

Instrukcja 3

Laboratorium z Podstaw Inżynierii Oprogramowania

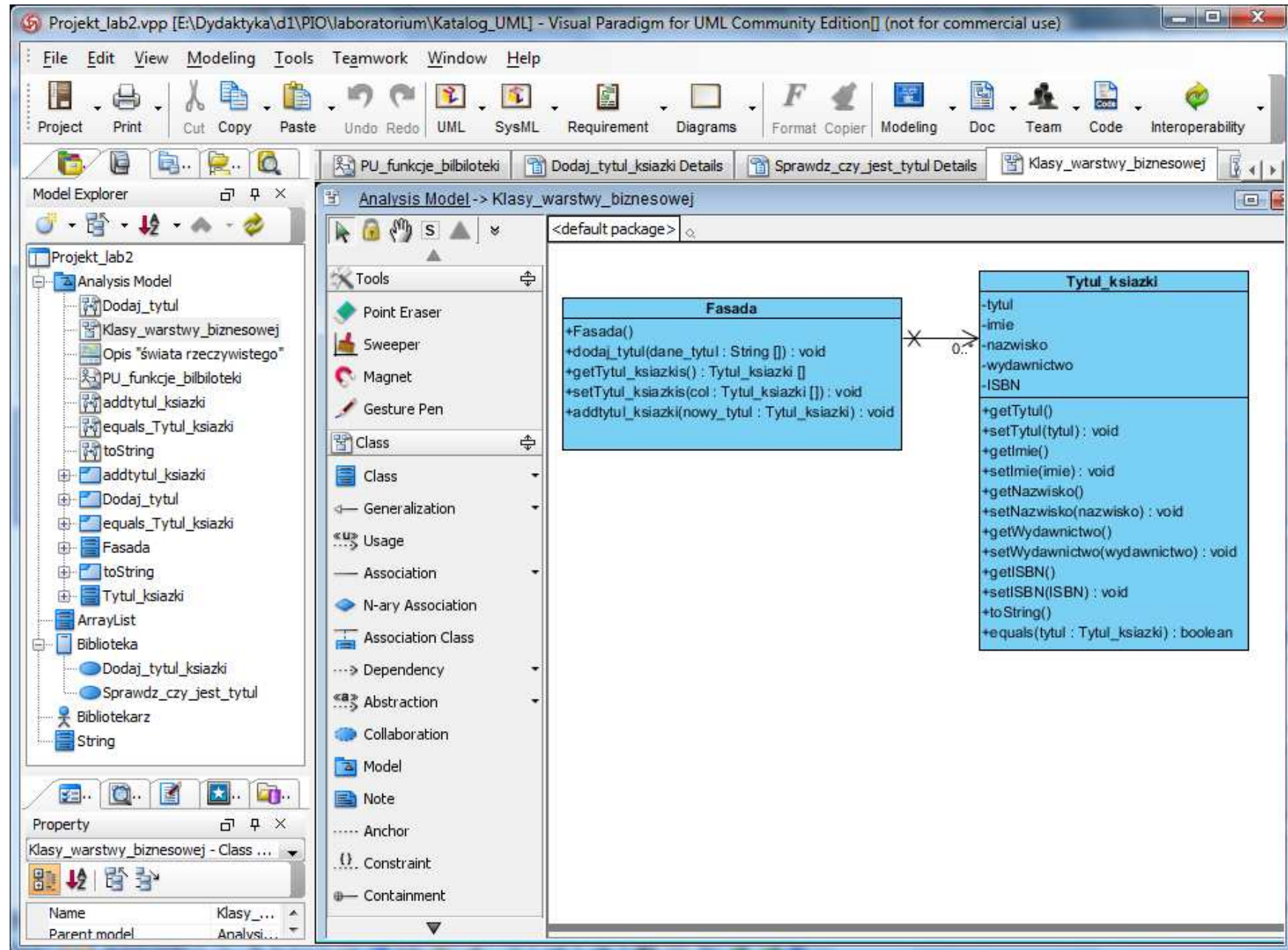
Diagramy klas i sekwencji:

Relacja 1 do 0..* między klasami
modelu obiektowego

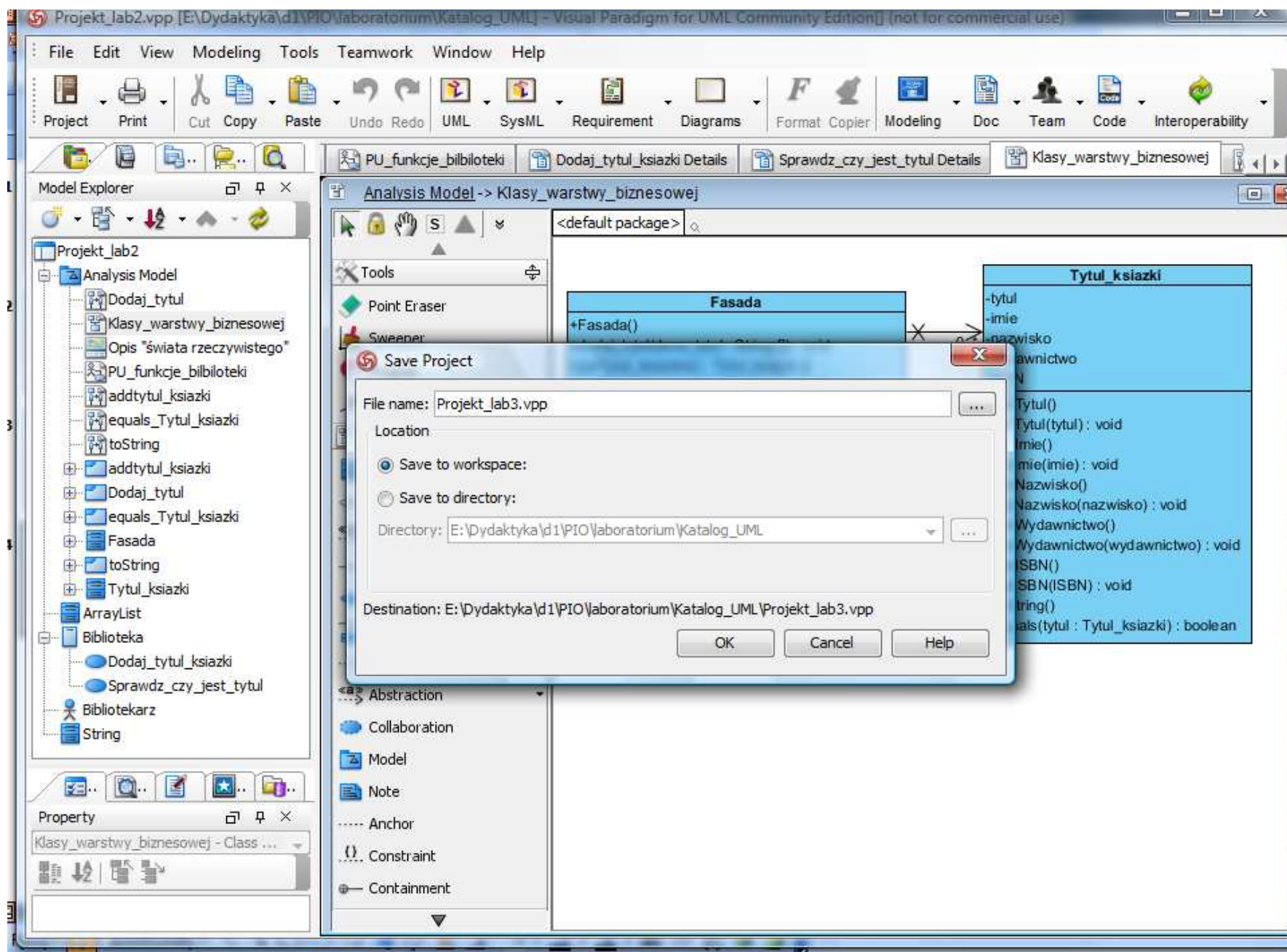
Cele laboratorium 3

- Dokończyć 1-y etap z lab2.
- 2-i etap projektu.
 - Instrukcja zawiera przykłady powiązań 1:* (jeden do wiele) między klasami reprezentującymi dane: z 1-go etapu (Tytuł_książki) oraz nową klasą Książka, która została zidentyfikowana po zaprojektowaniu kolejnego przypadku użycia PU Dodaj_książke.
 - Wykonaj kopię projektu UML z etapu 1 (laboratorium 2). Uzupełnij diagramy przypadków użycia, klas oraz uzupełnij i/lub dodaj nowe diagramy sekwencji. Należy wykorzystać te dwa typy powiązań 1:* oraz 1:0..1 w wykonywanym 2-im etapie projektu i jego implementacji na podstawie instrukcji do laboratorium 3.
 - Wykonaj prosty program stanowiący realizację 2-go etapu wybranego projektu. Program ten stanowi rozwinięcie programu z etapu 1. Instrukcja prezentuje przykładowy program.

