 - Wypożyczalnia
 - Tytul_ksiazki
 - Uchwyty
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul



Palette

- Derivation Edge
- Association**
 - Association
 - Composition
 - Navigable Composition
 - Aggregation
 - Navigable Aggregation
 - Navigable Association
- UML Documentation**
- Association - Prop...**
 - Association

| | |
|---------------|--------|
| Name | |
| Alias | |
| Visibility | public |
| Stereotypes | |
| Tanned Values | |
- Dynamic Help**
 - Finding Information At

Outline **Navigator**

<No View Available>

Output

- AplikacjaInternetowa1 (run) x
- Java DB Database Process x
- Sun Java System Application Server 9 x
- AplikacjaInternetowa1 (run) x
- AplikacjaInternetowa1 (run) x
- AplikacjaInternetowa1 (run) x
- Generate Code Log x

16) Wstawianie nowego diagramu – menu

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code of the `Tytul_książki.java` file, with the `public void setISBN(String val)` method highlighted. The left sidebar contains the 'Projects' and 'Files' views, where the 'UMLKatalog1' project is expanded to show the 'Model' package. A context menu is open over the 'Model' package, with the 'Add Diagram' option selected. The right sidebar shows the 'Diagrams - Properties' and 'Dynamic Help' panels. The bottom status bar indicates the cursor is at line 39, column 1, in the 'INS' state. The taskbar at the bottom shows several running applications, including 'AplikacjaInternetowa1 (run)', 'Java DB Database Process', 'Sun Java System Application Server 9', and 'AplikacjaInternetowa1 (run)'.

```
public void setISBN(String val) {
    this.ISBN = val;
}

public String getTytul() {
    return tytul;
}

public void setTytul(String val) {
    this.tytul = val;
}

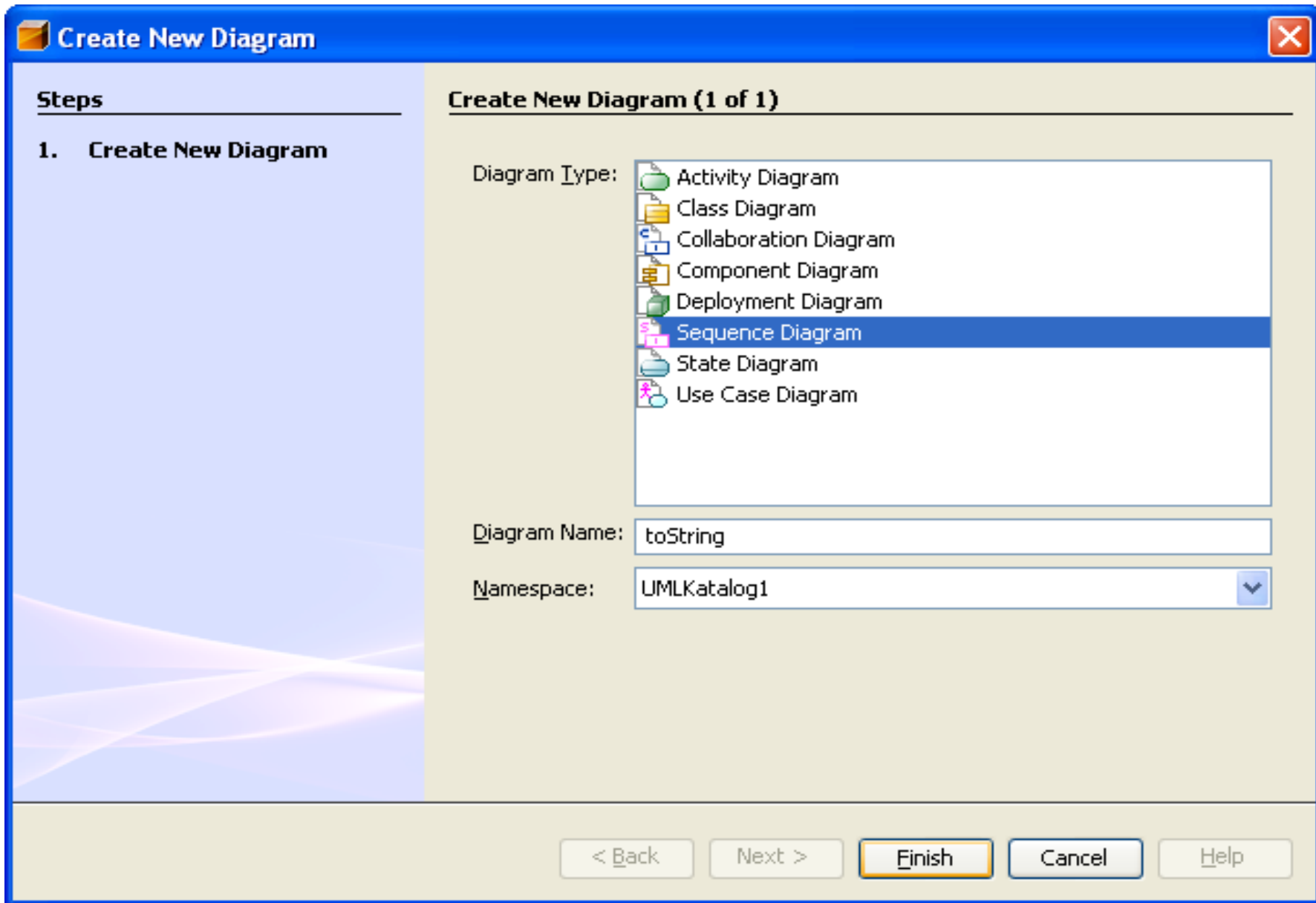
public String getNazwisko() {
    return nazwisko;
}

public void setNazwisko(String val) {
    this.nazwisko = val;
}

public String getImie() {
    return imie;
}

public void setImie(String val) {
    this.imie = val;
}
```

