

Wykład 2 – część druga

Iteracyjno-rozwojowy cykl oprogramowania 2

Przygotowanie powiązania 1:1

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows two class diagrams: **Tytul_ksiazki** and **Uchwyt**. The **Tytul_ksiazki** class has the following attributes and operations:

```
Attributes
private String wydawnictwo
private String ISBN
private String tytul
private String nazwisko
private String imie

Operations
public Tytul_ksiazki( )
public String getWydawnictwo( )
public void setWydawnictwo( String e )
public String getTytul( )
public void setISBN( String d )
public String getISBN( )
public void setTytul( String a )
```

The **Uchwyt** class has the following attributes and operations:

```
Attributes

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

An association line connects the two classes. The multiplicity at the **Tytul_ksiazki** end is **0..1**, and the multiplicity at the **Uchwyt** end is **1**. A context menu is open over the **0..1** multiplicity, with the **Set Multiplicity** option selected. The sub-menu shows the following options:

- 0..1
- 0..*
- *
- 1** (selected)
- 1..*

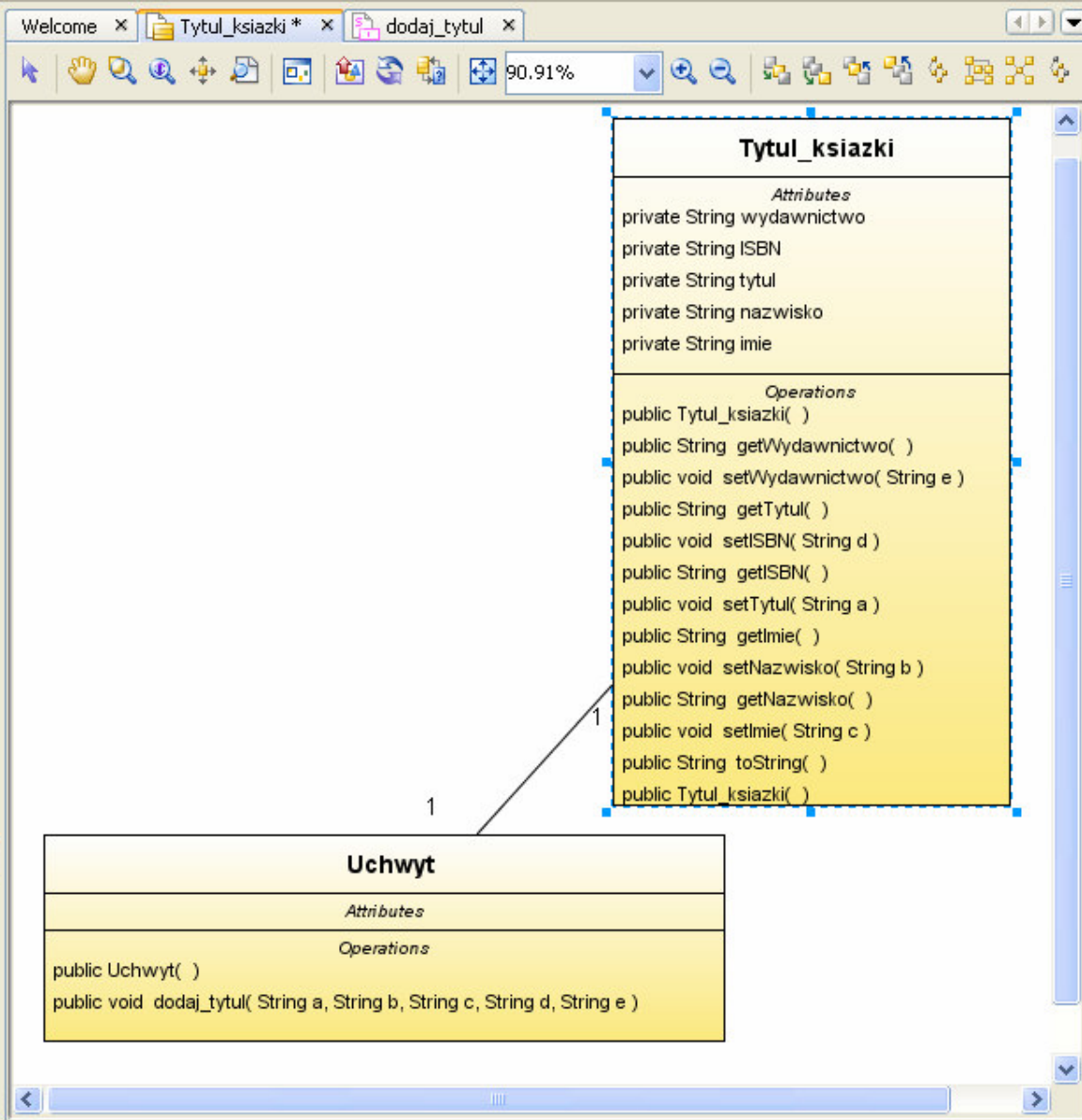
The **Association - Prop...** panel on the right shows the following properties:

Association	
Name	
Alias	
Visibility	public
Stereotypes	
Tagged Values	
Documentation	gfdgdfgfdgdf
Constraints	
Final	<input type="checkbox"/>

The **Dynamic Help** panel at the bottom right shows the text: "Finding Information A".

Proj... | Files | Runtime

- AplikacjaInternetowa5
- Katalog2
 - Source Packages
 - katalog2
 - Tytul_książki.java
 - Uchwyt.java
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Uchwyt
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul
 - Diagrams
 - Imported Elements



Palette

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

Tytul_książki - ...

Tytul_książki - Pr...

Class	
Name	Tytul_książki...
Alias	Tytul_książki...
Visibility	public
Stereotypes	
Tagged Values	
Documentation	
Constraints	
Final	<input type="checkbox"/>
Template Parameters	
Transient	<input checked="" type="checkbox"/>
Abstract	<input type="checkbox"/>

Tytul_książki

Dynamic Help

- Finding Information A...

Navigator | Outline

<No View Available>

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

- AplikacjaInternetowa5
- Katalog2
 - Source Packages
 - katalog2
 - Tytul_książki.java
 - Uchwyty.java
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Uchwyty
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul
 - Diagrams
 - Imported Elements

Welcome x Tytul_książki * x dodaj_tytul * x

90.91%

Tytul_książki

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 e)
- public String getTytul()
- public void setISBN(String d)
- public String getISBN()
- public void setTytul(String a)
- public String getImie()

Uchwyty

Attributes

Operations

- public Uchwyty()
- public void dodaj_tytul(String a, String b, String c, String d, ...)

1 1

- Edit
- Transform
 - To Ordinary Aggregate
 - To Composite Aggregate
 - Remove Aggregate
 - Navigable
 - Reverse Ends
- Find
- Labels
- Set Multiplicity
- Synchronize Element with Data
- Association
 - Show Qualifier
 - Generate Dependency Diagram
 - Apply Design Pattern...
 - Associate With...
- Select in Model Ctrl+Shift+5
- Properties

Palette

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

UML Documenta...

Association - Prop...

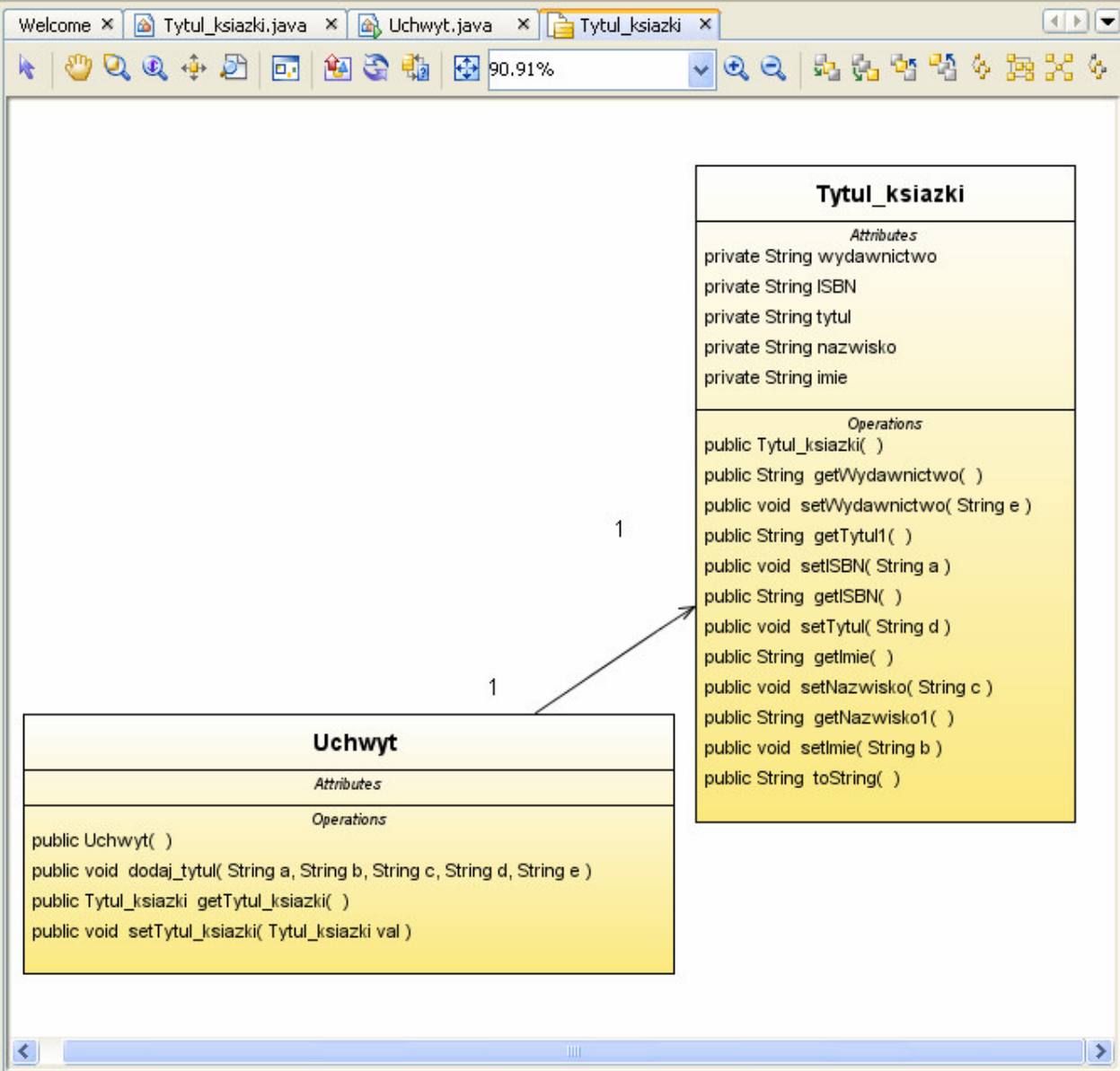
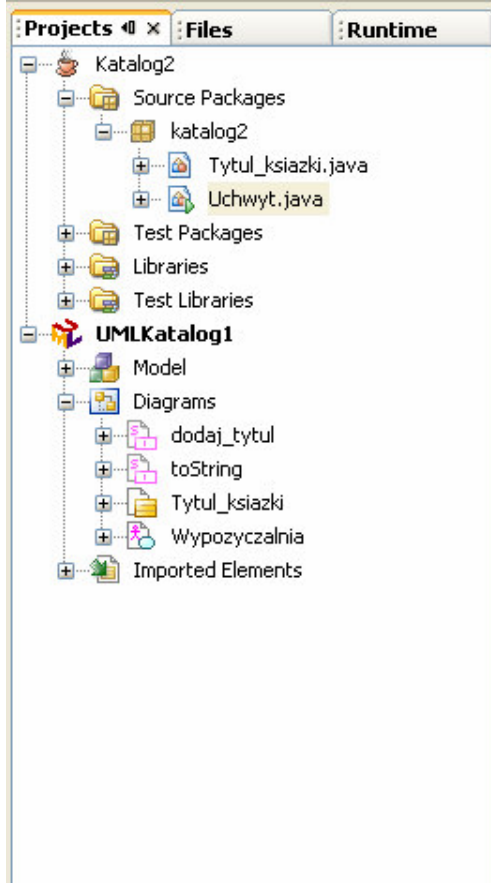
Association	
Name	
Alias	
Visibility	public
Stereotypes	
Tagged Values	
Documentation	gfdg...
Constraints	
Final	

Dynamic Help

- Finding Information A

Navigator **Outline**

<No View Available>



Palette

- Basic
 - Class
 - Interface
 - Package
 - Collaborator
 - Enumeral
 - Node
- Tytul...
 - b i u
- Tytul_...
 - Diagram
 - Name Ty..
 - Alias Ty..
 - File Name C:..
 - Layout Style ..
 - Diagram Kind Cl...
 - Documentat ...
- Tytul_ksi...
- Dyna...
 - Finding

Navigator Outline

<No View Available>

Generowanie kodu, który implementuje powiązanie 1:1

The screenshot shows the NetBeans IDE 5.5 interface. The main window displays two UML class diagrams: **Tytul_książki** and **Uchwyt**. A 1:1 association is shown between them. The **Tytul_książki** class has attributes: `private String wydawnictwo`, `private String ISBN`, `private String tytul`, `private String nazwisko`, and `private String imie`. Its operations include `public Tytul_książki()`, `public String getWydawnictwo()`, `public void setWydawnictwo(String e)`, `public String getTytuł1()`, `public void setISBN(String a)`, `public String getISBN()`, `public void setTytuł(String d)`, `public String getImie()`, `public void setNazwisko(String c)`, `public String getNazwisko1()`, `public void setImie(String b)`, and `public String toString()`. The **Uchwyt** class has operations: `public Uchwyt()`, `public void dodaj_tytuł(String a, String b, String c, String d, String e)`, `public Tytul_książki getTytuł_książki()`, and `public void setTytuł_książki(Tytul_książki val)`. A context menu is open over the **Uchwyt** class, with **Generate Code...** selected. The IDE also shows a project tree on the left, a palette on the right, and a status bar at the bottom indicating 'Finished building Katalog2 (run)'.

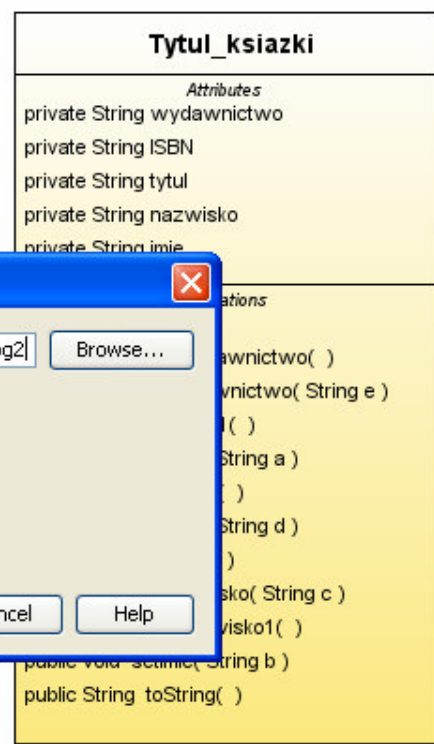
```
classDiagram
    class Tytul_książki {
        private String wydawnictwo
        private String ISBN
        private String tytul
        private String nazwisko
        private String imie
        Tytul_książki()
        String getWydawnictwo()
        void setWydawnictwo(String e)
        String getTytuł1()
        void setISBN(String a)
        String getISBN()
        void setTytuł(String d)
        String getImie()
        void setNazwisko(String c)
        String getNazwisko1()
        void setImie(String b)
        String toString()
    }
    class Uchwyt {
        Uchwyt()
        void dodaj_tytuł(String a, String b, String c, String d, String e)
        Tytul_książki getTytuł_książki()
        void setTytuł_książki(Tytul_książki val)
    }
    Tytul_książki "1" -- "1" Uchwyt
```

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

- Katalog2
 - Source Packages
 - katalog2
 - Tytul_ksiazki.java
 - Uchwyt.java
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1**
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_ksiazki
 - Wypożyczalnia
 - Imported Elements

Welcome | Tytul_ksiazki.java | Uchwyt.java | Tytul_ksiazki

90.91%



Generate Code

Folder Location:

Backup Existing Source Files

Navigator | **Outline**

<No View Available>

Palette

- Basic
 - Class
 - Interface
 - Package
 - Collaborator
 - Enumeral
 - Node
- UML D...
- UMLKa...

UMLKatal...
UMLKatalog1

Dyna...
Projects
See Also

Output | HTTP Monitor

Finished building Katalog2 (run).

Pr... x Files Runtime Welcome x Tytul_książki x

Katalog2
UMLKatalog1
Model
Tytul_książki

90.91%

Basic
Class
Interface

Output - Generate Code Log