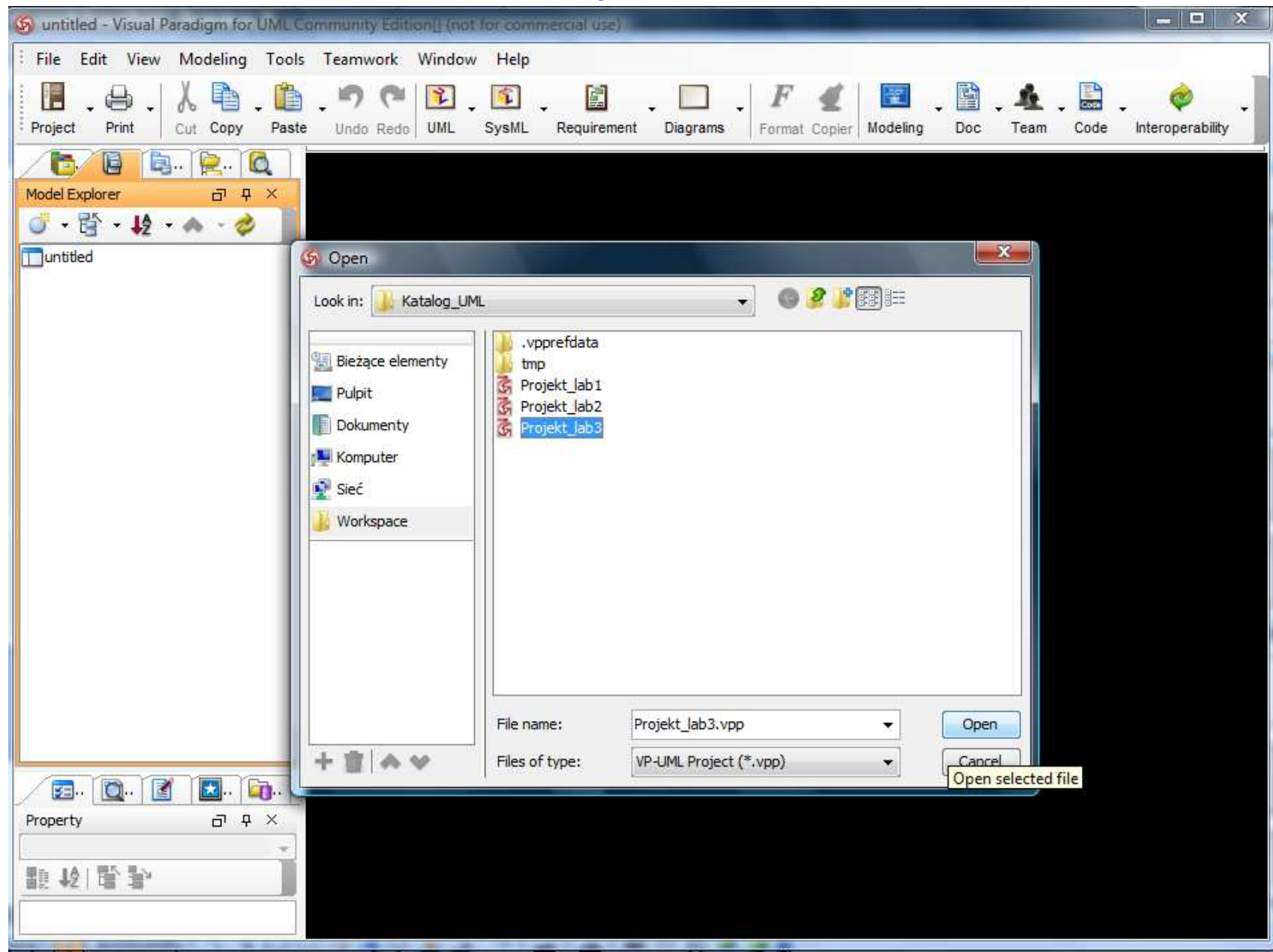
Stan projektu po zakończeniu etapu 1



Wykonanie kopii projektu Projekt_lab2 jako Projekt_lab3



Po zamknięciu projektu **Projekt_lab2** należy otworzyć projekt **Projekt_lab3**



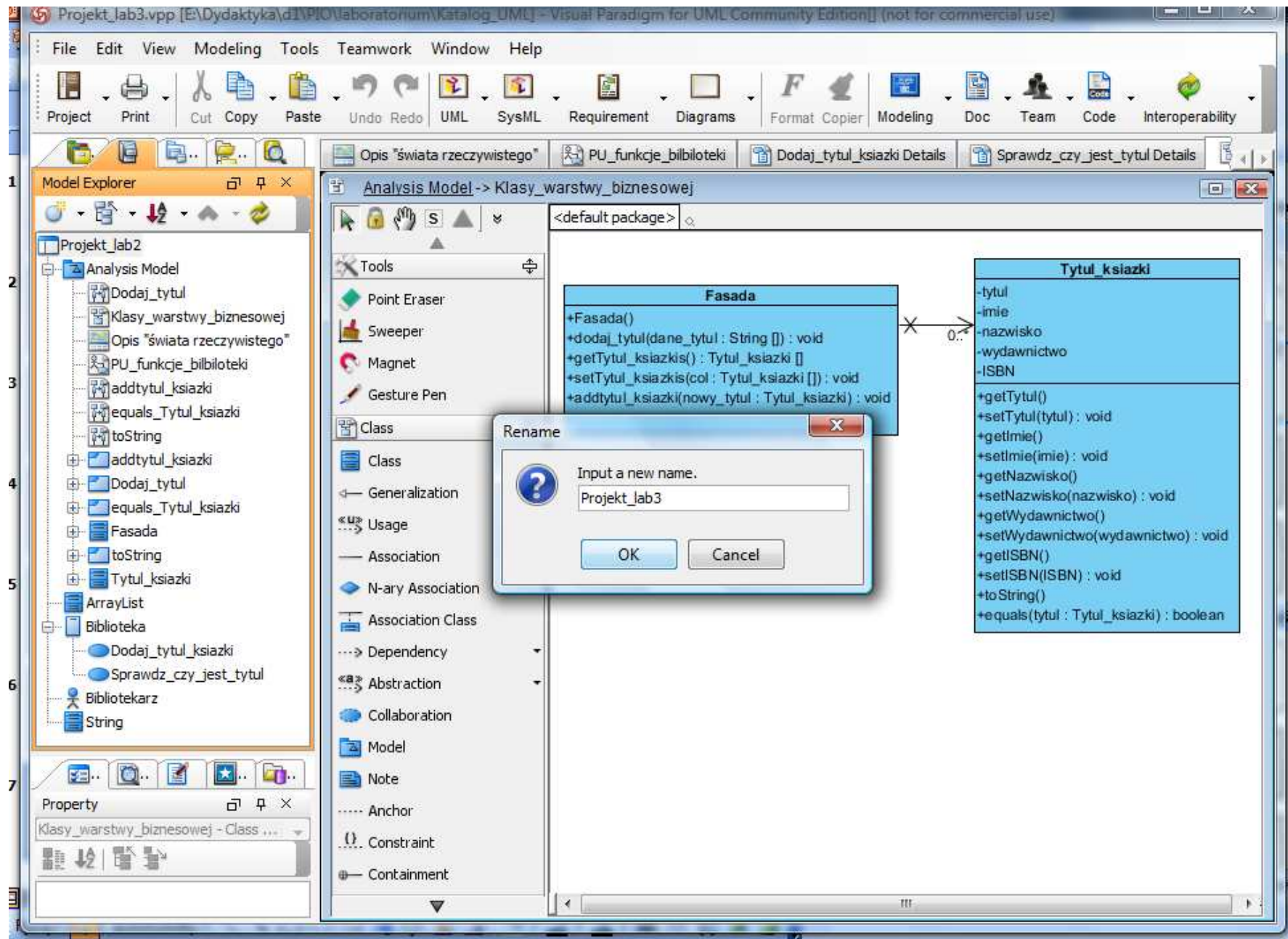
Należy nadać nazwę **Projekt_lab3** tworzonyj kopii projektu

The screenshot displays the Visual Paradigm for UML Community Edition interface. The main window shows a class diagram with two classes: **Fasada** and **Tytul_książki**. The **Fasada** class has methods: `+Fasada()`, `+dodaj_tytul(dane_tytul : String []) : void`, `+getTytul_książkis() : Tytul_książki []`, `+setTytul_książkis(col : Tytul_książki []) : void`, and `+addtytul_książki(nowy_tytul : Tytul_książki) : void`. The **Tytul_książki** class has attributes: `-tytul`, `-imie`, `-nazwisko`, `-wydawnictwo`, and `-ISBN`. It also has methods: `+getTytul()`, `+setTytul(tytul) : void`, `+getImie()`, `+setImie(imie) : void`, `+getNazwisko()`, `+setNazwisko(nazwisko) : void`, `+getWydawnictwo()`, `+setWydawnictwo(wydawnictwo) : void`, `+getISBN()`, `+setISBN(ISBN) : void`, `+toString()`, and `+equals(tytul : Tytul_książki) : boolean`. A dependency relationship is shown between the two classes, with a multiplicity of `0..*` at the **Tytul_książki** end.