17) Wstawianie nowego diagramu typu sekwencji - wybór



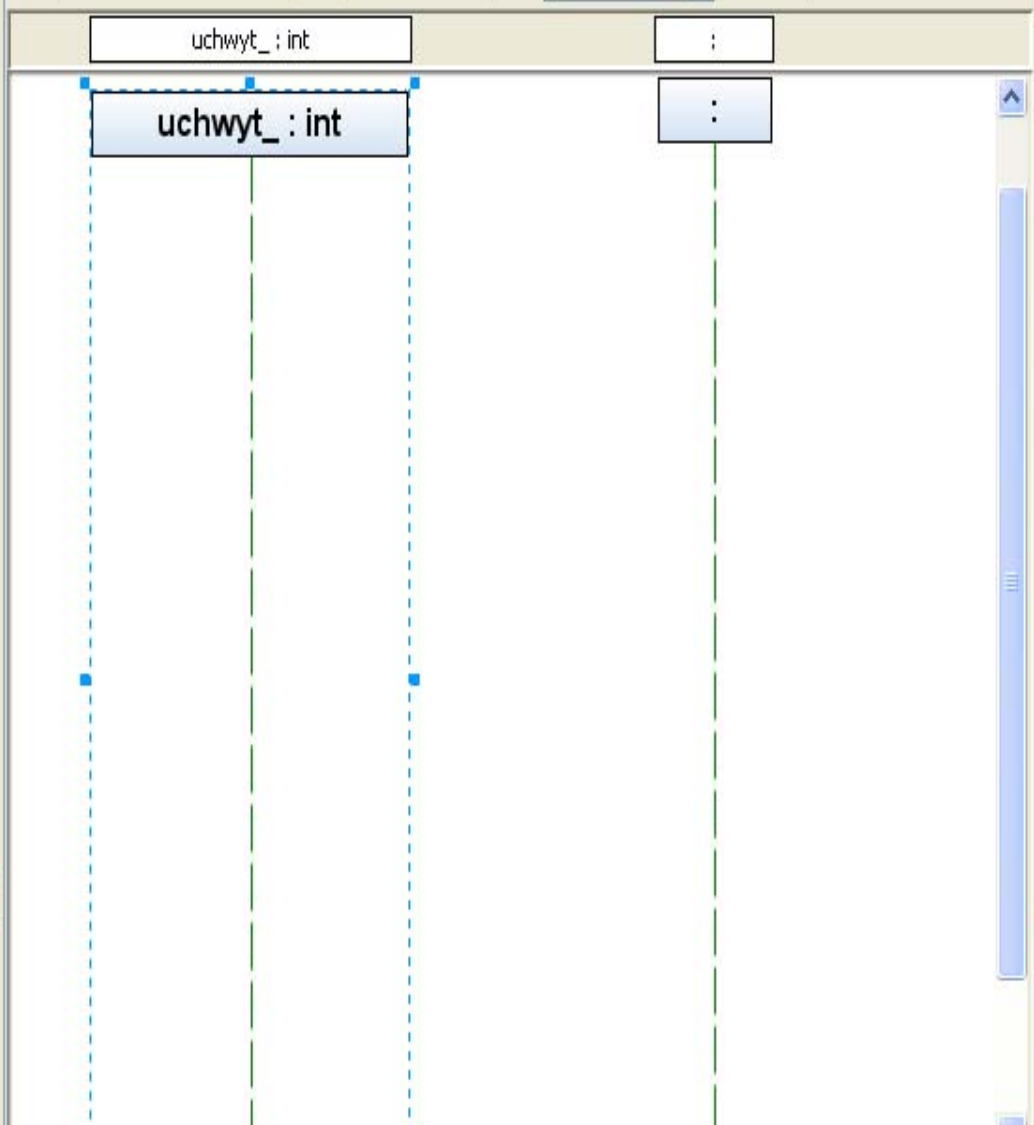
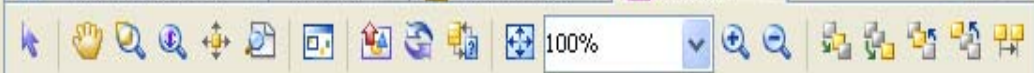


P... Files Run...

Katalog1

- UMLKatalog1
 - Model
 - Diagrams
 - Imported Elements

Tytul_ksiazki.java x Welcome x Tytul_ksiazki x toString * x



Palette

Basic

- Lifeline
- Actor
- Synchronous Message
- Asynchronous Message
- Create Message
- Message To Self

Control

- Destroy Lifeline
- Combined Fragment

Comments

- Comment
- Link Comment

Lifeline - Properties

Lifeline

| | |
|-------------------------|---------------|
| Name | uchwyt_ |
| Alias | uchwyt_ |
| Visibility | public |
| Stereotypes | |
| Tagged Values | |
| Documentation | |
| Constraints | |
| Representing Classifier | int |
| | int |
| | String |
| | void |
| Representing Classifier | Tytul_ksiazki |
| | Uchwyt |