```
Begin processing Generate Code: 2 items |
```

```
Code Generation selected options:
```

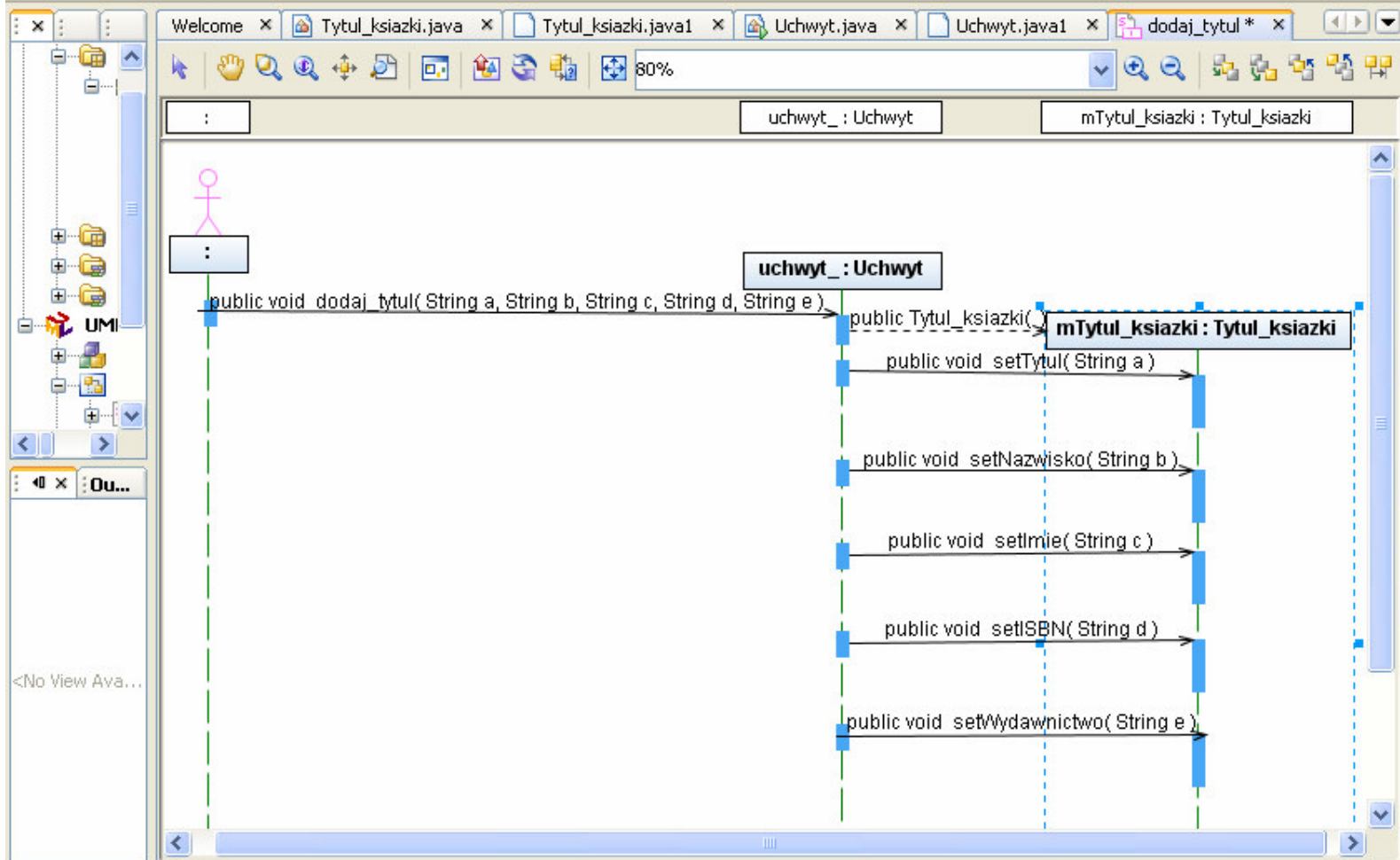
```
[ Source Folder Location C:\Documents and Settings\Kruczkiewicz\rozne\dydaktyka\PI0\wk2\P1\Katalog2\src\katalog2 ]
```

```
Processing element 1 of 2: Class Tytul_książki
```

```
Processing element 2 of 2: Class Uchwył
```

```
=====
```

```
Task Successful (total time: 9 seconds)
```

Palette

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

mTytul_książki : Tytul...

mTytul_książki - Prop...

Lifeline	
Name	mTytul_książki...
Alias	mTytul_książki...
Visibility	public
Stereotypes	...
Tagged Values	...
Documentation	...
Constraints	...
Representing Class	Tytul

Dynamic Help

- Finding Information About

Refactoring

Rename× Rename× Rename×

Rename katalog1 to katalog2 [2 occurrences] - NEEDS REFRESH

Do Refactoring Cancel

Output HTTP Monitor

Wygenerowany kod klasy Uchwyt

The screenshot displays the NetBeans IDE 5.5 interface. The main window shows the source code for the `Uchwyt` class in the `katalog2` package. The code includes a package declaration, an import for `java.util.ArrayList`, a class comment, and three methods: a constructor, `dodaj_tytul`, and `getTytul_książki`. The `setTytul_książki` method is also present but its body is empty in the visible code.