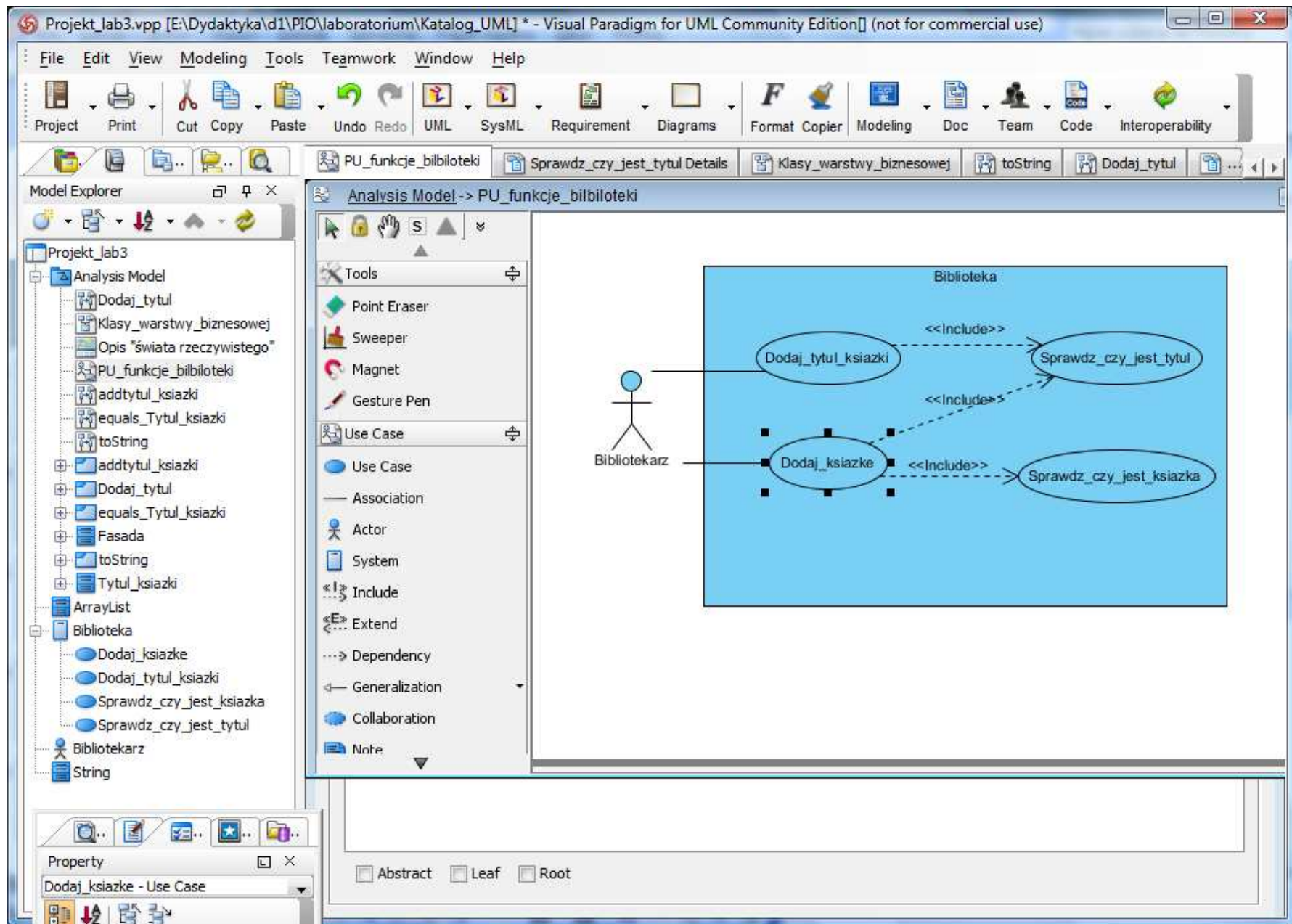
The **Model Explorer** on the left shows a project named **Projekt_lab3**. A context menu is open over this project, with the **Rename...** option highlighted. Other options include **Viewless Model Elements...**, **Merge Model Elements...**, **Nicknames**, **Model**, **Model Element**, **Configure Programming Language...**, **Sorting**, **Show Relationships**, **Show Sub Diagrams**, **Show Data Types**, **Teamwork**, **Refresh**, **Collapse All**, and **Expand All**.

The **Property** window at the bottom left shows the selected element is **Klasy_warstwy_biznesowej - Class ...**. The **Diagram** window at the bottom right shows the selected element is **Model**.

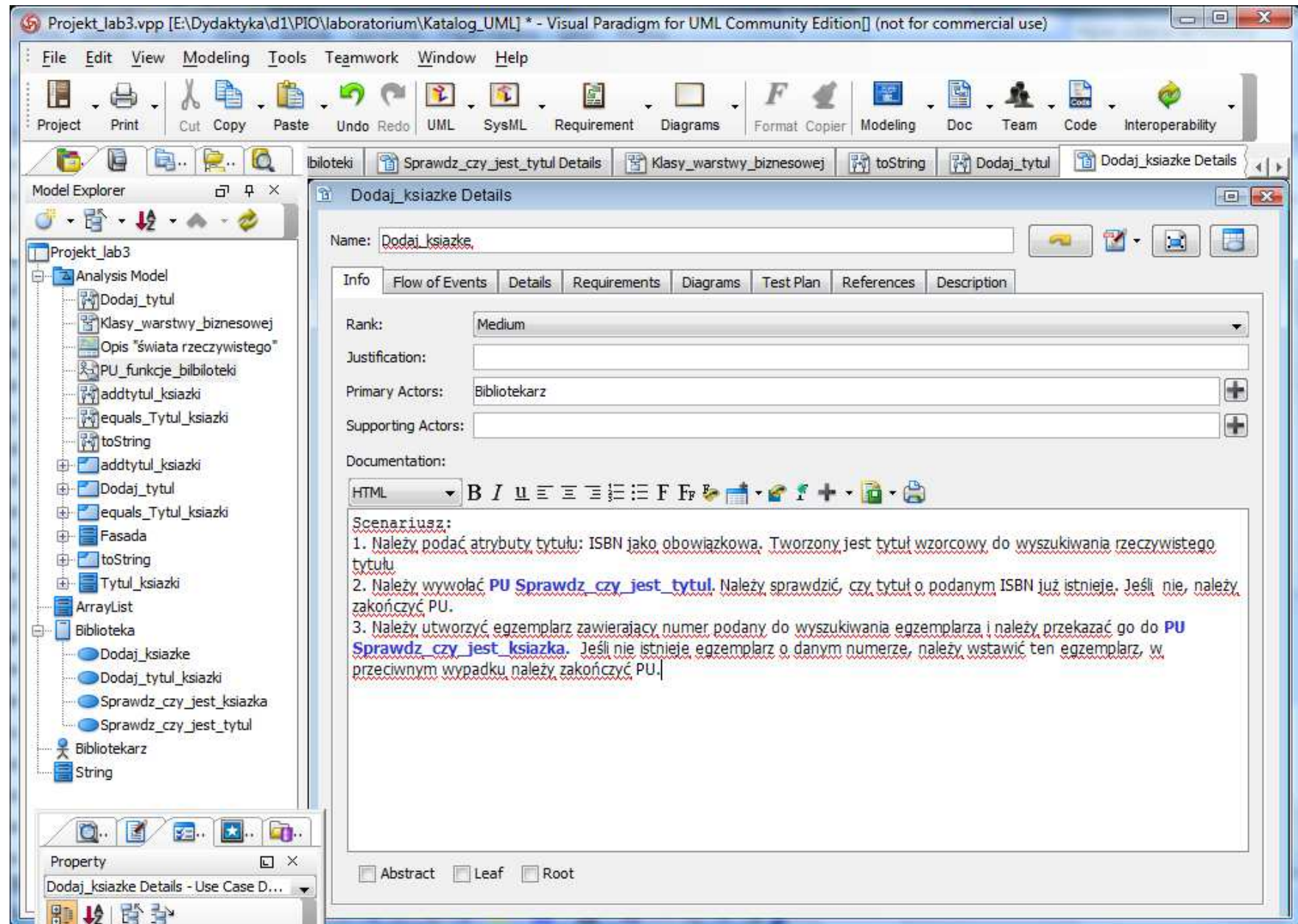
Zakończenie tworzenia kopii projektu Projekt_lab3



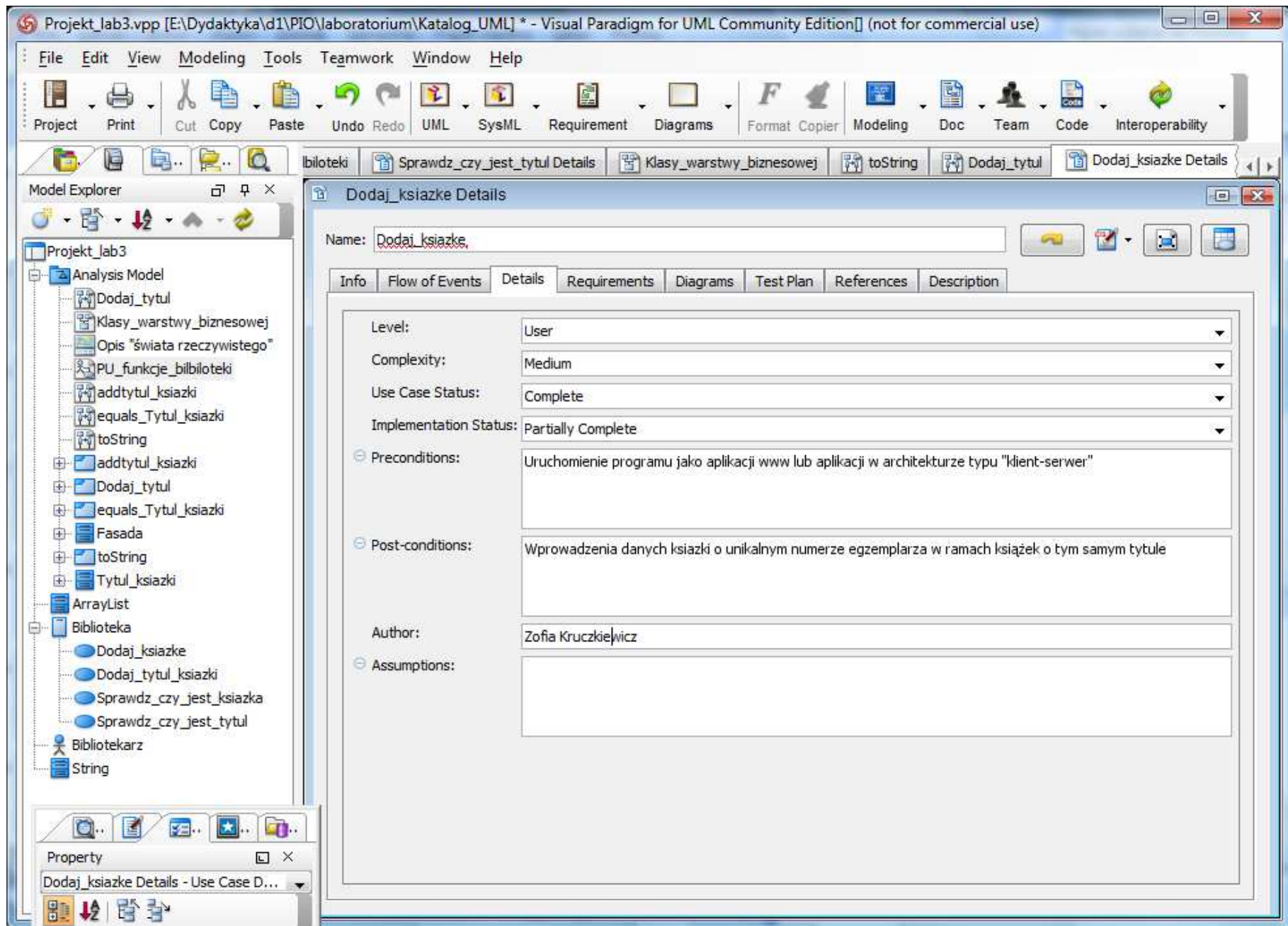
Dodanie nowych przypadków użycia: PU Dodaj_książke i PU Sprawdź_czy_jest_książka