Outline Navigator

<No View Available>

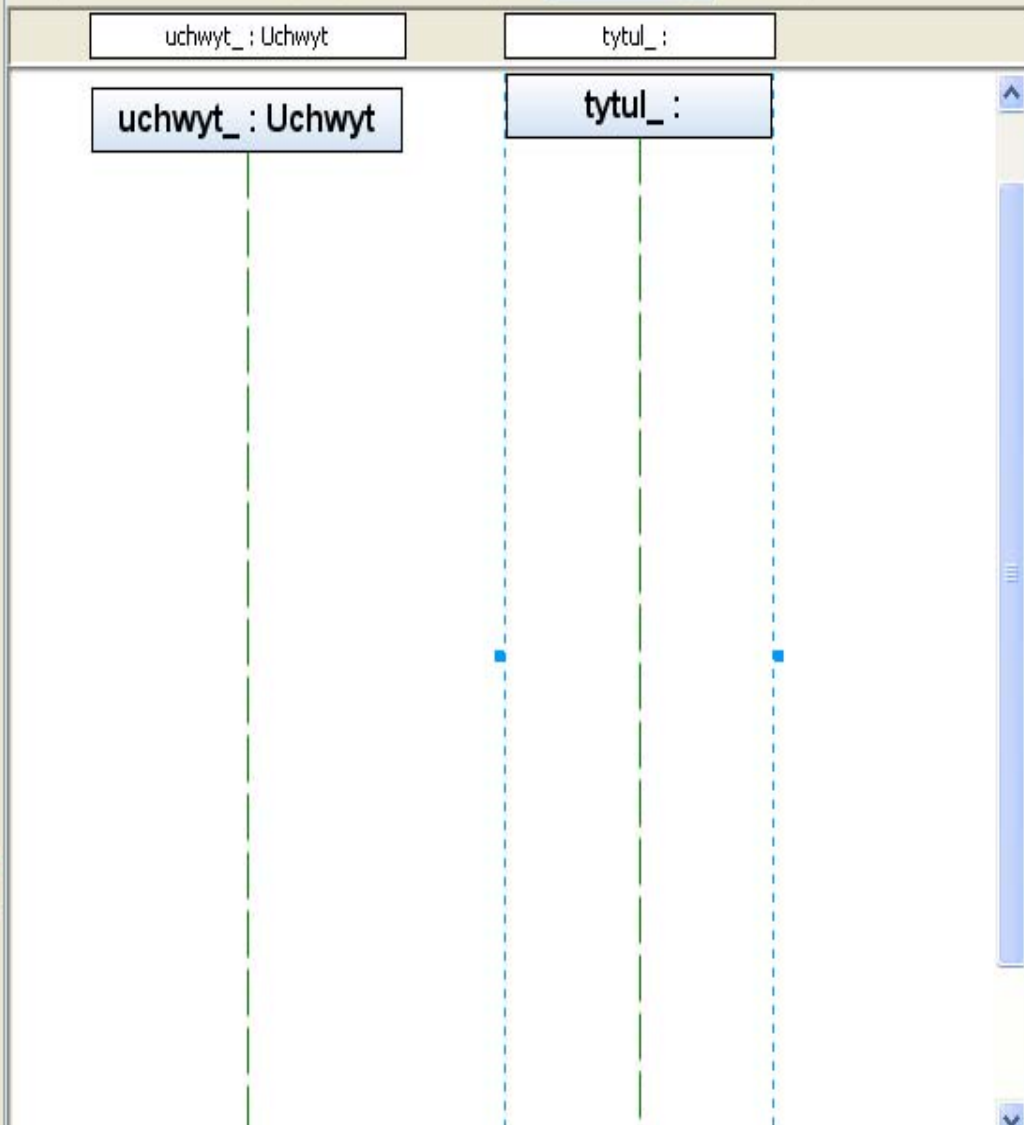
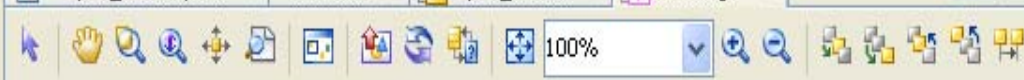


P... Files Run...

Katalog1

- UMLKatalog1
 - Model
 - Diagrams
 - Imported Elements

Tytul_ksiazki.java x Welcome x Tytul_ksiazki x toString * x



Palette

Basic

- Lifeline
- Actor
- Synchronous Message
- Asynchronous Message
- Create Message
- Message To Self

Control

- Destroy Lifeline
- Combined Fragment

Comments

- Comment
- Link Comment

tytul_ - Properties

| Lifeline | |
|--------------------------------|---------------|
| Name | tytul_ |
| Alias | tytul_ |
| Visibility | public |
| Stereotypes | |
| Tagged Values | |
| Documentation | |
| Constraints | |
| Representing Classifier | int |
| | int |
| | String |
| | void |
| Representing Classifier | Tytul_ksiazki |
| | Uchwyt |

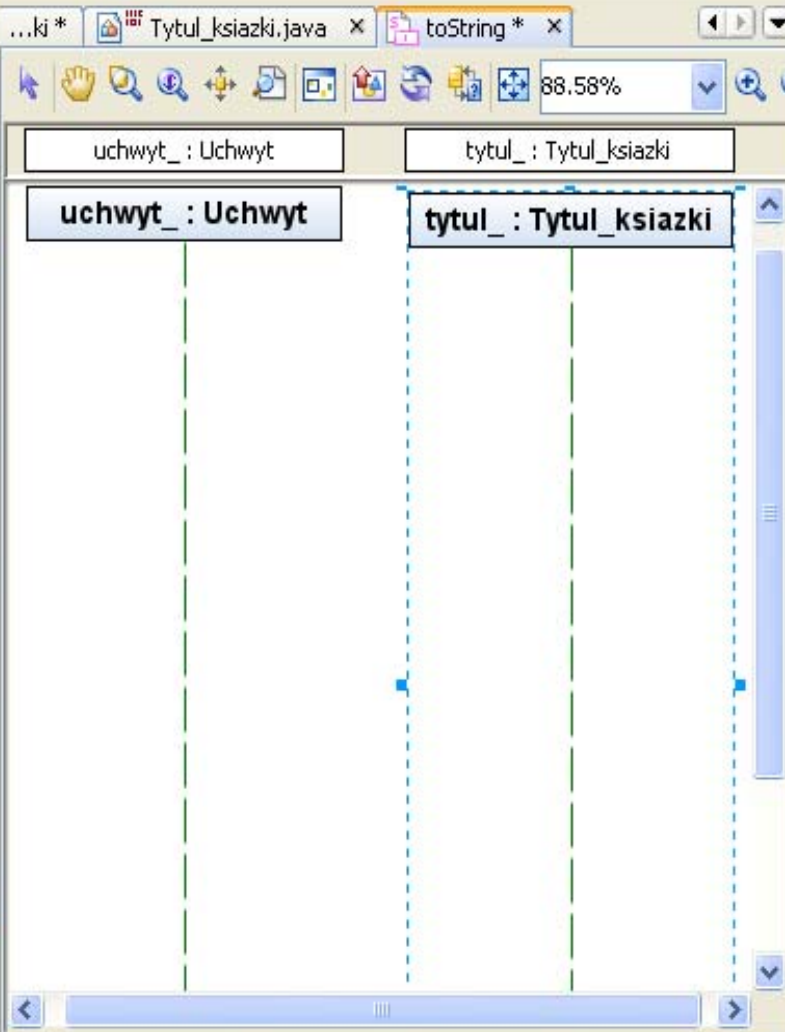
Outline Navigator

<No View Available>



Projects | **Files** | **Runtime**

- Uchwyty
- Test Packages
- Libraries
- Test Libraries
- UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - toString
 - Tytul_książki
 - Uchwyty
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul
 - Diagrams
 - toString
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements



Palette

- Basic
 - Lifeline
 - Actor
 - Synchronous Message
 - Asynchronous Message
 - Create Message

tytul_ - Properties

| | |
|---------------|--------|
| Lifeline | |
| Name | tytul_ |
| Alias | tytul_ |
| Visibility | public |
| Stereotypes | |
| Tagged Values | |
| Documentation | |

tytul_

Dynamic Help

- Finding Information About the IDE

Outline | **Navigator**

<No View Available>

Output

- AplikacjaInternetowa1 (run) ×
- Java DB Database Process ×
- Sun Java System Application Server 9 ×
- AplikacjaInternetowa1 (run) ×
- AplikacjaInternetowa1 (run) ×
- AplikacjaInternetowa1 (run) ×
- Generate Code Log ×

18) Wstawienie synchronicznej wiadomości („Synchronous Message”) – metody klasy Tytul_książki

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a UML sequence diagram with two lifelines: `uchwyt_ : Uchwyt` and `tytul_ : Tytul_książki`. A synchronous message arrow points from the `uchwyt_ : Uchwyt` lifeline to the `tytul_ : Tytul_książki` lifeline, labeled with the method signature `public void Unnamed()`. The diagram is zoomed in at 88.58%.