```
1  /**
2  * Uchwyt.java
3  *
4  * Created on 2 marzec 2007, 22:44
5  *
6  * To change this template, choose Tools | Template Manager
7  * and open the template in the editor.
8  */
9
10 package katalog2;
11
12 import java.util.ArrayList;
13
14 /**
15  * bmbnmbmb
16  */
17 public class Uchwyt {
18     private Tytul_książki mTytul_książki;
19
20     public Uchwyt() {
21     }
22
23     public void dodaj_tytul(String a, String b, String c, String d, String e) {
24     }
25
26     public Tytul_książki getTytul_książki() {
27         return mTytul_książki;
28     }
29
30     public void setTytul_książki(Tytul_książki val) {
31         this.mTytul_książki = val;
32     }
33 }
```

The interface also shows the project structure on the left, the Outline view at the bottom left, and the Windows taskbar at the bottom.

Uzupełnienie kodu klasy Uchwyt

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the code for the `Uchwyt` class in the `katalog2` package. The code includes a package declaration, an import for `java.util.ArrayList`, and the implementation of the `Uchwyt` class with a private attribute `mTytul_książki`, a constructor, and methods `dodaj_tytul`, `getTytul_książki`, and `setTytul_książki`. A `main` method is also present for testing.

```
7  * and open the template in the editor.
8  */
9
10 package katalog2;
11
12 import java.util.ArrayList;
13
14 public class Uchwyt {
15     private Tytul_książki mTytul_książki;
16
17     public Uchwyt() { }
18
19     public void dodaj_tytul(String a, String b, String c, String d, String e) {
20         mTytul_książki=new Tytul_książki();
21         mTytul_książki.setTytul(a);
22         mTytul_książki.setNazwisko(b);
23         mTytul_książki.setImie(c);
24         mTytul_książki.setISBN(d);
25         mTytul_książki.setWydawnictwo(e);
26     }
27     public Tytul_książki getTytul_książki() {
28         return mTytul_książki;
29     }
30     public void setTytul_książki( Tytul_książki val) {
31         this.mTytul_książki = val;
32     }
33     public static void main(String t[]) // your code here
34     { Uchwyt ap=new Uchwyt();
35       ap.dodaj_tytul("1","1","1","1","1");
36       String lan=ap.getTytul_książki().toString();
37       System.out.println(lan); }
38 }
39
```

The left sidebar shows the project structure for 'Katalog2' and 'UMLKatalog1'. The 'Uchwyt' class is highlighted in the UML diagram. The bottom status bar shows the cursor is at line 14, column 1.

```

package katalog1;

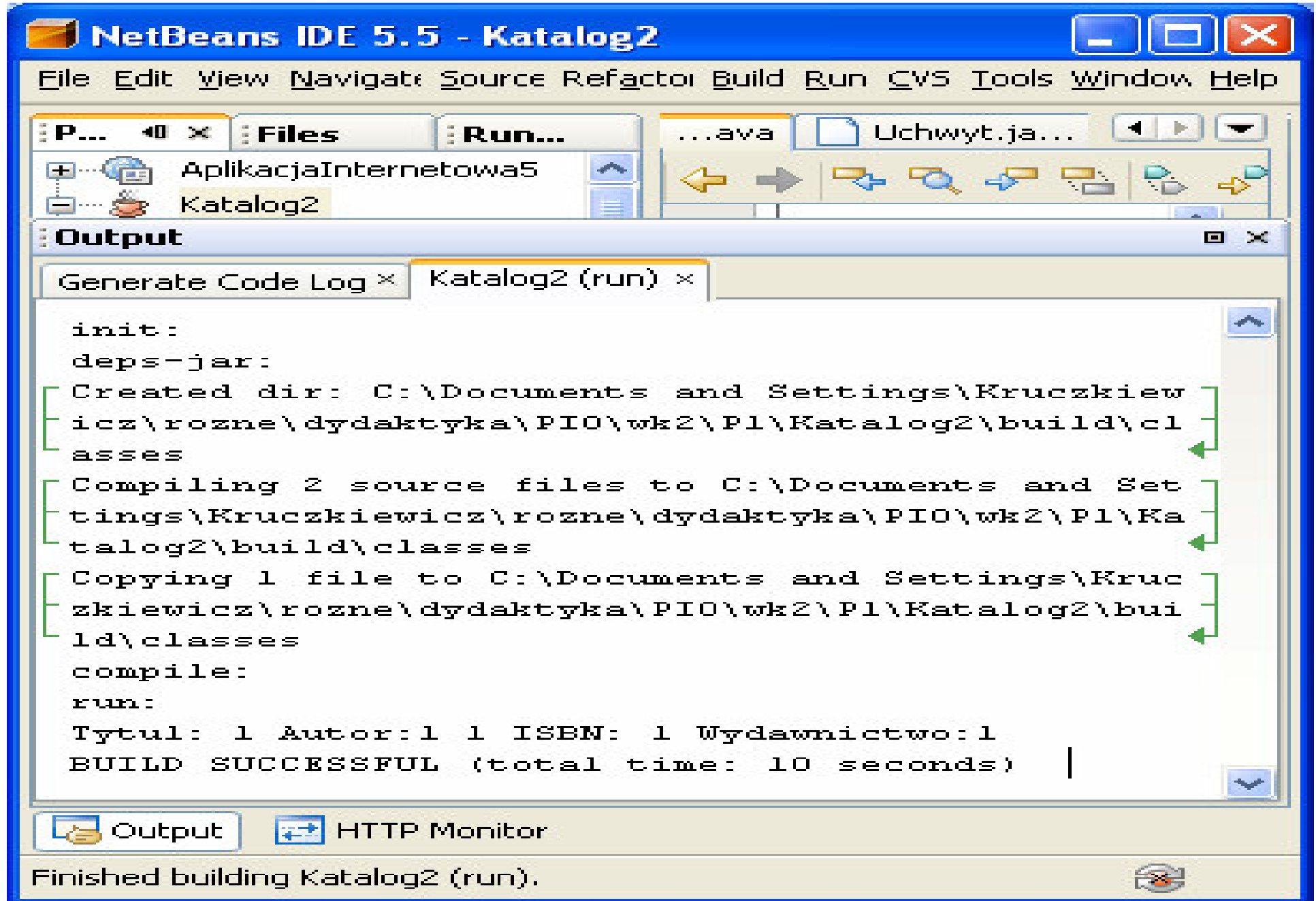
public class Uchwyt {
    private Tytul_ksiazki mTytul_ksiazki;
    public Uchwyt() { }
    public void dodaj_tytul(String a, String b, String c, String d, String e) // your code here
    { mTytul_ksiazki=new Tytul_ksiazki();
      mTytul_ksiazki.setTytul(a);
      mTytul_ksiazki.setNazwisko(b);
   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

    public Tytul_ksiazki getTytul_ksiazki()      { return mTytul_ksiazki;      }
    public void setTytul_ksiazki(Tytul_ksiazki val) { this.mTytul_ksiazki = val; }

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

```

Uruchomienie programu



NetBeans IDE 5.5 interface showing a UML class diagram illustrating a one-to-many relationship between **Uchwył** and **Tytul_książki**.

Uchwył Class:

```

public Uchwył( )
public void dodaj_tytul( String a, String b, String c, String d, String e )
public Tytul_książki[1..*] getTytul_książki( )
public void setTytul_książki( Tytul_książki val[1..*] )
public void addTytul_książki( Tytul_książki tytul_książki )

```

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 e )
public String getTytul( )
public void setISBN( String d )
public String getISBN( )
public void setTytul( String a )
public String getImie( )
public void setNazwisko( String b )
public String getNazwisko( )
public void setImie( String c )
public String toString( )

```

The relationship is shown as a directed association from **Uchwył** to **Tytul_książki** with a multiplicity of **1..*** at the **Tytul_książki** end.

Left sidebar (Project Explorer):

- Interfejs1 - Interfejs8
- JavaAp1
- Katalog3
- Komunikator1 - komunikator3
- Rys1 - Rys4
- UMLKatalog1
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements

Right sidebar (Palette):

- Basic
 - Class
 - Interface
 - Package
 - Collaboration Lifeline
- equals - Properties
 - Operation
 - Name: equals
 - Alias: equals
 - Return Type: bo...
 - Parameters: pu...
 - Visibility: p...
 - Documentation: ...
 - Stereotypes: ...
 - Tagged Value: ...
 - Abstract:
 - Final:
- Dynamic Help
 - Finding Informatic...

Bottom panels:

- Navigator: <No View Available>
- Usages: <No Usages>
- Output: HTTP Monitor

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

- Tytul_książki.java
- Uchwyt.java
- Test Packages
- Libraries
- Test Libraries
- UMLKatalog1**
 - Model
 - Tytul_książki
 - Wypożyczalnia
 - toString
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Uchwyt
 - int
 - String
 - void
 - Bibliotekarz
 - Dodaj_tytul
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements

Welcome | Tytul_książki.java | Uchwyt.java | Tytul_książki | 90.91%

Tytul_książki

Attributes

- private String wydawnictwo
- private String ISBN
- private String tytul
- private String nazwisko
- private String imie

Operations

- public void setWydawnictwo(String e)
- public String getWydawnictwo()
- public void setISBN(String e)
- public String getISBN()
- public void setTytul(String a)
- public String getTytul()
- public void setNazwisko(String d)
- public String getNazwisko()
- public void setImie(String b)
- public String toString()

Palette

- Basic
 - Class
 - Interface
 - Package
 - Collaborator
 - Enumeral
 - Node
- UML D...
- UMLKa...

UMLKatalog1

Dyna...

- Projects
- See Also

Generate Code

Folder Location:

Backup Existing Source Files

Navigator | **Outline**

<No View Available>

Wygenerowany kod implementujący relację 1:wiele

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code for the `Uchwyt.java` file, which implements a 1-to-many relationship. The code is as follows:

```
1  /**
2   * Uchwyt.java
3   *
4   * Created on 2 marzec 2007, 22:44
5   *
6   * To change this template, choose Tools | Template Manager
7   * and open the template in the editor.
8   */
9
10 package katalog1;
11
12 import java.util.ArrayList;
13
14 /**
15  * bmbnmbmb
16  */
17 public class Uchwyt {
18     private ArrayList<Tytul_książki> mTytul_książki;
19
20     public Uchwyt() {
21     }
22
23     public void dodaj_tytul(String a, String b, String c, String d, String e) {
24     }
25
26     public ArrayList<Tytul_książki> getTytul_książki() {
27         return mTytul_książki;
28     }
29
30     public void setTytul_książki(ArrayList<Tytul_książki> val) {
31         this.mTytul_książki = val;
32     }
33 }
34
```

The IDE interface includes a Project Explorer on the left showing the project structure, a Navigator window below it, and a Dynamic Help window on the right. The status bar at the bottom indicates the cursor is at line 1, column 1 in the `INS` mode.