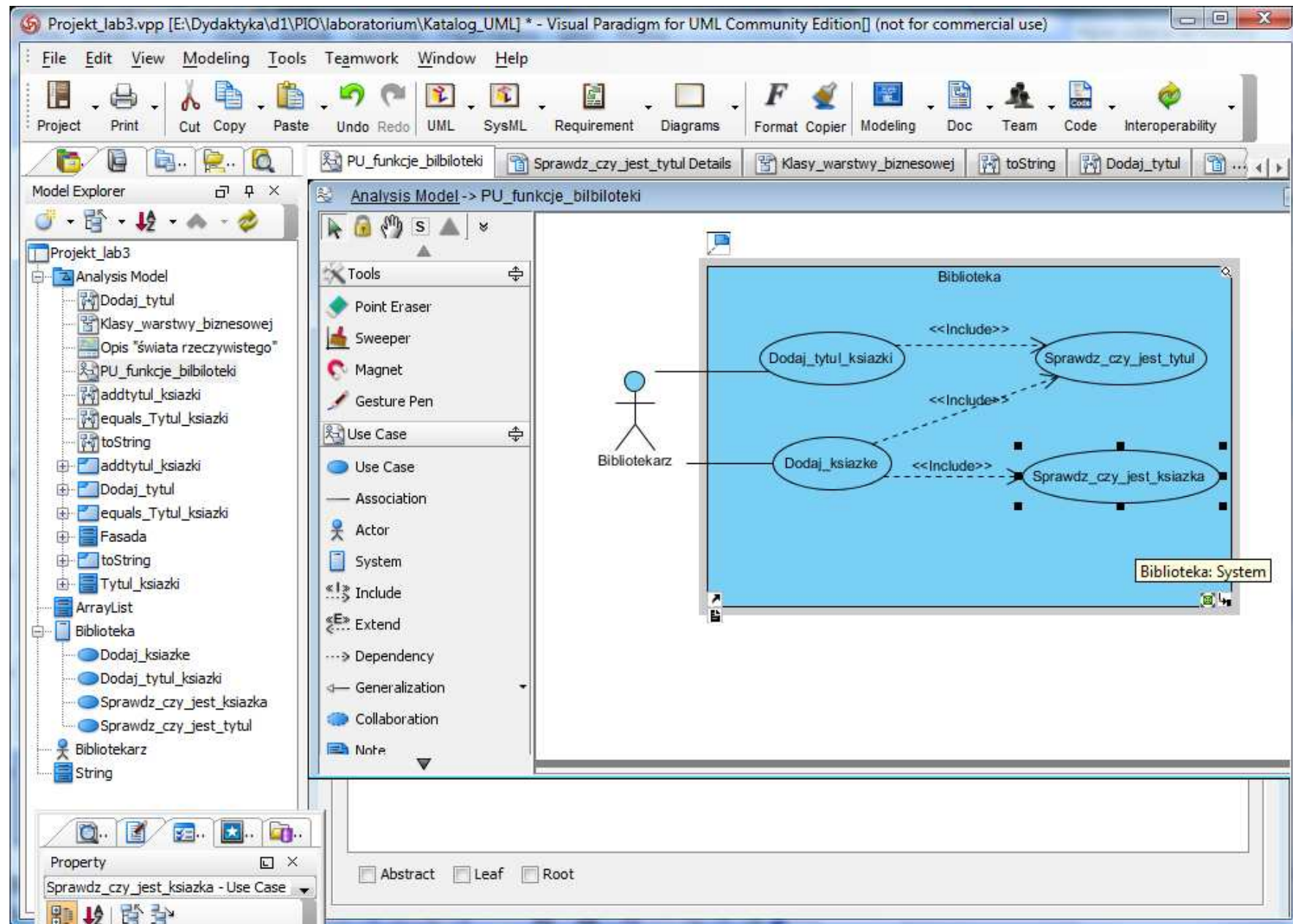
Po wybraniu prawym klawiszem **PU Dodaj_książke**, należy wybrać z listy pozycję *Open Use Case Details...* i wypełnić formularz w zakładce *Info*



Następnie należy wypełnić formularz wybrany z zakładki *Details*



Po wybraniu prawym klawiszem **PU Sprawdz_czy_jest_książka**, należy wybrać z listy pozycję *Open Use Case Details...*



Następnie należy wypełnić formularz w zakładce *Info*

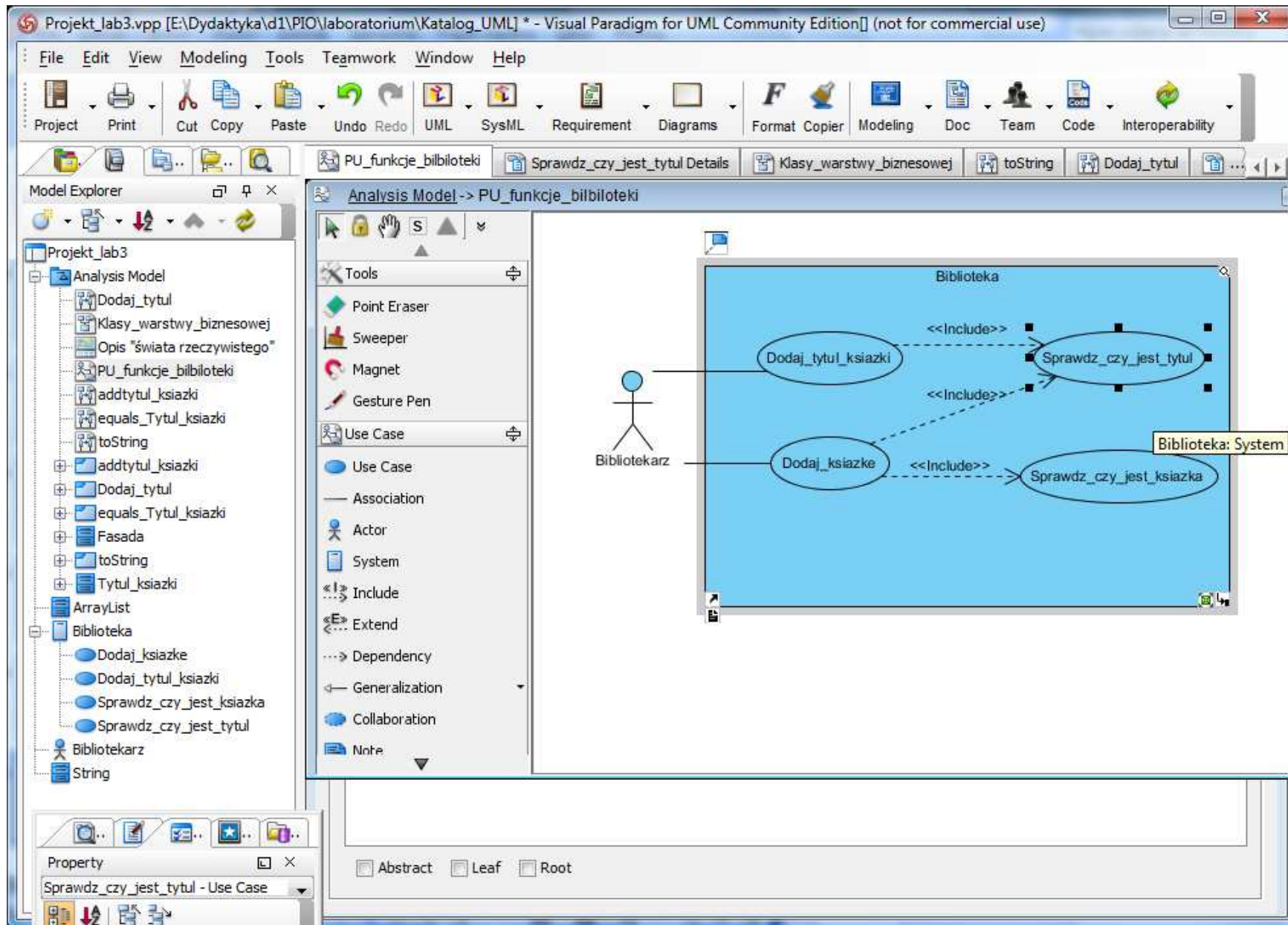
The screenshot displays the Visual Paradigm for UML Community Edition interface. The main window is titled "Projekt_lab3.vpp [E:\Dydaktyka\d1\PIO\laboratorium\Katalog_UML] * - Visual Paradigm for UML Community Edition[] (not for commercial use)". The menu bar includes File, Edit, View, Modeling, Tools, Teamwork, Window, and Help. The toolbar contains icons for Project, Print, Cut, Copy, Paste, Undo, Redo, UML, SysML, Requirement, Diagrams, Format, Copier, Modeling, Doc, Team, Code, and Interoperability. The Model Explorer on the left shows a project structure with folders like "Projekt_lab3", "Analysis Model", and "Biblioteka", and elements such as "Dodaj_tytul", "Klasy_warstwy_biznesowej", "Opis 'swiata rzeczywistego'", "PU_funkcje_bibiloteki", "addtytul_książki", "equals_Tytul_książki", "toString", "addtytul_książki", "Dodaj_tytul", "equals_Tytul_książki", "Fasada", "toString", "Tytul_książki", "ArrayList", "Biblioteka", "Dodaj_książke", "Dodaj_tytul_książki", "Sprawdz_czy_jest_książka", "Sprawdz_czy_jest_tytul", "Bibliotekarz", and "String". The main workspace shows the "Sprawdz_czy_jest_książka" details form. The "Name" field is filled with "Sprawdz_czy_jest_książka". The "Info" tab is selected, showing fields for Rank (Medium), Justification, Primary Actors, and Supporting Actors. The Documentation field contains a rich text editor with the following text: "Scenariusz", "1. Szukanie książki przebiega według atrybutu: numer egzemplarza (obowiązkowo) zgodnie z danymi tytułu podanego do przypadku użycia. Przeszukiwane są egzemplarze należące do konkretnego tytułu", "2. Jeśli istnieje egzemplarz o podanym numerze, zwracany jest egzemplarz z zasobów wypożyczalni, w przeciwnym wypadku zwracana jest informacja o braku egzemplarza." The bottom of the form has checkboxes for Abstract, Leaf, and Root.