On the right side, the **Palette** panel shows the **Basic** category with options for Lifeline, Actor, Synchronous Message, Asynchronous Message, and Create Message. The **Message - Properties** panel shows the following details for the selected message:

| Message | |
|---------------|--------|
| Name | |
| Alias | |
| Visibility | public |
| Stereotypes | |
| Tagged Values | |
| Documentation | |

The **Dynamic Help** panel shows the text: "Finding Information About the IDE".

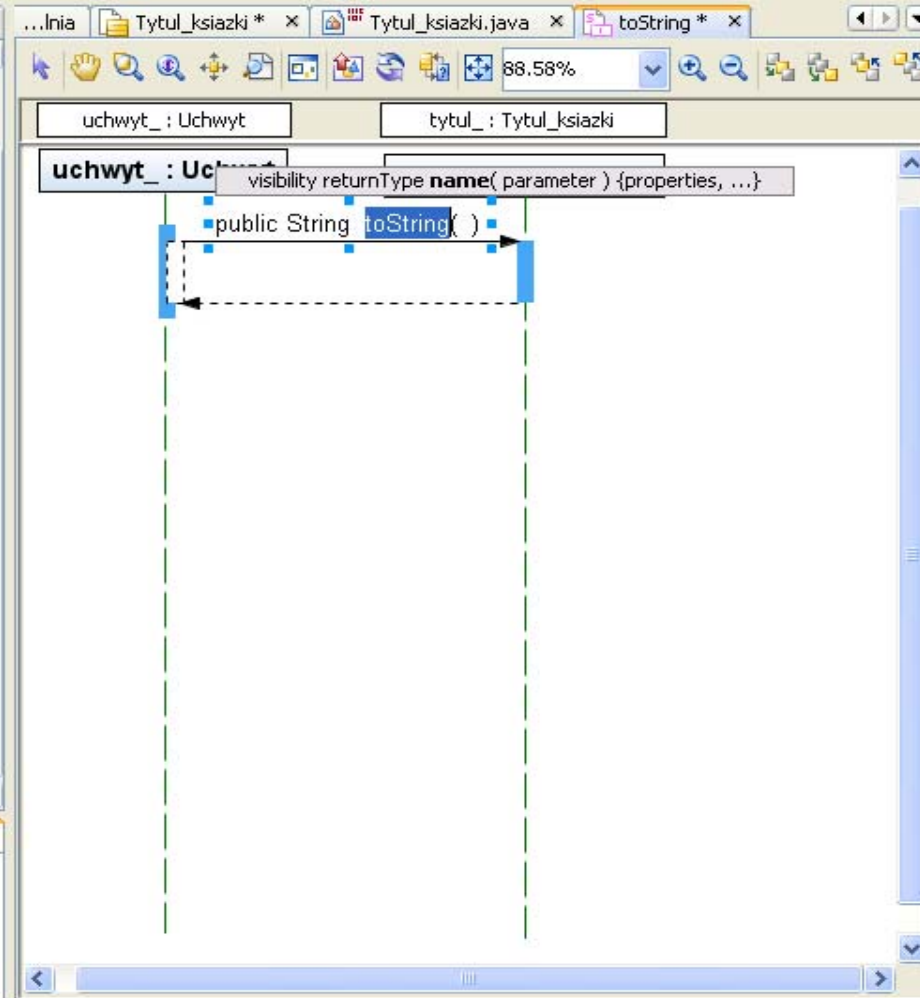
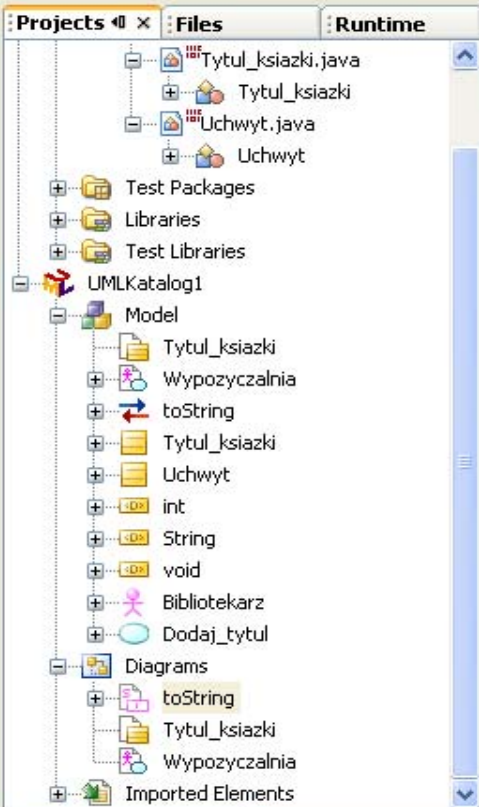
At the bottom, the **Output** window shows several running processes:

- AplikacjaInternetowa1 (run)
- Java DB Database Process
- Sun Java System Application Server 9
- AplikacjaInternetowa1 (run)
- AplikacjaInternetowa1 (run)
- AplikacjaInternetowa1 (run)
- Generate Code Log

19) Nadawanie nazwy `toString` nowej metodzie w klasie `Tytul_książki`

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help



Palette

- Basic
 - Lifeline
 - Actor
 - Synchronous Message
 - Asynchronous Message
 - Create Message
 - Message To Self
- public String toString() - UML Docu...**
- toString - Properties**

| | |
|---------------|------------------------|
| Name | toString |
| Alias | toString |
| Return Type | String |
| Parameters | public String toStr... |
| Visibility | public |
| Documentation | |
| Stereotypes | |
- Dynamic Help**
 - Finding Information About the IDE