Zmiana scenariusza dodaj_tytul - dodanie LifeLine dla kolekcji mTytul_ksiazki

The screenshot shows the NetBeans IDE 5.5 interface. The main window displays a UML sequence diagram for the `dodaj_tytul` method. The diagram features three lifelines: an actor (represented by a stick figure), `uchwyty : Uchwyty`, and `tytul : Tytul_ksiazki`. The actor sends a synchronous message `public void dodaj_tytul(String a, String b, String c, String d, String e)` to the `uchwyty` object. The `uchwyty` object then sends a series of messages to the `tytul` object to set its attributes: `public Tytul_ksiazki()`, `public void setTytul(String a)`, `public void setNazwisko(String b)`, `public void setmie(String c)`, `public void setISBN(String d)`, and `public void setWydawnictwo(String e)`. The `tytul` lifeline is shown as a dashed line, indicating it is a collection or a dynamically created object. The IDE's left sidebar shows a project tree with a package `Tytul_ksiazki` containing a `dodaj_tytul` method. The right sidebar contains the `Palette` and `Properties` windows.

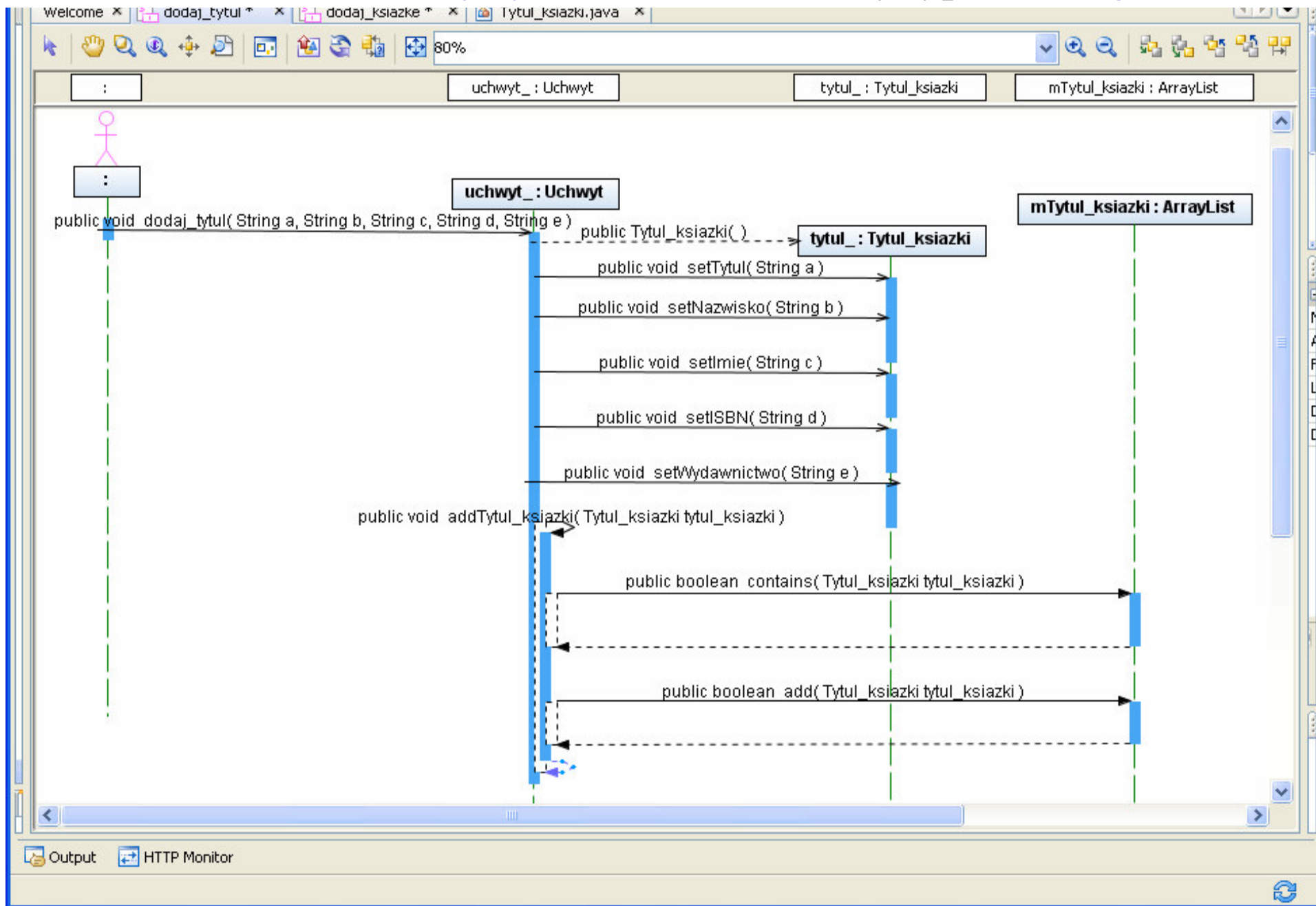
Zmiana scenariusza dodaj_tytul - dodanie metody typu Message To Self

The screenshot displays the NetBeans IDE interface with a UML sequence diagram for the `dodaj_tytul` method. The diagram shows the following interactions:

- An actor (represented by a stick figure) sends a message `public void dodaj_tytul(String a, String b, String c, String d, String e)` to the `uchwyty : Uchwyty` object.
- The `uchwyty : Uchwyty` object sends several messages to the `tytul : Tytul_książki` object:
 - `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)`
- The `uchwyty : Uchwyty` object sends a message `public void addTytul_książki(Tytul_książki tytul_książki)` to the `mTytul_książki : ArrayList` object.
- The `mTytul_książki : ArrayList` object sends a message `public boolean contains(Tytul_książki tytul_książki)` back to the `tytul : Tytul_książki` object.

At the bottom of the IDE, a **Refactoring** dialog is open, showing a rename operation: `Rename katalog1 to katalog2 [2 occurrences] - NEEDS REFRESH`. The dialog includes buttons for `Do Refactoring` and `Cancel`.

Zmiana scenariusza `dodaj_tytul` - dodanie metody typu **Message To Self**

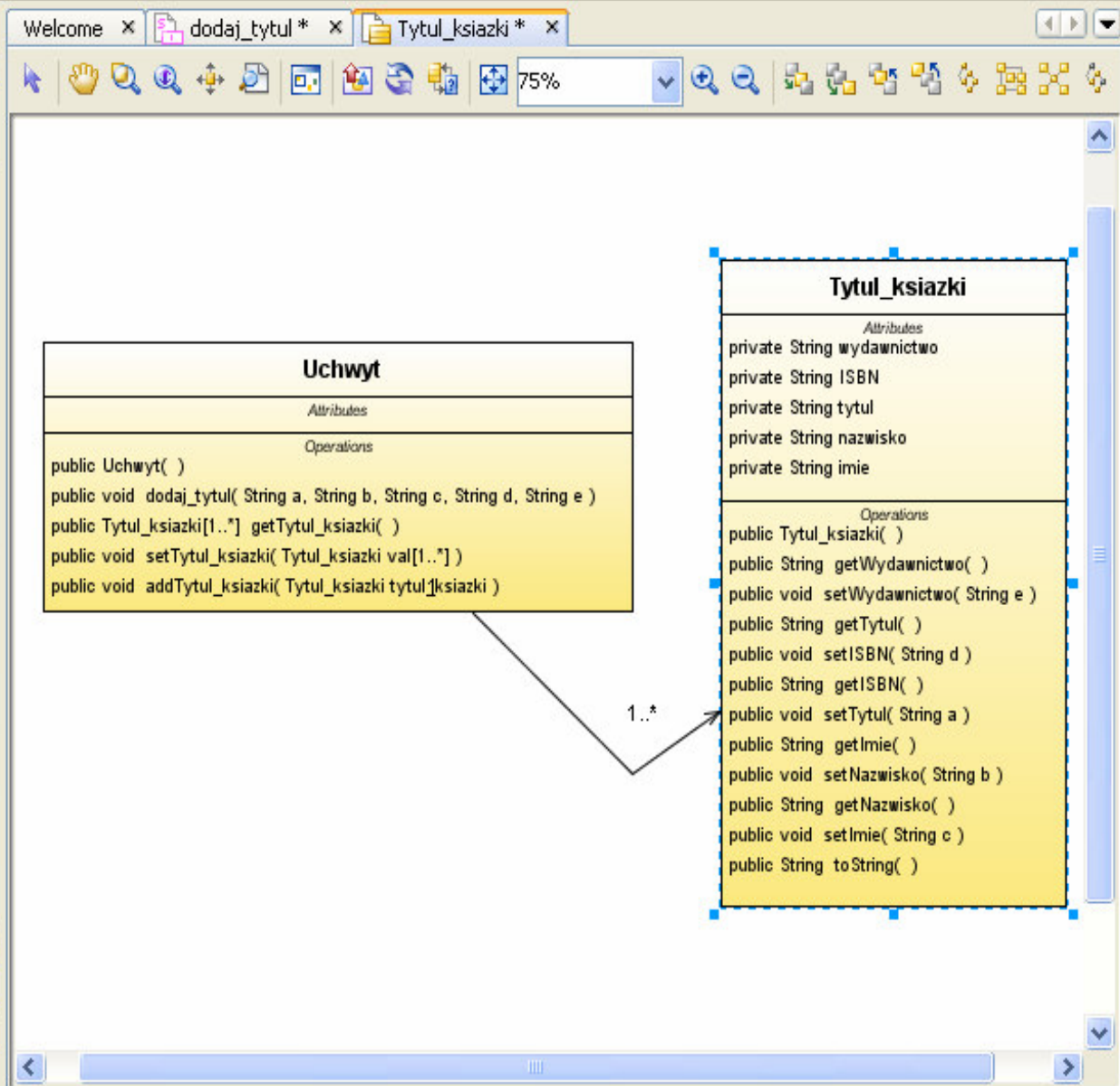


Pr... x Files Run...

- Interfejs1
- Interfejs2
- Interfejs3
- Interfejs4
- Interfejs5
- Interfejs6
- Interfejs7
- Interfejs8
- JavaAp1
- Katalog3
- Komunikator1
- komunikator2
- komunikator3
- Rys1
- Rys2
- Rys3
- Rys4
- UMLKatalog1
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements

Navigator x Outline

<No View Available>



Palette

- Basic
 - Class
 - Interface
 - Package
 - Collaboration Lifeli

equals - Prope...