Następnie należy wypełnić formularz wybrany z zakładki *Details*

The screenshot displays the Visual Paradigm for UML interface. The main window shows the 'Sprawdz_czy_jest_książka' use case details in the 'Details' tab. The 'Name' field is filled with 'Sprawdz_czy_jest_książka'. The 'Level' is set to 'Subfunction', 'Complexity' to 'Medium', 'Use Case Status' to 'Complete', and 'Implementation Status' to 'Partially Complete'. A precondition is defined as 'Dodaj książkę'. The 'Post-conditions' field contains the text: 'Zwraca wynik, określający, czy podany numer jest unikatowy lub podaje informację, że dany numer już istnieje'. The 'Author' is 'Zofia Ktucziewicz'. A 'Select Use Case' dialog is open in the foreground, showing a table of use cases. An arrow points from the 'Insert Use Case...' button in the main window to the dialog.

ID	Name	Primary actors	Supporting actors
8	Sprawdz_czy_jest_tytul		
12	Sprawdz_czy_jest_książka		
7	Dodaj_tytul_książki	Bibliotekarz	
11	Dodaj_książkę	Bibliotekarz	

Należy zmodyfikować warunki wstępne (*Preconditions*) w Zakładce *Details* przypadku użycia (PU) [Sprawdz_czy_jest_tytul](#) – w tym celu należy po kliknięciu prawym klawiszem na PU wybrać z listy pozycję *Open Use Case Details...*



W zakładce Details należy dodać do pola *Preconditions* możliwość wywołania PU *Sprawdz_czy_jest_książka* z PU *Dodaj_książke*

The screenshot shows the Visual Paradigm for UML interface. The main window displays the details of a use case named "Sprawdz czy jest tytuł". The "Preconditions" field contains two entries: "Dodaj tytuł książki" and "Dodaj książkę". A "Select Use Case" dialog is open, showing a table of use cases. The table has columns for ID, Name, Primary actors, and Supporting actors. The use case "Dodaj książkę" is selected.

ID	Name	Primary actors	Supporting actors
8	Sprawdz czy jest tytuł		
12	Sprawdz czy jest książka		
7	Dodaj tytuł książki	Bibliotekarz	
11	Dodaj książkę	Bibliotekarz	

Wykonać diagram sekwencji reprezentujący metodę `dodaj_książke` w klasie `Fasada` (po kliknięciu prawym klawiszem na *Analysis Model* wybrać *Diagram/UML Diagrams/Sequence Diagram*)

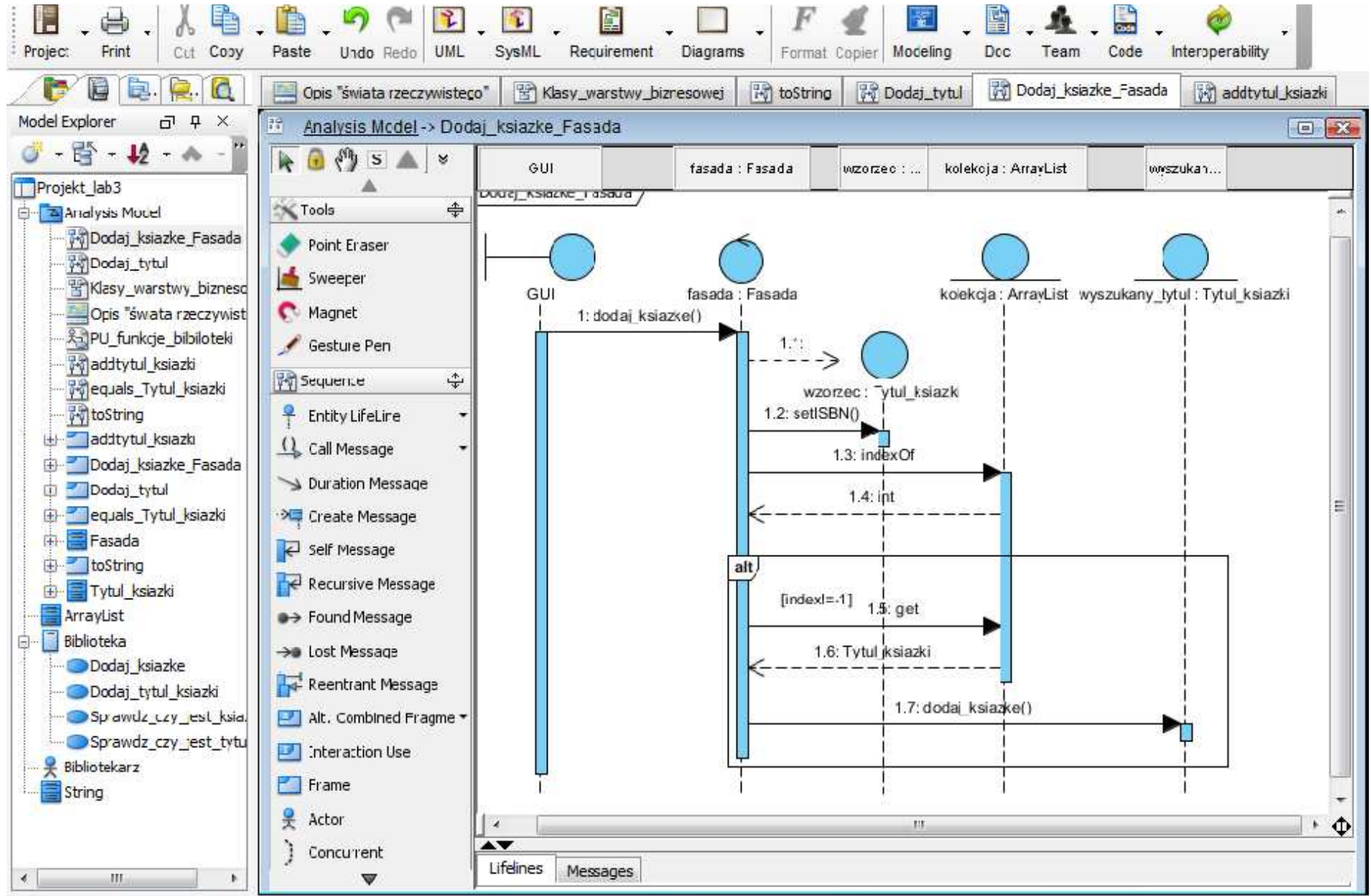
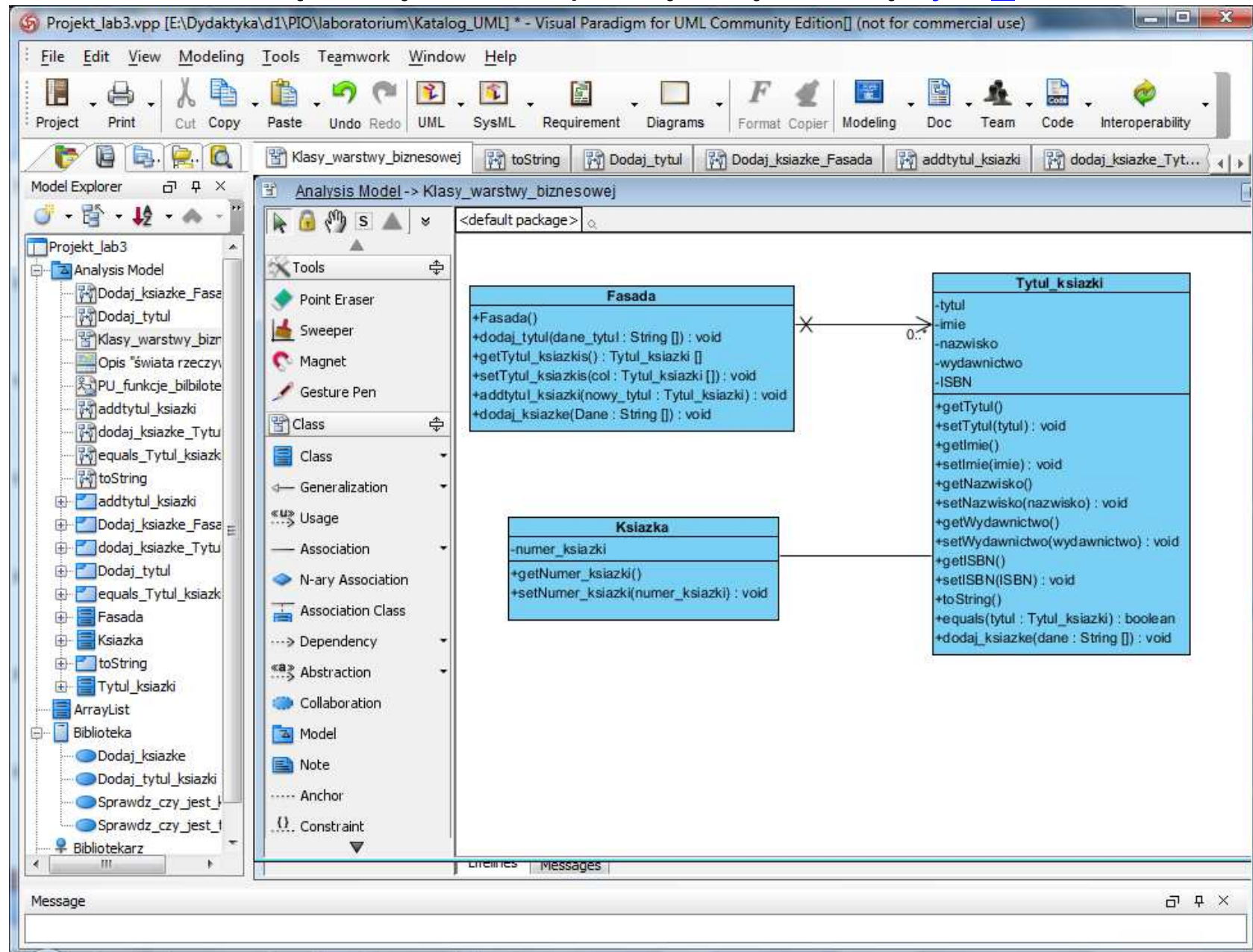
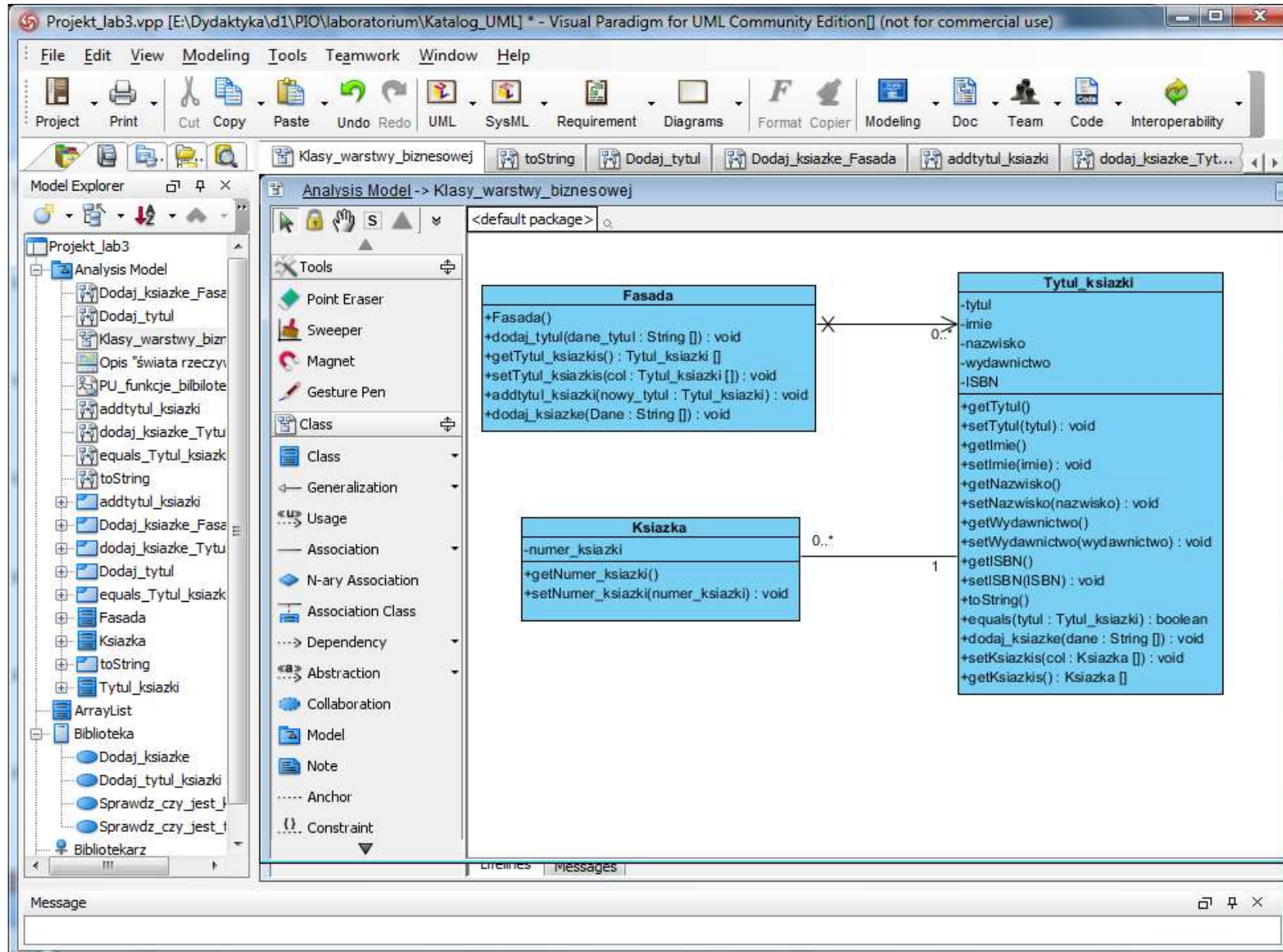