Outline | **Navigator**

<No View Available>

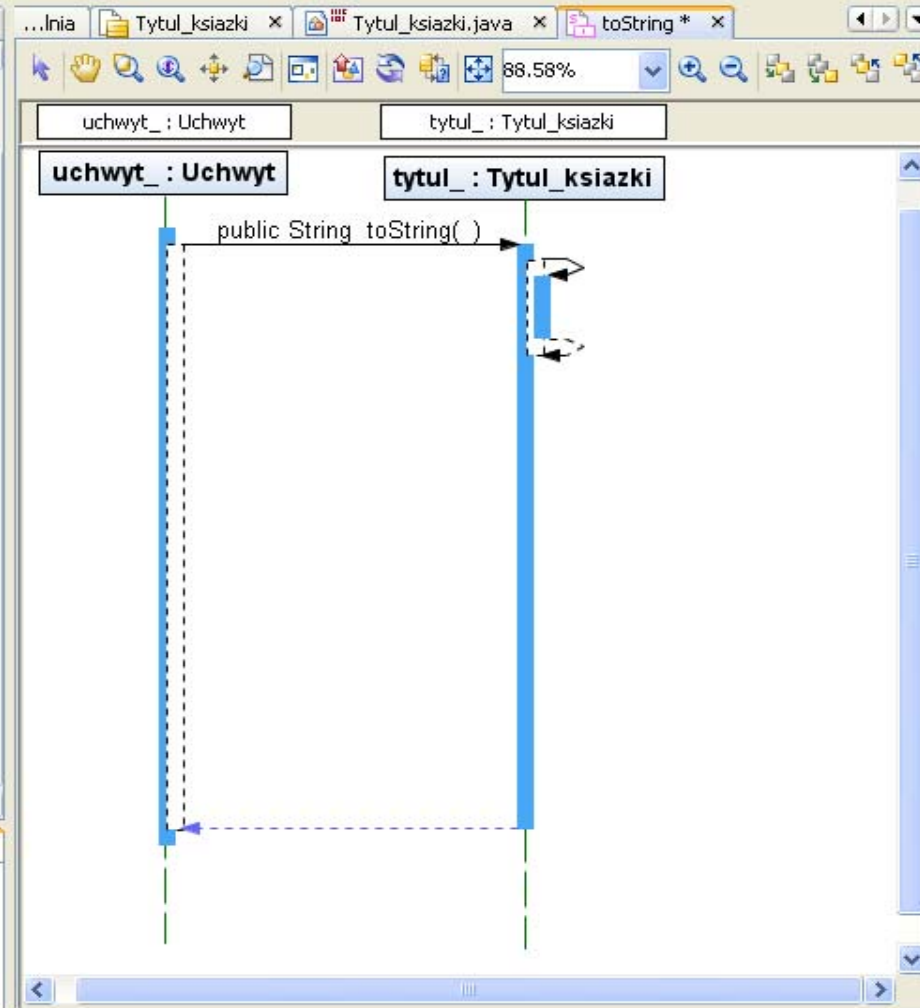
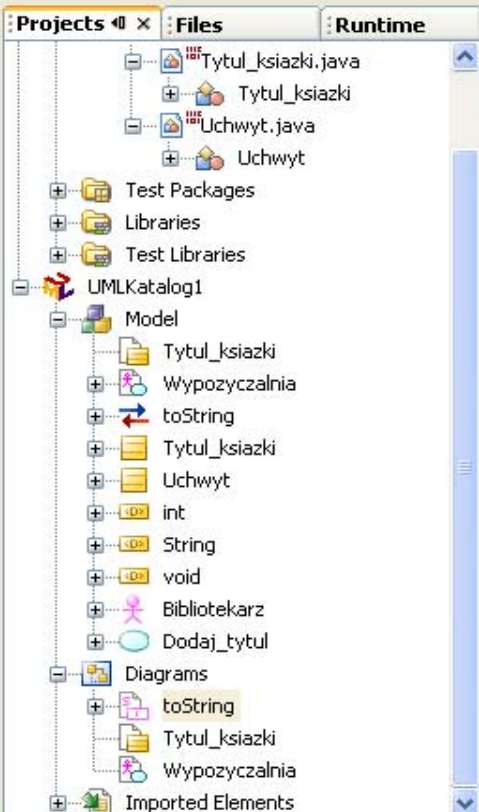
Output

| | | | |
|-----------------------------|-----------------------------|--------------------------------------|---|
| AplikacjaInternetowa1 (run) | Java DB Database Process | Sun Java System Application Server 9 | AplikacjaInternetowa1 (run) |
| AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) Generate Code Log |

20) Wstawienie wiadomości typu „Message to Self” – ciało metody `toString`

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help



Palette

- Basic
 - Lifeline
 - Actor
 - Synchronous Message
 - Asynchronous Message
 - Create Message
 - Message To Self
- Control

toString - Properties

| | |
|---------------|------------------------|
| Name | toString |
| Alias | toString |
| Return Type | String |
| Parameters | public String toStr... |
| Visibility | public |
| Documentation | |
| Stereotypes | |