Operation

Name	equals
Alias	equals
Return Type	bo...
Parameters	pu...
Visibility	p...
Documentation	
Stereotypes	
Tagged Value	
Abstract	<input type="checkbox"/>

equals

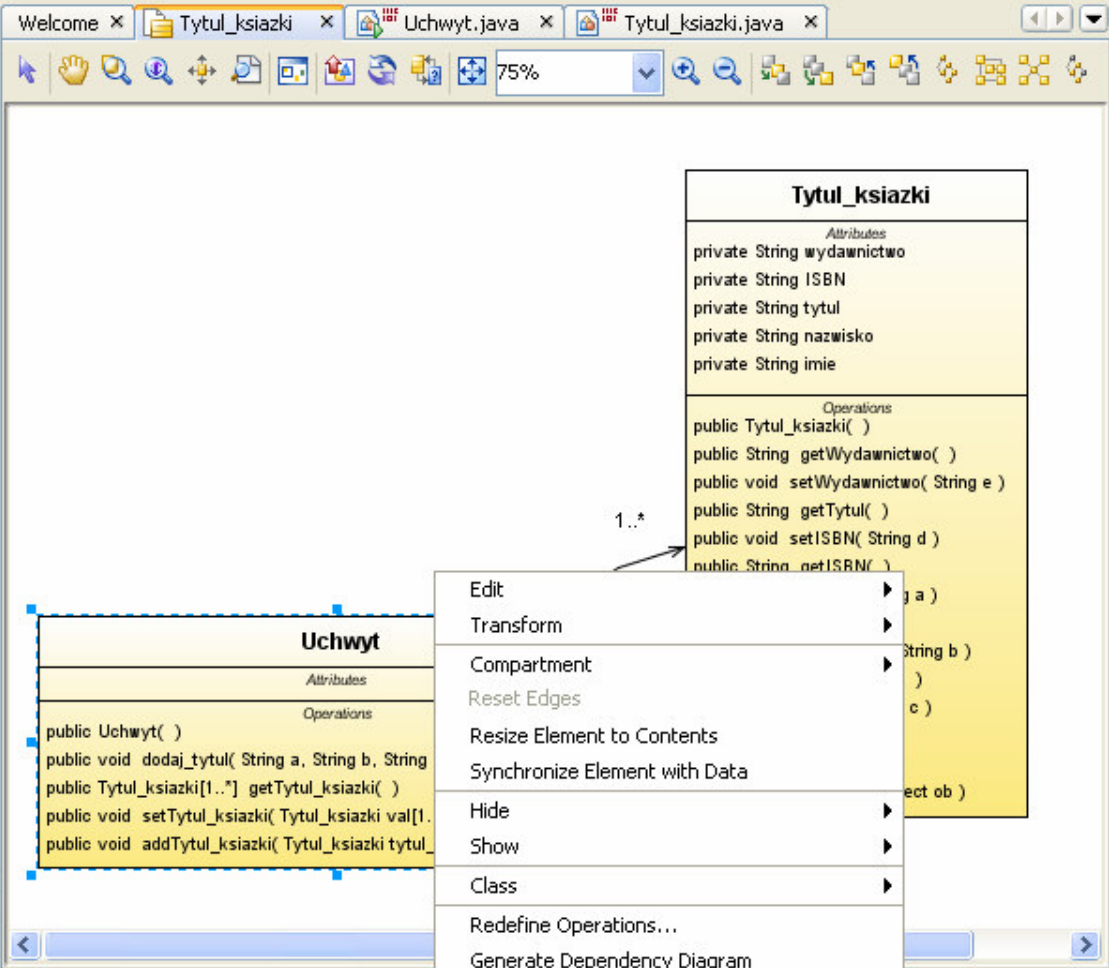
Dynamic Help

Finding Informati



Proj... Files Runtime

- Katalog3
 - Source Packages
 - katalog3
 - Tytul_ksiazki.java
 - Uchwyt.java
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_ksiazki
 - Wypożyczalnia
 - Imported Elements



- 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

Palette

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

Uchwyt - UML Documenta...

Uchwyt - Properties

Class	
Name	Uchwyt
Alias	Uchwyt
Visibility	public

Uchwyt

Dynamic Help

Finding Information About the

Navigator Outline

<No View Available>

Refactoring

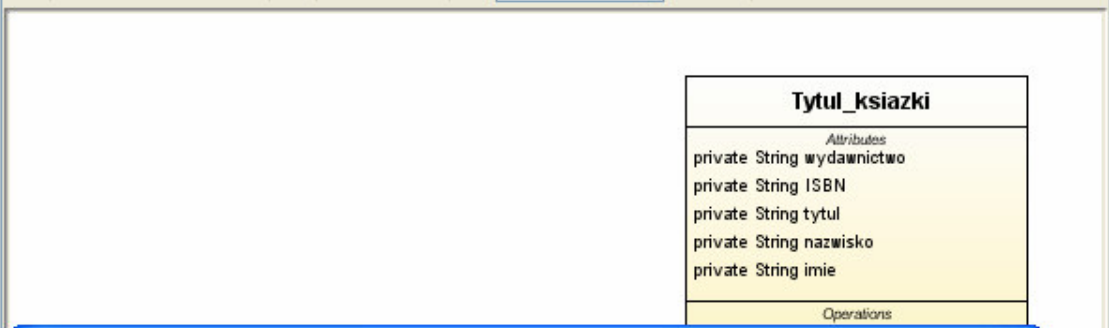
Output HTTP Monitor

Proj... Files Runtime

- Katalog3
 - Source Packages
 - katalog3
 - Tytul_książki.java
 - Uchwyty.java
 - Test Packages
 - Libraries
 - Test Libraries
- UMLKatalog1
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki
 - Wypożyczalnia
 - Imported Elements

Welcome Tytul_książki Uchwyty.java Tytul_książki.java

75%



Generate Code

Folder Location: Browse...

Backup Existing Source Files

OK Cancel Help

```
public void addTytul_książki( Tytul_książki tytul_książki )
```

Palette

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

Uchwyty - UML Documenta...

Uchwyty - Properties

Name	Uchwyty
Alias	Uchwyty
Visibility	public

Uchwyty

Dynamic Help

Finding Information About the

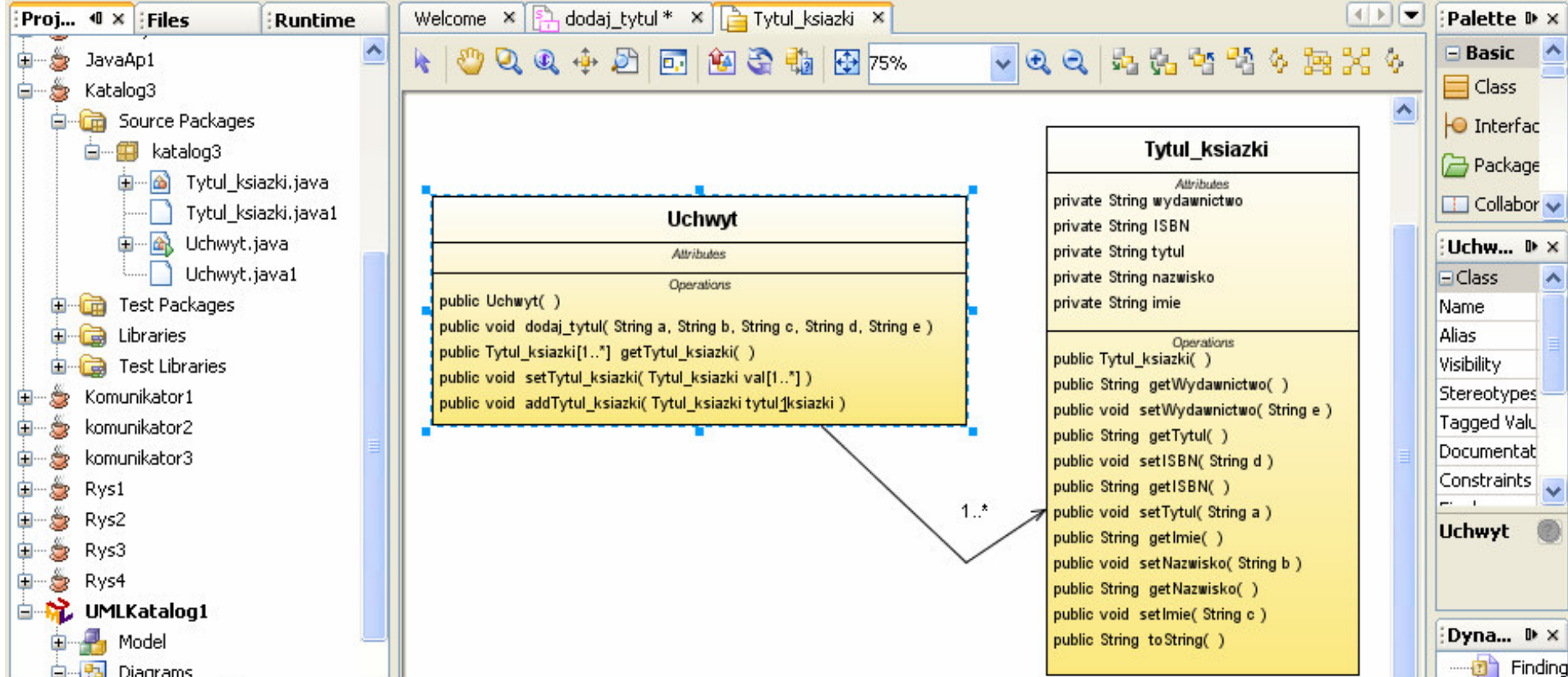
Navigator Outline

<No View Available>

Refactoring Usages

<No Usages>

Output HTTP Monitor



Output - Generate Code Log

Begin processing Generate Code: 1 items

Code Generation selected options:

Source Folder Location C:\Documents and Settings\Kruczkiewicz\rozne\dydaktyka\PI0\wk2\P3\Katalog3\src\katalog3

Processing element 1 of 1: Class Uchwyt

=====

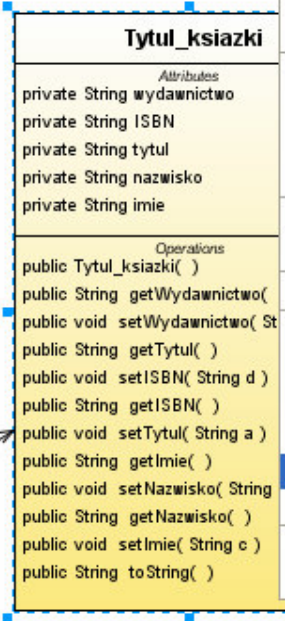
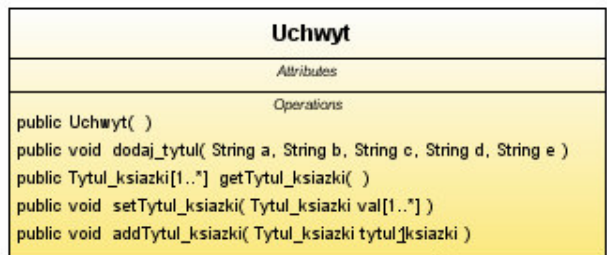
Task Successful (total time: 6 seconds)

Proj... Files Runtime

- JavaAp1
- Katalog3
 - Source Packages
 - katalog3
 - Tytul_książki.java
 - Tytul_książki.java1
 - Uchwyt.java
 - Uchwyt.java1
- Test Packages
- Libraries
- Test Libraries
- Komunikator1
- komunikator2
- komunikator3
- Rys1
- Rys2
- Rys3
- Rys4
- UMLKatalog1
 - Model
 - Diagrams
 - dodaj_tytul
 - toString
 - Tytul_książki