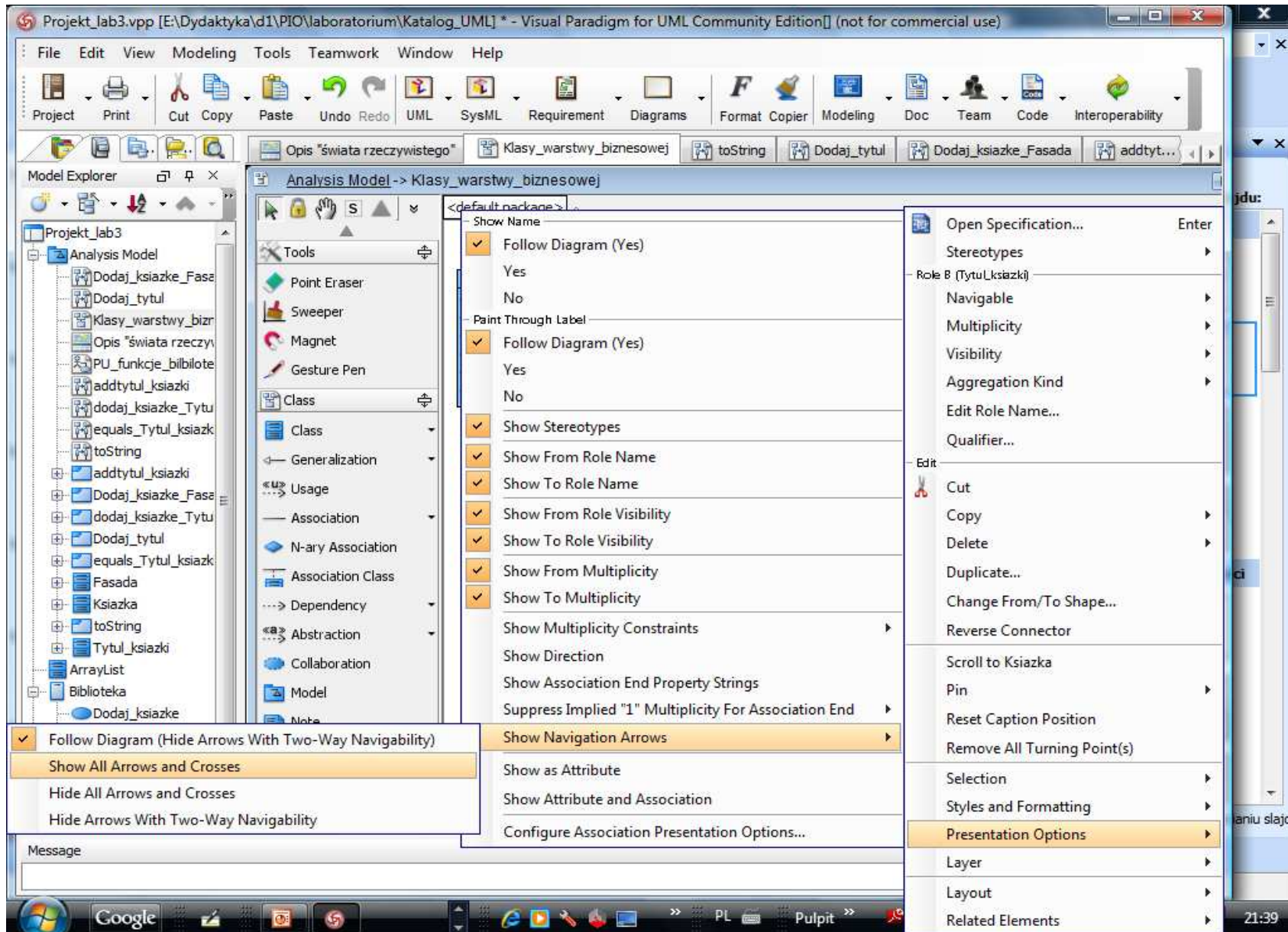
Diagram klas z nową metodą `dodaj_książke` w klasie `Fasada` oraz dodaną klasą `Książka` powiązaną z klasą `Tytul_książki`