toString

Dynamic Help

- Finding Information About the IDE

Outline Navigator

<No View Available>

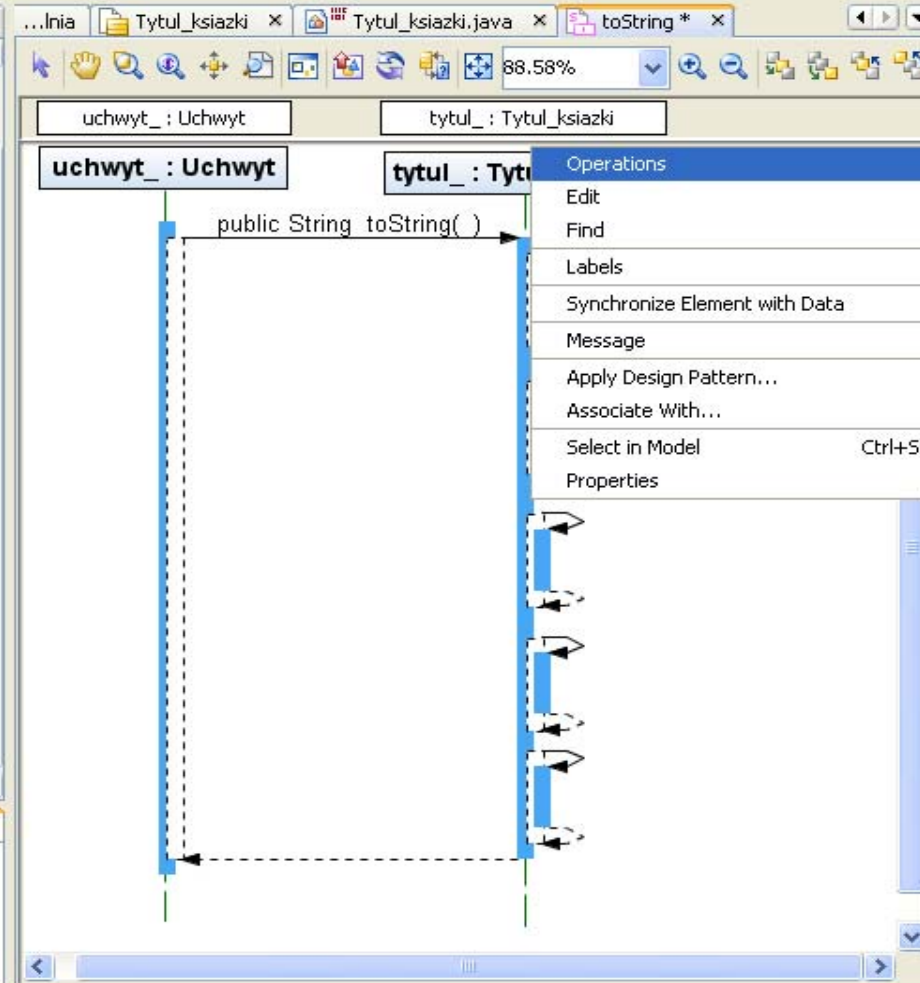
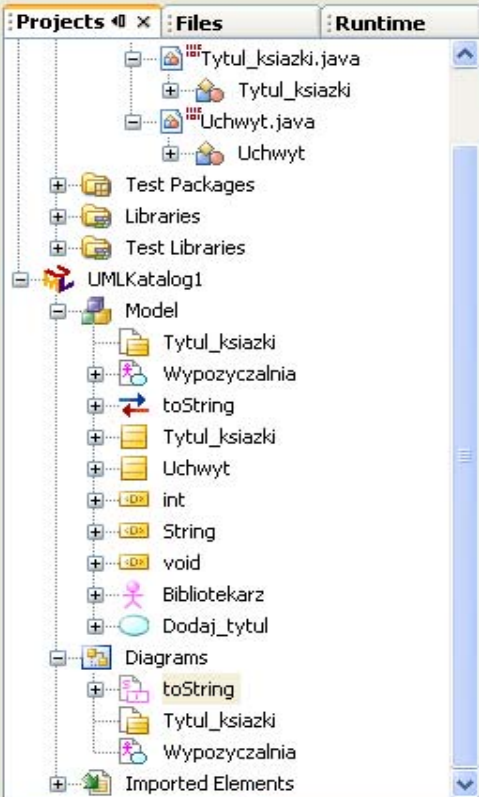
Output

| | | | |
|-----------------------------|-----------------------------|--------------------------------------|---|
| AplikacjaInternetowa1 (run) | Java DB Database Process | Sun Java System Application Server 9 | AplikacjaInternetowa1 (run) |
| AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) Generate Code Log |

21) Kojarzenie wiadomości z konkretną metodą klasy **Tytuł_książki**

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help



Operations

- Add Operation
- Protected
- Private
- Package
- public String getISBN()
- public String getImie()
- public String getNazwisko()
- public String getTytuł()
- public String getWydawnictwo()
- public void setISBN(String val)
- public void setImie(String val)
- public void setNazwisko(String val)
- public void setTytuł(String val)
- public void setWydawnictwo(String val)
- public String toString()

Message

Dynamic Help

Finding Information About the IDE

Output

| | | | |
|-----------------------------|-----------------------------|--------------------------------------|-----------------------------|
| AplikacjaInternetowa1 (run) | Java DB Database Process | Sun Java System Application Server 9 | AplikacjaInternetowa1 (run) |
| AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) | AplikacjaInternetowa1 (run) |