Welcome x dodaj_tytul * x Tytul_książki x

75%



1..*

- 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

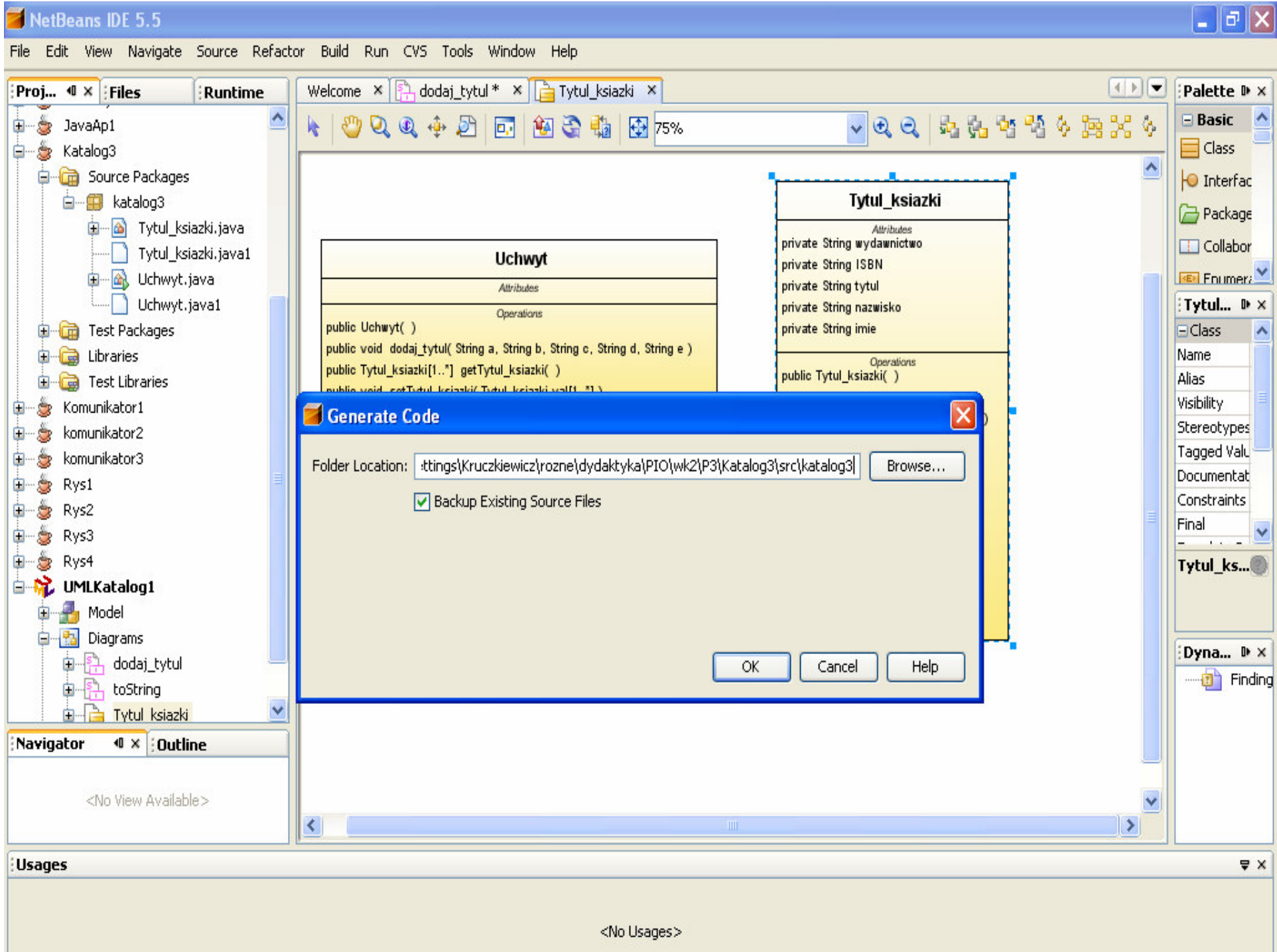
Navigator Outline

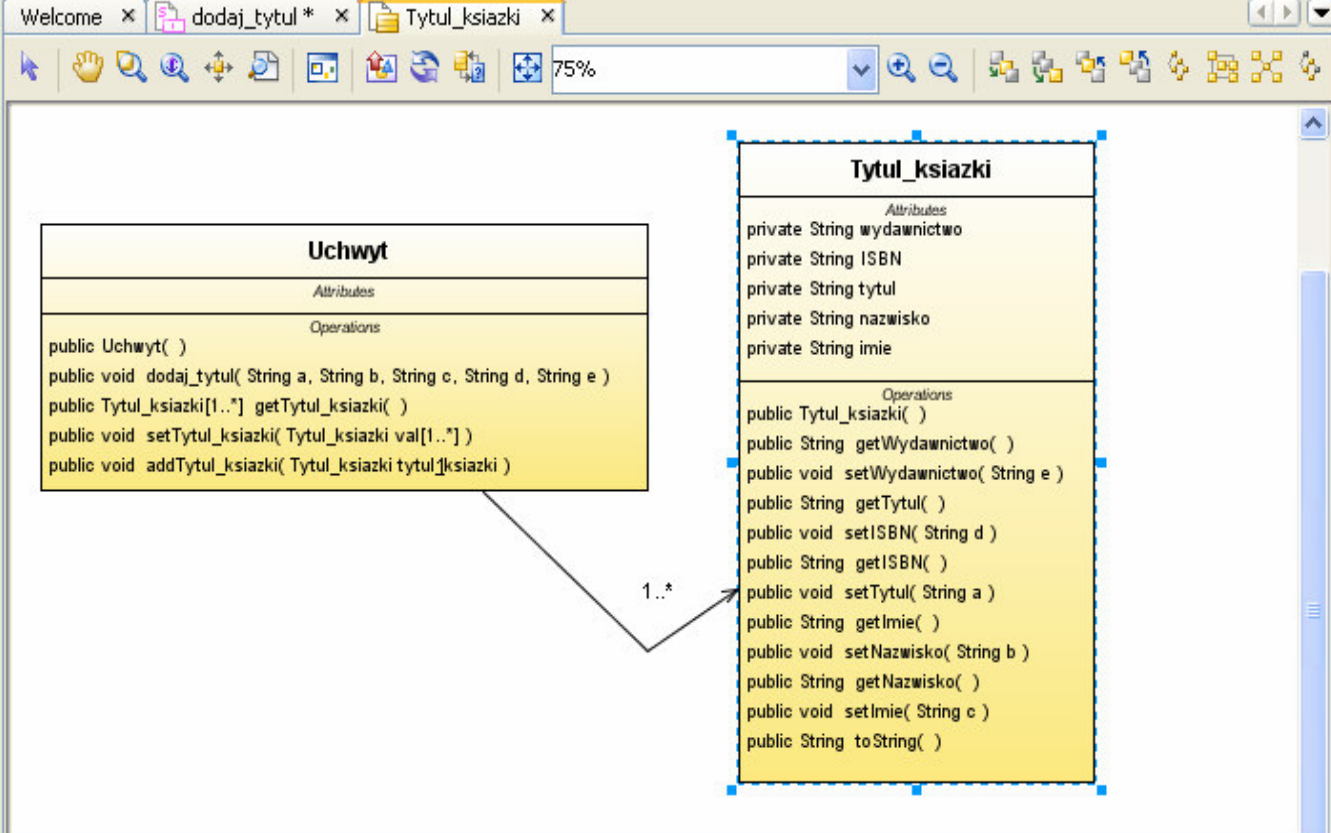
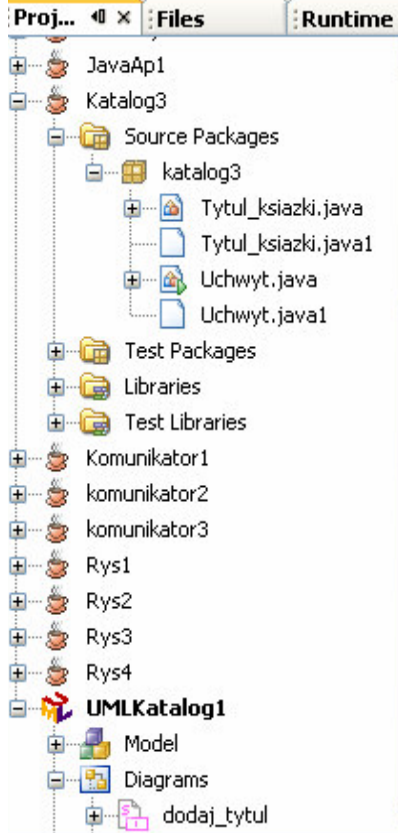
<No View Available>

Usages

<No Usages>

Output HTTP Monitor





Palette

- Basic
 - Class
 - Interface
 - Package
 - Collabor
 - Frumerz
- Tytul...
 - Class
 - Name
 - Alias
 - Visibility
 - Stereotypes
 - Tagged Val
 - Documentat
 - Constraints
 - Final
- Dyna...
 - Findin

Output - Generate Code Log

```

Begin processing Generate Code: 1 items

Code Generation selected options:
  Source Folder Location C:\Documents and Settings\Kruczkiewicz\rozne\dydaktyka\PI0\wk2\P3\Katalog3\src\katalog3

Processing element 1 of 1: Class Tytul_książki

=====
Task Successful (total time: 5 seconds) |
    
```

Uzupełniono kod metody `addTytul_książki`, wg scenariusza ze zmodyfikowanego diagramu sekwencji `dodaj_tytul`

```
public class Uchwyt {  
    private ArrayList<Tytul_książki> mTytul_książki = new ArrayList<Tytul_książki>();  
    public Uchwyt() { }  
    public void dodaj_tytul(String a, String b, String c, String d, String e) {  
        Tytul_książki tytul_książki=new Tytul_książki();  
        tytul_książki.setTytul(a);  
        tytul_książki.setNazwisko(b);  
        tytul_książki.setImie(c);  
        tytul_książki.setISBN(d);  
        tytul_książki.setWydawnictwo(e);  
        addTytul_książki(tytul_książki); }  
    public ArrayList<Tytul_książki> getTytul_książki() { return mTytul_książki;}  
    public void setTytul_książki(ArrayList<Tytul_książki> val)  
        { this.mTytul_książki = val;}  
    public void addTytul_książki(Tytul_książki tytul_książki)  
        { if (! this.mTytul_książki.contains(tytul_książki))  
            this.mTytul_książki.add(tytul_książki); }  
}
```

```

public class Uchwyt {
    private ArrayList<Tytul_ksiazki> mTytul_ksiazki = new ArrayList<Tytul_ksiazki>();
    public Uchwyt() { }
    public void dodaj_tytul(String a, String b, String c, String d, String e) {
        Tytul_ksiazki tytul_ksiazki=new Tytul_ksiazki();
        tytul_ksiazki.setTytul(a);
        tytul_ksiazki.setNazwisko(b);
        tytul_ksiazki.setImie(c);
        tytul_ksiazki.setISBN(d);
        tytul_ksiazki.setWydawnictwo(e);
        addTytul_ksiazki(tytul_ksiazki); }
    public ArrayList<Tytul_ksiazki> getTytul_ksiazki()           { return mTytul_ksiazki; }
    public void setTytul_ksiazki(ArrayList<Tytul_ksiazki> val) { this.mTytul_ksiazki = val; }
    public void addTytul_ksiazki(Tytul_ksiazki tytul_ksiazki)
        { if (! this.mTytul_ksiazki.contains(tytul_ksiazki))
            this.mTytul_ksiazki.add(tytul_ksiazki); }
    public static void main(String t[]) // your code here
    { Uchwyt ap=new Uchwyt();
        ap.dodaj_tytul("1","1","1","1","1");
        ap.dodaj_tytul("2","2","2","2","2");
        ap.dodaj_tytul("2","2","2","2","2");
        String lan=ap.getTytul_ksiazki().toString();
        System.out.println(lan); } }