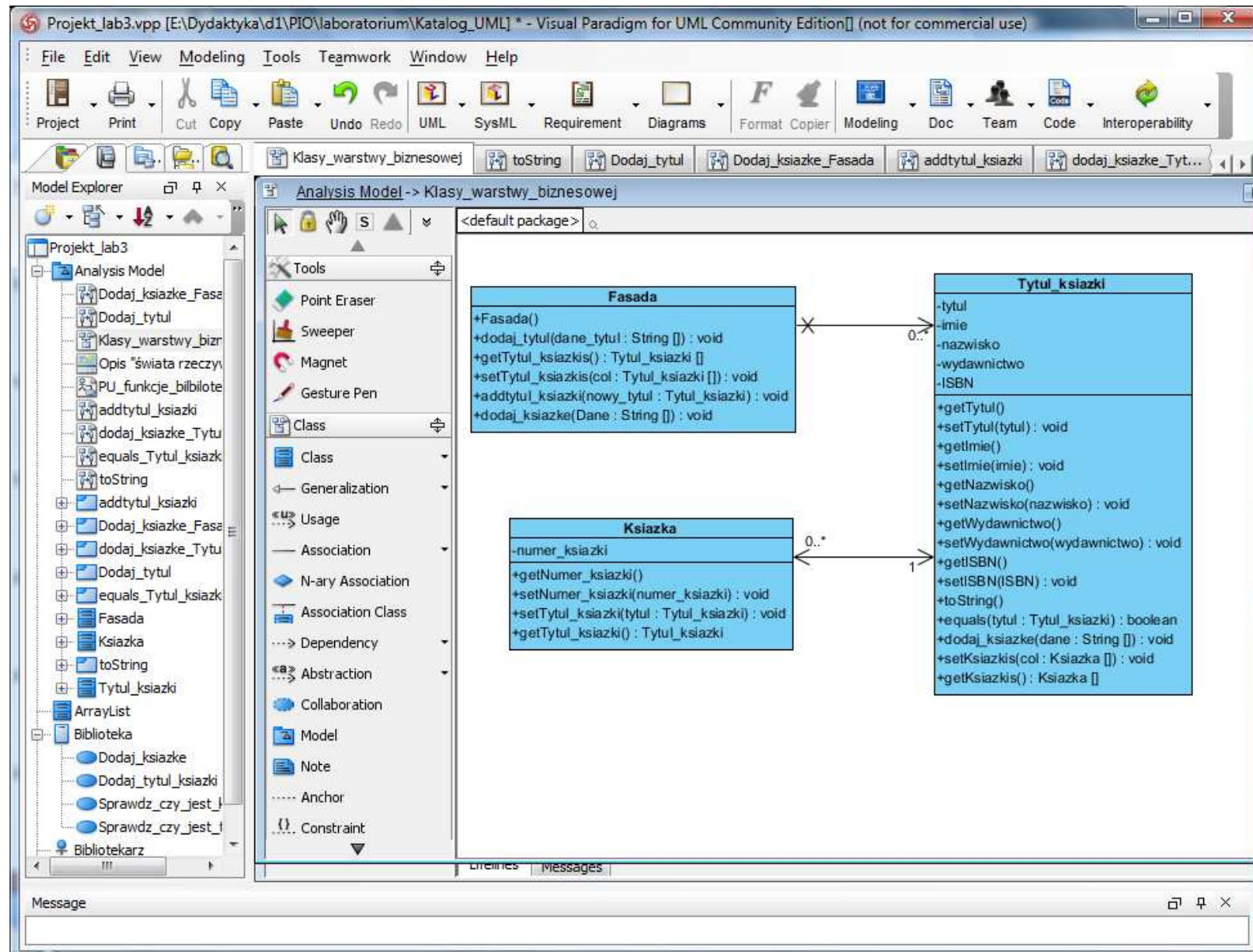
Ustawienie właściwości *Multiplicity* związku **Tytul_książki:Książka** ma 1:0..*. Dodanie dwóch metod typu **setKsiążkis** i **getKsiążkis** dla atrybutu typu kolekcja, który implementuje dwukierunkowy związek 1:0..* po stronie klasy typu **Tytul_książki**



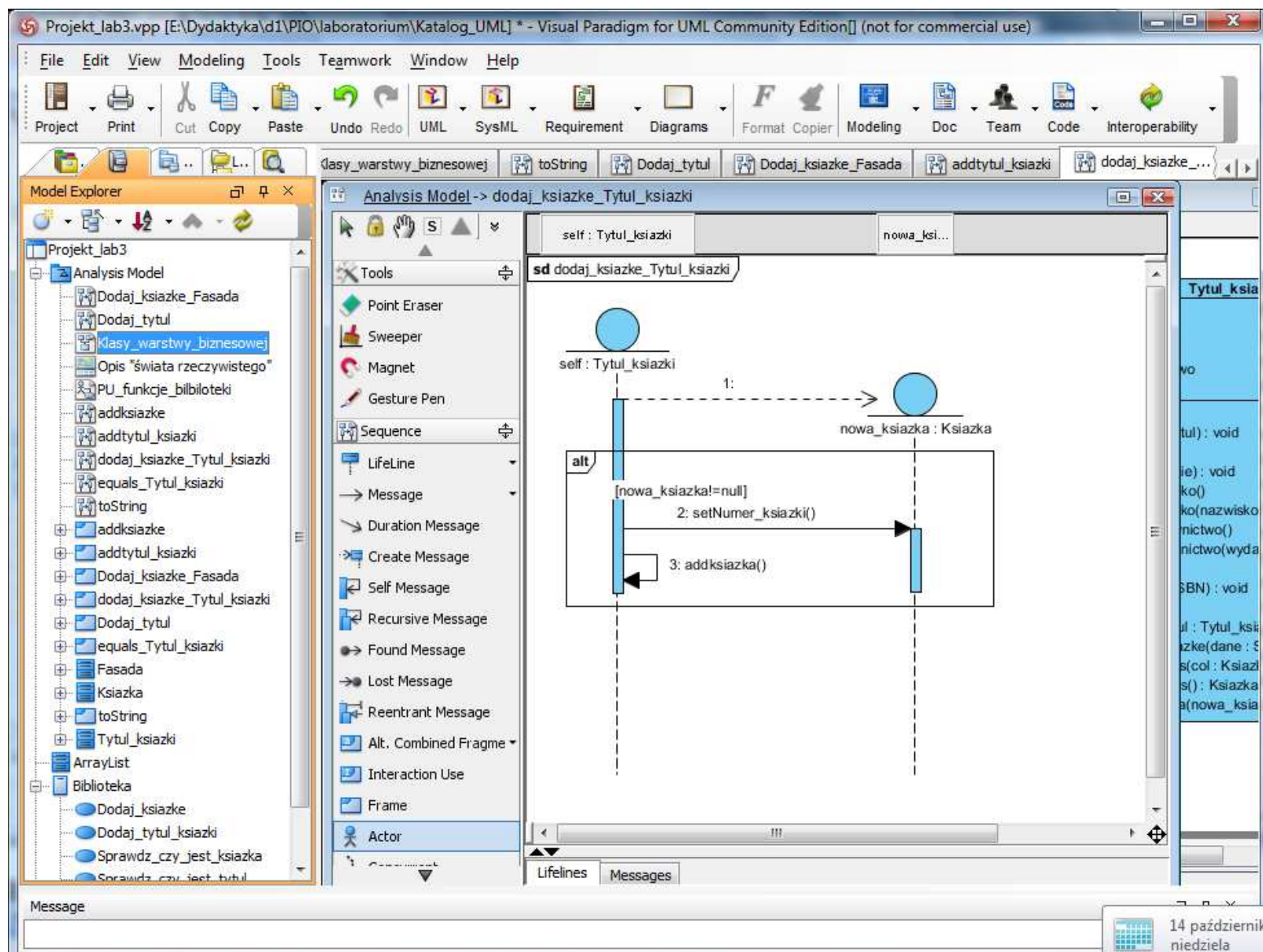
Uwidocznienie związku dwukierunkowego: obiekty typu **Tytul_książki** posiadają wiele obiektów typu **Książka** i każda obiekt btypu **Książka** posiada jeden obiekt typu **Tytul_książki**



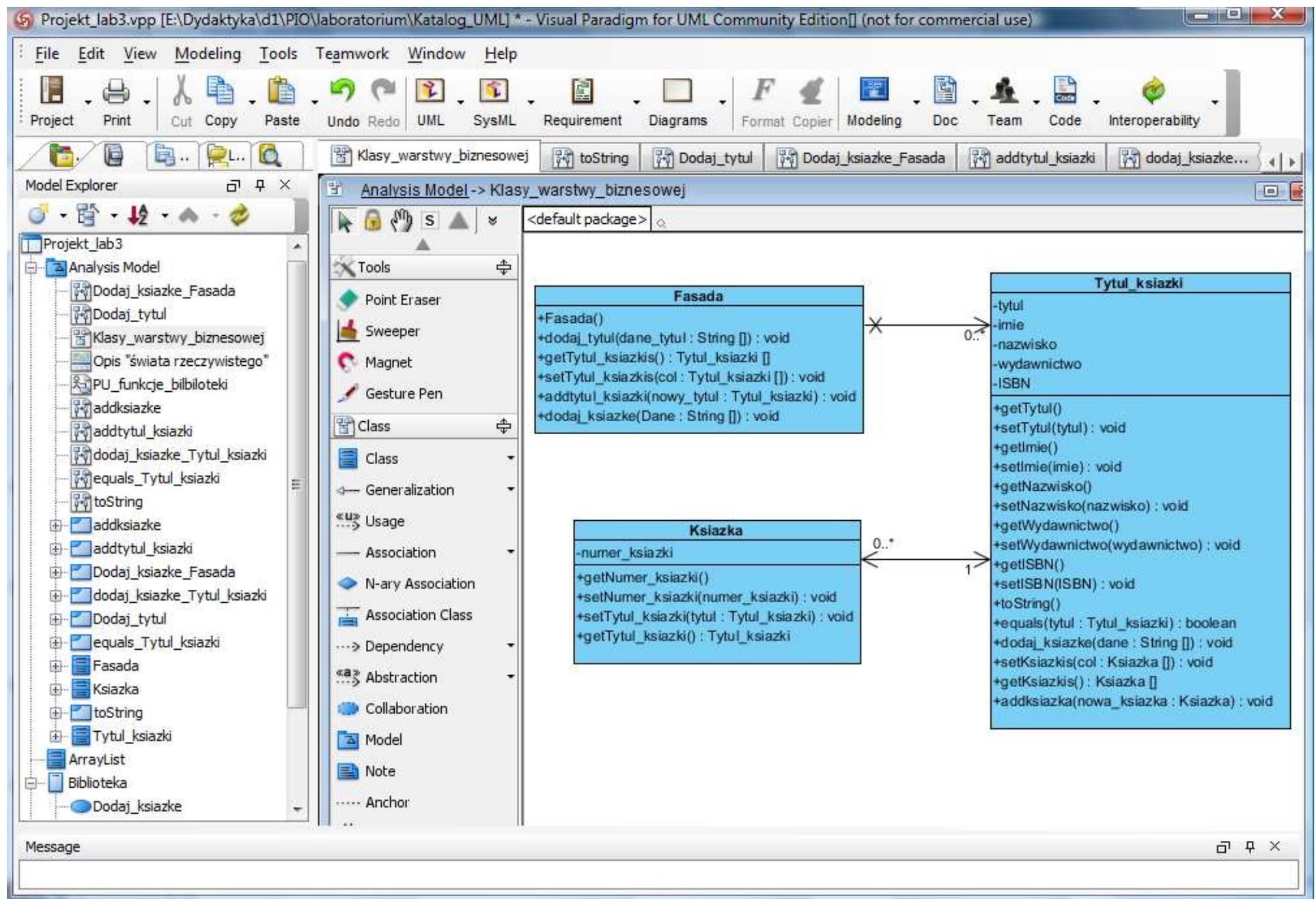
Efekt uwidocznienia związku dwukierunkowego **Tytul_książki: Książka**. Dodanie dwóch metod typu **setTytul_książki** i **getTytul_książki** dla atrybutu typu **Tytul_książki**, który implementuje dwukierunkowy związek 1:0..* po stronie klasy typu **Książka**