Generate Code Log



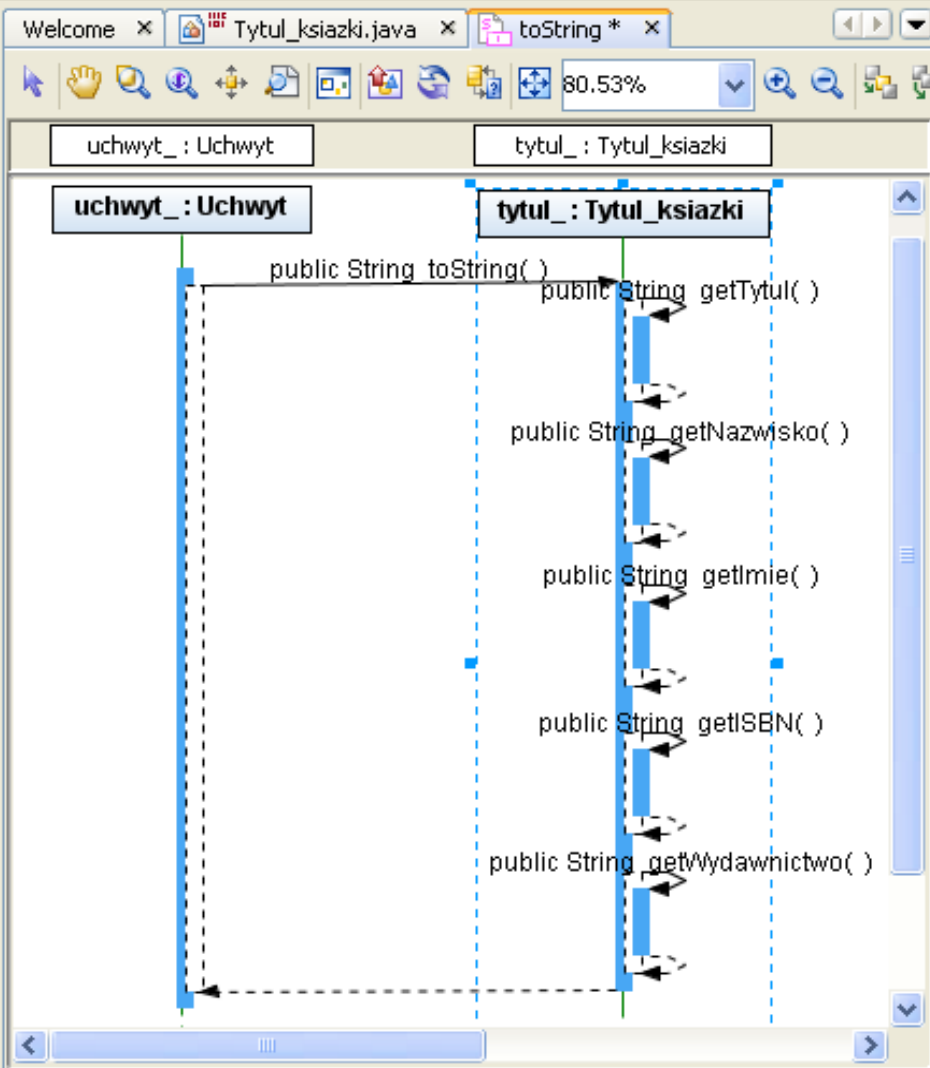
Proj... **Files** **Runtime**

- Libraries
- Test Libraries
- UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - toString
 - dodaj_tytul
 - toString
 - katalog1
 - Tytul_książki
 - Uchwył
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul
- Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Wypożyczalnia
- Imported Elements

Outline **Navigator**

<No View Available>

Output



Palette

- Basic**
 - Lifeline
 - Actor
 - Synchronous Message
 - Asynchronous Message
 - Create Message
 - Message To Self
- Control**
 - Destroy Lifeline
 - Combined Fragment
- Comments**

toString - Properties

| | |
|---------------|--------------------|
| Diagram | |
| Name | toString |
| Alias | toString |
| File Name | C:\Documents an... |
| Layout Style | sequence |
| Diagram Kind | Sequence Diagram |
| Documentation | |

toString

22) Pojawienie się nowej metody `toString` w klasie `Tytul_książki` – z diagramu sekwencji

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a UML class diagram with two classes: `Uchwył` and `Tytul_książki`. The `Uchwył` class has an operation `public Uchwył()`. The `Tytul_książki` class has attributes `private String wydawnictwo`, `private String ISBN`, `private String tytuł`, `private String nazwisko`, and `private String imie`. Its operations include `public Tytul_książki()`, `public String getWydawnictwo()`, `public void setWydawnictwo(String val)`, `public String getISBN()`, `public void setISBN(String val)`, `public String getTytuł()`, `public void setTytuł(String val)`, `public String getNazwisko()`, `public void setNazwisko(String val)`, `public String getImie()`, `public void setImie(String val)`, and `public String toString()`. A dependency arrow points from `Uchwył` to `Tytul_książki` with a multiplicity of `1` at the `Uchwył` end and `0..1` at the `Tytul_książki` end.

The right-hand side of the IDE features several panels: **Palette** (Basic, Interface, Package, Collaboration Lifeline, Enumeration, Node, Datatype), **Tytul_książki - UML Documenta...**, **Tytul_książki - Properties** (Class, Name: Tytul_książki, Alias: Tytul_książki, Visibility: public), and **Dynamic Help** (Finding Information About the IDE).

Save All finished.

23) Wygenerowanie nowej metody `toString` do klasy kodu klasy `Tytul_książki`

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the `Tytul_książki.java` file with the following code:

```
public void setTytul(String val) {
    this.tytul = val;
}

public String getNazwisko() {
    return nazwisko;
}

public void setNazwisko(String val) {
    this.nazwisko = val;
}

public String getImie() {
    return imie;
}

public void setImie(String val) {
    this.imie = val;
}

public String toString() {
    return null;
}
```

The `toString` method is currently returning `null`. The status bar at the bottom indicates the cursor is at line 62, column 1, with the text `INS`.

The left sidebar shows the project structure for `Katalog1`, including source packages, test packages, and libraries. The `UMLKatalog1` model is also visible, showing the `Tytul_książki` class and its relationships.

The right sidebar contains the `Tytul_książki - Properties` window, showing the class name, modifiers, and extends information. Below it is the `Dynamic Help` window, which provides information about the source editor.


```

public class Tytul_ksiazki
{
    private String wydawnictwo;
    private String ISBN;
    private String tytul;
    private String nazwisko;
    private String imie;

    public Tytul_ksiazki()                {} /** @ generated */

    public String getWydawnictwo()        { return wydawnictwo; } /** @ generated */

    public void setWydawnictwo(String val) { wydawnictwo = val; }

    public String getISBN()               { return ISBN; } /** @ generated */

    public void setISBN(String val)       { ISBN = val; } /** @ generated */

    public String getTytul()              { return tytul; } /** @ generated */

    public void setTytul(String val)      { tytul = val; } /** @ generated */

    public String getNazwisko()           { return nazwisko; } /** @ generated */

    public void setNazwisko(String val)   { nazwisko = val; } /** @ generated */

    public String getImie()               { return imie; } /** @ generated */

    public void setImie(String val)       { imie = val; } /** @ generated */

    public String toString() // your code here
    {
        String pom="Tytul: "+getTytul();
        pom+=" Autor:"+getNazwisko() +" "+getImie();
        pom+=" ISBN: "+getISBN();
        pom+=" Wydawnictwo:"+getWydawnictwo();
        return pom;
    }
}

```

24) Wstawienie ręczne kodu nowej metody `toString` do klasy `Tytul_książki`

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code for the `Tytul_książki` class in `Tytul_książki.java`. The `toString` method is being manually added and is highlighted in yellow. The code is as follows:

```
public String toString() {  
    String pom="Tytuł: "+getTytuł();  
    pom+=" Autor:"+getNazwisko() +" "+getImię();  
    pom+=" ISBN: "+getISBN();  
    pom+=" Wydawnictwo:"+getWydawnictwo();  
    return pom;  
}
```

The `Properties` window on the right shows the details for the `toString` method:

| :toString - Properties | |
|------------------------|-------------------|
| Modifiers | public |
| Name | toString |
| Type Parameters | |
| Parameters | |
| Return Type | String |
| Exceptions | |
| Javadoc Comment | <No comment pr... |

The `Dynamic Help` window shows links for "About the Source Editor" and "See Also". The `Outline` window at the bottom left lists the class members: `setTytuł(String val)`, `setWydawnictwo(String val)`, and `toString()`.