```

```

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 _wydawnictwo) { wydawnictwo = _wydawnictwo; }

  public String getISBN() {return ISBN; }

  public void setISBN(String _ISBN) {ISBN = _ISBN; }

  public String getTytul() {return tytul; }

  public void setTytul(String _tytul) {tytul = _tytul; }

  public String getNazwisko() {return nazwisko;}

  public void setNazwisko(String _nazwisko) {nazwisko = _nazwisko;}

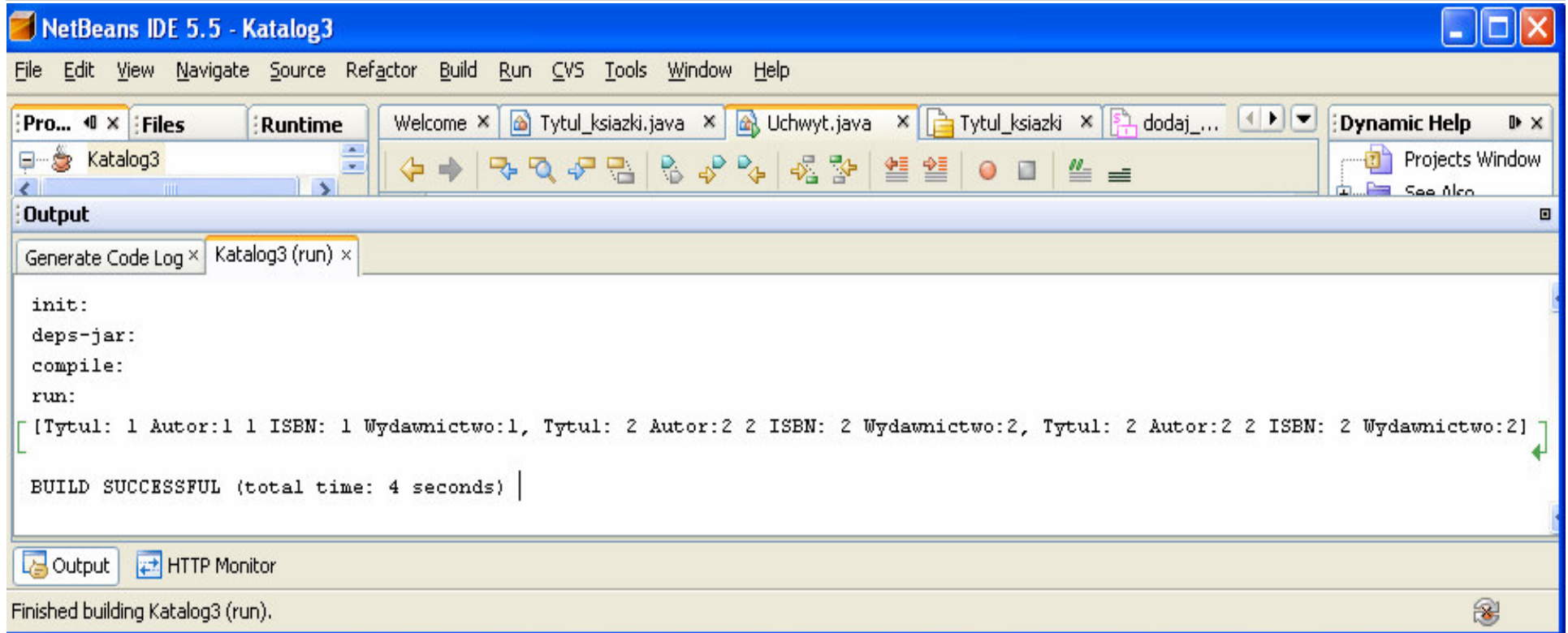
  public String getImie() {return imie; }

  public void setImie(String _imie) {imie = _imie; }

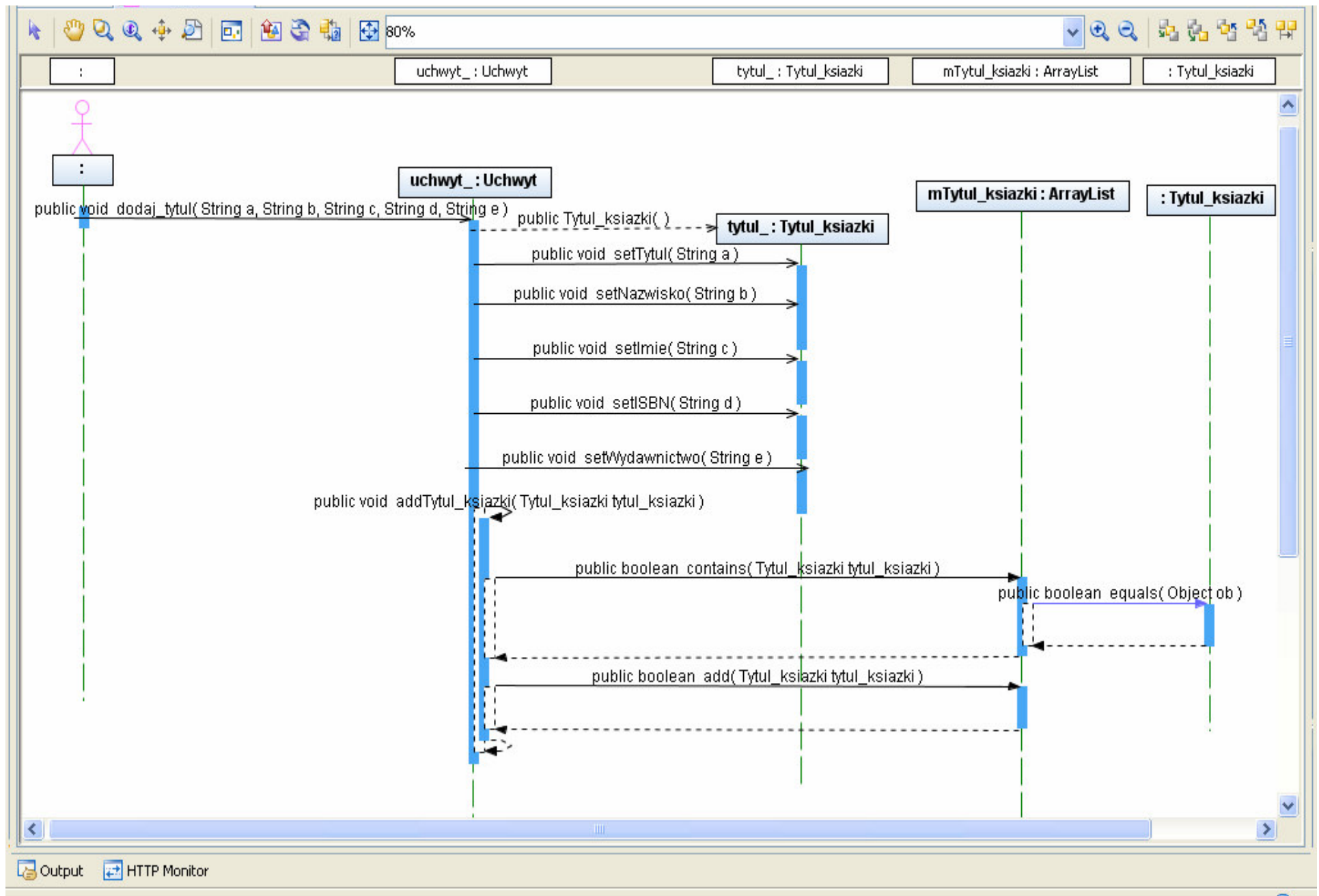
  public String toString() // your code here
  { String pom="Tytul: "+getTytul();
    pom+=" Autor: "+getNazwisko() + " "+getImie();
    pom+=" ISBN: "+getISBN();
    pom+=" Wydawnictwo: "+getWydawnictwo();
    return pom;
  }
}

```

Skutki wywołania metody `equals` w metodzie `contains`,
dziedziczonej od klasy `Object`, która nie uwzględnia właściwości
klasy dziedziczącej `Tytul_książki`



Uzupełniono scenariusz metody `dodaj_tytul` o wywołanie metody `equals` dziedziczonej od klasy `Object`, którą przesłania w metodzie `contains`



Uzupełniona definicja klasy Tytul_książki o metodę `equals`, która przesłania metodę `equals` dziedziczoną od klasy `Object`

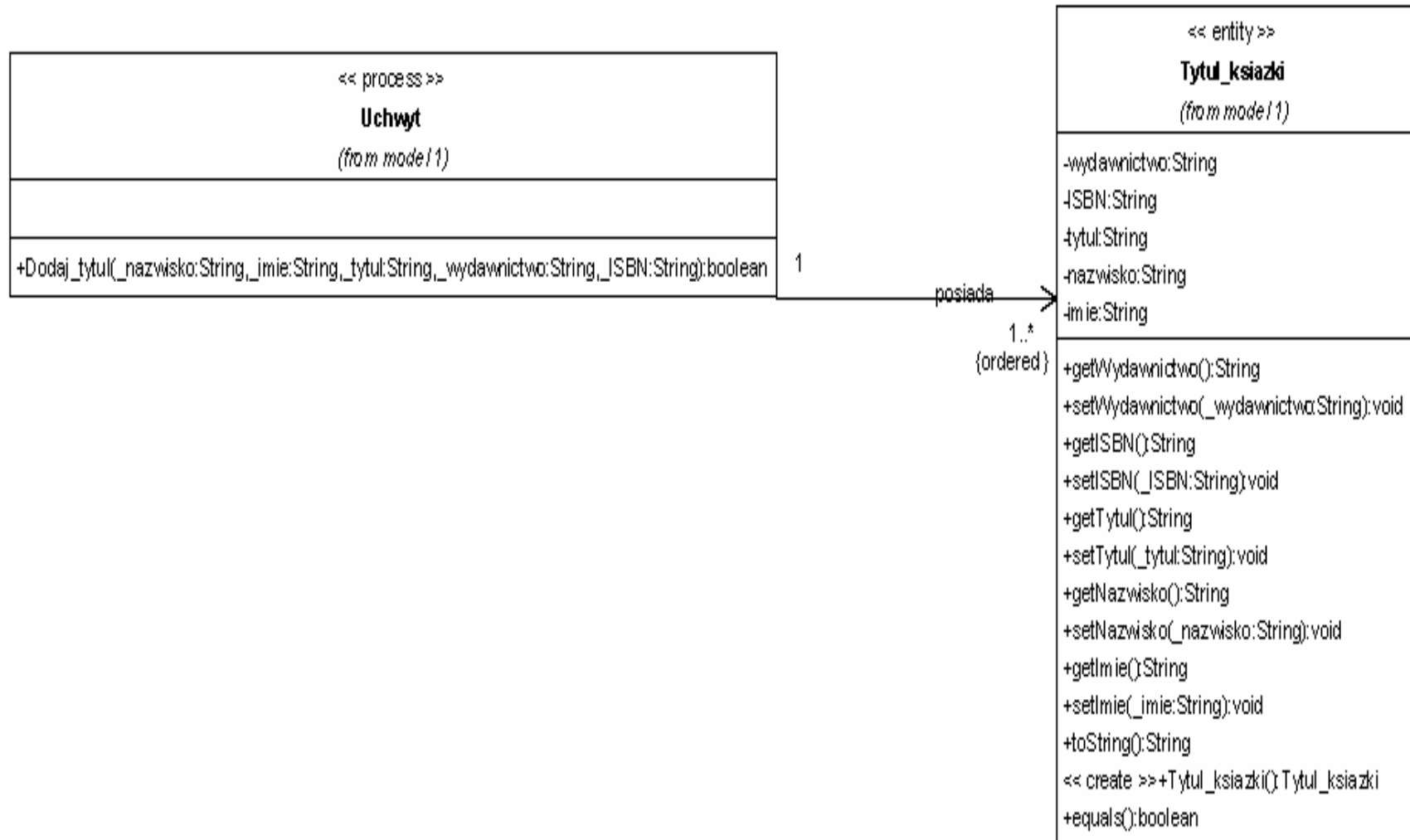
The screenshot shows an IDE with the following components:

- Left Panel (Project Explorer):** Shows a project structure with folders like 'katalog3', 'Test Packages', 'Libraries', and 'UMLKatalog1'. Under 'UMLKatalog1', there is a 'Diagrams' folder containing 'Tytul_książki'.
- Top Panel (Editors):** Displays several open files: 'Welcome', 'dodaj_tytul', 'Tytul_książki', 'Uchwy.java', and 'Tytul_książki.java...'. The 'Tytul_książki' diagram is active.
- UML Class Diagram:** Shows two classes: 'Uchwy' and 'Tytul_książki'.
 - Uchwy Class:**
 - Attributes: (None listed)
 - Operations:

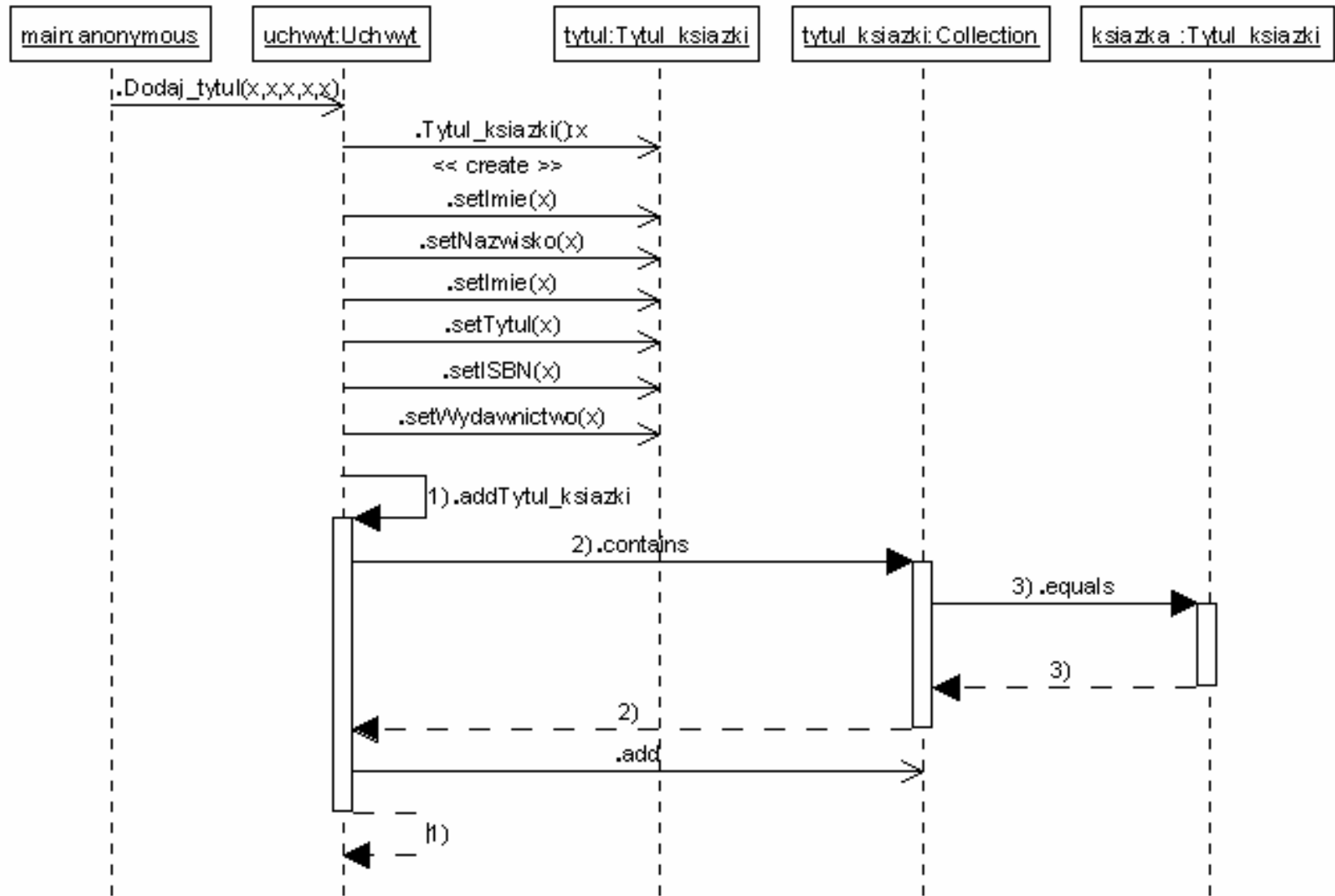
```
public Uchwy( )
public void dodaj_tytul( String a, String b, String c, String d, String e )
public Tytul_książki[1..*] getTytul_książki( )
public void setTytul_książki( Tytul_książki val[1..*])
public void addTytul_książki( Tytul_książki tytul_książki )
```
 - Tytul_książki Class:**
 - Attributes:

```
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 e )
public String getTytul( )
public void setISBN( String d )
public String getISBN( )
public void setTytul( String a )
public String getImie( )
public void setNazwisko( String b )
public String getNazwisko( )
public void setImie( String c )
public String toString( )
public boolean equals( Object ob )
```
- Relationship:** An association arrow points from 'Uchwy' to 'Tytul_książki' with the multiplicity '1..*' at the 'Tytul_książki' end.
- Right Panel (Palette):** Shows a 'Basic' palette with 'Class', 'Interface', 'Package', and 'Collaboration Li'.
- Properties Panel:** Shows details for the selected 'Tytul_książki' diagram, including Name, Alias, File Name, Layout Style, Diagram Kind, and Documentatic.
- Dynamic Help:** Shows 'Projects Wind' and 'See Also'.
- Bottom Panel (Usages):** Displays '<No Usages>'. Below it are 'Output' and 'HTTP Monitor' tabs.



Dodaj_tytul



Uzupełniono kod metody `equals`, która przesłania metodę `equals` dziedziczoną od klasy `Object`

The screenshot displays the NetBeans IDE 5.5 interface. The main editor window shows the source code for `Tytul_książki.java`. The `equals` method is highlighted in yellow, indicating it is the current focus. The code defines a class with several methods: `getImie`, `setNazwisko`, `getNazwisko`, `setImie`, `toString`, and `equals`. The `toString` method constructs a string representation of the object, and the `equals` method compares the ISBN of the current object with the ISBN of the provided object.

```
44     this.tytul = a;
45 }
46
47 public String getImie() {
48     return nazwisko;
49 }
50
51 public void setNazwisko(String b) {
52     this.nazwisko = b;
53 }
54
55 public String getNazwisko() {
56     return imie;
57 }
58
59 public void setImie(String c) {
60     this.imie = c;
61 }
62
63 public String toString() {
64     String pom="Tytul: "+getTytul();
65     pom+=" Autor: "+getNazwisko() + " "+getImie();
66     pom+=" ISBN: "+getISBN();
67     pom+=" Wydawnictwo: "+getWydawnictwo();
68     return pom;
69 }
70
71 public boolean equals(Object ob) {
72     boolean a=ISBN.equals(((Tytul_książki)ob).getISBN());
73     System.out.println(a);
74     return a;
75 }
76 }
```

The IDE interface includes a Project Explorer on the left showing the project structure, a Navigator window below it, and a Refactoring window at the bottom. The Windows taskbar at the bottom shows the Start button and several open applications, including Windows Commander, Wykład2_2, and NetBeans IDE 5.5 - K...

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()
```

```
public void setWydawnictwo(String _wydawnictwo)
```

```
public String getISBN()
```

```
public void setISBN(String _ISBN)
```

```
public String getTytul()
```

```
public void setTytul(String _tytul)
```

```
public String getNazwisko()
```

```
public void setNazwisko(String _nazwisko)
```

```
public String getImie()
```

```
public void setImie(String _imie)
```

```
public boolean equals(Object ob)
```

```
{ boolean a=ISBN.equals(((Tytul_książki)ob).getISBN());
```

```
  //System.out.println(a);
```

```
  return a; }
```

```
public String toString() // your code here
```

```
{ String pom="Tytul: "+getTytul();  
  pom+=" Autor:"+getNazwisko() +" "+getImie();  
  pom+=" ISBN: "+getISBN();  
  pom+=" Wydawnictwo:"+getWydawnictwo();  
  return pom; } }
```

```
{}
```

```
{return wydawnictwo; }
```

```
{ wydawnictwo = _wydawnictwo; }
```

```
{ return ISBN; }
```

```
{ ISBN = _ISBN; }
```

```
{ return tytul; }
```

```
{ tytul = _tytul; }
```

```
{ return nazwisko; }
```

```
{ nazwisko = _nazwisko;}
```

```
{ return imie; }
```

```
{ imie = _imie; }
```

```
//your code here
```

NetBeans IDE 5.5 - Katalog3

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

Projects Files Runtime ...lnia Tytul_ksiazki.java... Dynamic ...

Katalog3 Source Packages

44 | this tytul = a*

Projects Window See Also

Output

Generate Code Log x Katalog3 (run) x

```
init:
deps-jar:
compile:
run:
false
false
true
[ [Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1, Tytul: 2 Autor:2 2 ISBN: 2 Wydawnictwo:2] ]
BUILD SUCCESSFUL (total time: 2 seconds)
```

Output HTTP Monitor

Finished building Katalog3 (run).