25) Wstawienie diagramu sekwencji dla przypadku użycia `dodaj_tytul`: użycie wiadomości typu „Asynchronous Message” oraz „Create Message” – konstruktor klasy `Tytul_książki`

NetBeans IDE 5.5

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Welcome x Tytul_książki.java x toString * x dodaj_tytul * x

90.91%

uchwyt_ : Uchwyt tytul_ : Tytul_książki

public void dodaj_tytul(String a, String b, String c, String d, String e)

public Tytul_książki()

public void setTytul(String a)

public void setNazwisko(String b)

public void setImie(String c)

public void setISBN(String d)

public void setWydawnictwo(String e)

Palette

- Basic
 - Lifeline
 - Actor
 - Synchronous Mess...
 - Asynchronous Mes...
 - Create Message
 - Message To Self
- Control
 - Destroy Lifeline
 - Combined Fragmer
- Comments
 - Comment

tytul_ - Proper...

| | |
|---------------|----------|
| Name | tytul_ |
| Alias | tytul_ |
| Visibility | public |
| Stereotypes | |
| Tagged Values | |
| Documentation | |
| Constraints | |
| Representing | Tytul... |

tytul_

Output

<No View ...

Save All Finished.

Start Windows Commander... Microsoft PowerPoint... NetBeans IDE 5.5 PL 07:35

26) Wygenerowanie nowej metody `dodaj_tytul` w klasie fasadowej `Uchwyt` z diagramu sekwencji

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a UML class diagram with two classes: `Uchwyt` and `Tytul_książki`. The `Uchwyt` class is highlighted with a dashed border, indicating it is the active class. The `Tytul_książki` class is shown to its right. A line with multiplicity '1' at the `Uchwyt` end and '0..1' at the `Tytul_książki` end connects the two classes, representing an association.

Uchwyt Class:

- Attributes:** (None listed)
- Operations:**
 - `public Uchwyt()`
 - `public void dodaj_tytul(String a, String b, String c, String d, String e)`

Tytul_książki Class:

- Attributes:**
 - `private String wydawnictwo`
 - `private String ISBN`
 - `private String tytul`
 - `private String nazwisko`
 - `private String imie`
- Operations:**
 - `public Tytul_książki()`
 - `public String getWydawnictwo()`
 - `public void setWydawnictwo(String val)`
 - `public String getISBN()`
 - `public void setISBN(String val)`
 - `public String getTytul()`
 - `public void setTytul(String val)`
 - `public String getNazwisko()`
 - `public void setNazwisko(String val)`
 - `public String getImie()`
 - `public void setImie(String val)`
 - `public String toString()`

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Build, Run, CVS, Tools, Window, Help), a toolbar with various icons, and a project explorer on the left showing a project structure with folders like 'Test F', 'Librari', 'Test L', 'UMLKatalc', and 'Model'. The 'Outline' window at the bottom left shows '<No View Available>'. The 'Output' window at the bottom is also empty.

27) Wygenerowanie kodu nowej metody `dodaj_tytul` w klasie `Uchwyt`

The screenshot shows the NetBeans IDE 5.5 interface. The main editor window displays the source code of the `Uchwyt` class in the `Uchwyt.java` file. The code is as follows:

```
/*
 * Uchwyt.java
 *
 * Created on 2 marzec 2007, 22:44
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package katalog1;

public class Uchwyt {
    public Uchwyt () {
    }

    public void dodaj_tytul(String a, String b, String c, String d, String e) {
    }
}
```

The `dodaj_tytul` method is highlighted in yellow. The left sidebar shows the project structure for `UMLKatalog1`, including source packages, test packages, and libraries. The `Uchwyt` class is visible under the `katalog1` source package. The `Outline` and `Navigator` panes are empty, showing `<No View Available>`. The `Output` pane is also empty. The status bar at the bottom indicates the cursor is at line 19, column 1, in `INS` mode.

28) Wstawienie ręczne kodu nowej metody **dodaj_tytul** oraz metody **main** testującej działanie wykonanego przypadku użycia

The screenshot shows the NetBeans IDE 5.5 interface. The main editor window displays the source code for the `Uchwyty` class in the `katalog1` package. The code includes a constructor, a `dodaj_tytul` method, and a `main` method. The `main` method is highlighted in yellow, showing an instance of `Uchwyty` being created and the `dodaj_tytul` method being called with five "1" arguments. The `toString` method of the `tytul_` object is also called and its output is printed to the console.

```
32:28 INS
package katalog1;

public class Uchwyty {
    public Tytul_książki tytul_;

    public Uchwyty () {
    }

    public void dodaj_tytul (String a, String b, String c, String d, String e)
    {
        // your code here
        tytul_ = new Tytul_książki ();
        tytul_.setTytul (a);
        tytul_.setNazwisko (b);
        tytul_.setImie (c);
        tytul_.setISBN (d);
        tytul_.setWydawnictwo (e);
    }

    public static void main (String t[]) // your code here
    {
        Uchwyty ap = new Uchwyty ();
        ap.dodaj_tytul ("1", "1", "1", "1", "1");
        String lan = ap.tytul_.toString ();
        System.out.println (lan);
    }
}
```

The interface also shows a project tree on the left with the following structure:

- Katalog1
 - Source Packages
 - katalog1
 - Tytul_książki.java
 - Tytul_książki
 - Uchwyty.java
 - Uchwyty
 - Test Packages
 - Libraries
 - Test Libraries
 - UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - dodaj_tytul
 - toString
 - tytul_
 - uchwyty_
 - Messages

The Output window at the bottom shows the following text:

```
Java DB Database Process Sun Java System Application Server 9 Generate Code Log Katalog1 (jar)
```

```
public class Uchwyt {  
public Tytul_ksiazki tytul_  
public void dodaj_tytul(String a, String b, String c, String d, String e)  
  { // your code here  
    tytul_=new Tytul_ksiazki();  
    tytul_.setTytul(a);  
    tytul_.setNazwisko(b);  
    tytul_.setImie(c);  
    tytul_.setISBN(d);  
    tytul_.setWydawnictwo(e);  
  }  
}
```

```
public static void main(String t[]) // your code here  
  { Uchwyt ap=new Uchwyt();  
    ap.dodaj_tytul("1","1","1","1","1");  
    String lan=ap.tytul_.toString();  
    System.out.println(lan);  
  }  
}
```

29) Uruchomienie programu