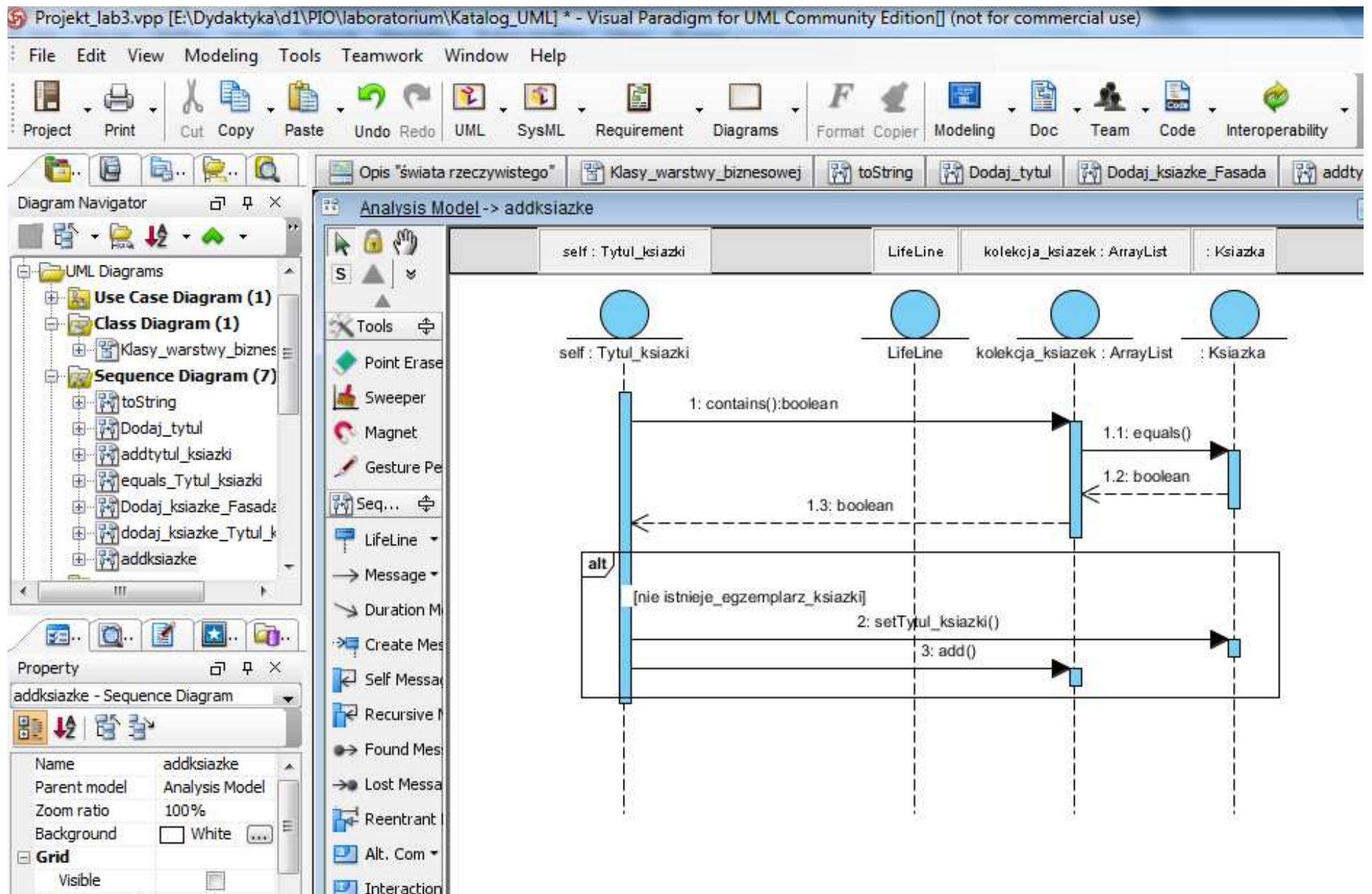
Dodanie nowego diagramu sekwencji reprezentującego metodę `dodaj_książke` w klasie `Tytul_książki` wywołaną w metodzie `dodaj_książke` klasy `Fasada` (po kliknięciu prawym klawiszem na *Analysis Model* wybrać *Diagram/UML Diagrams/Sequence Diagram*)



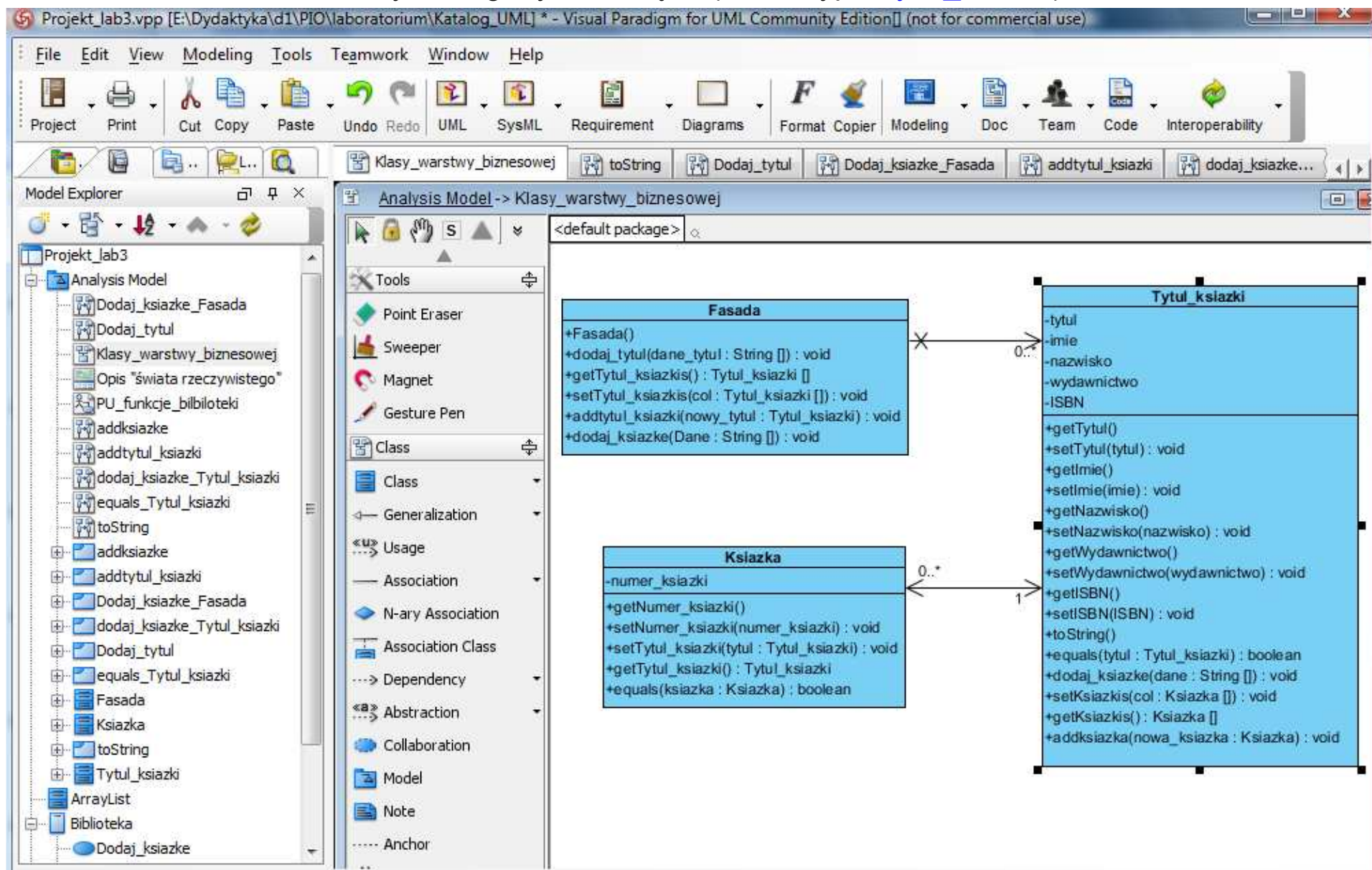
Klasa Tytul_książki ma nowe metody: dodaj_książke oraz addksiążka



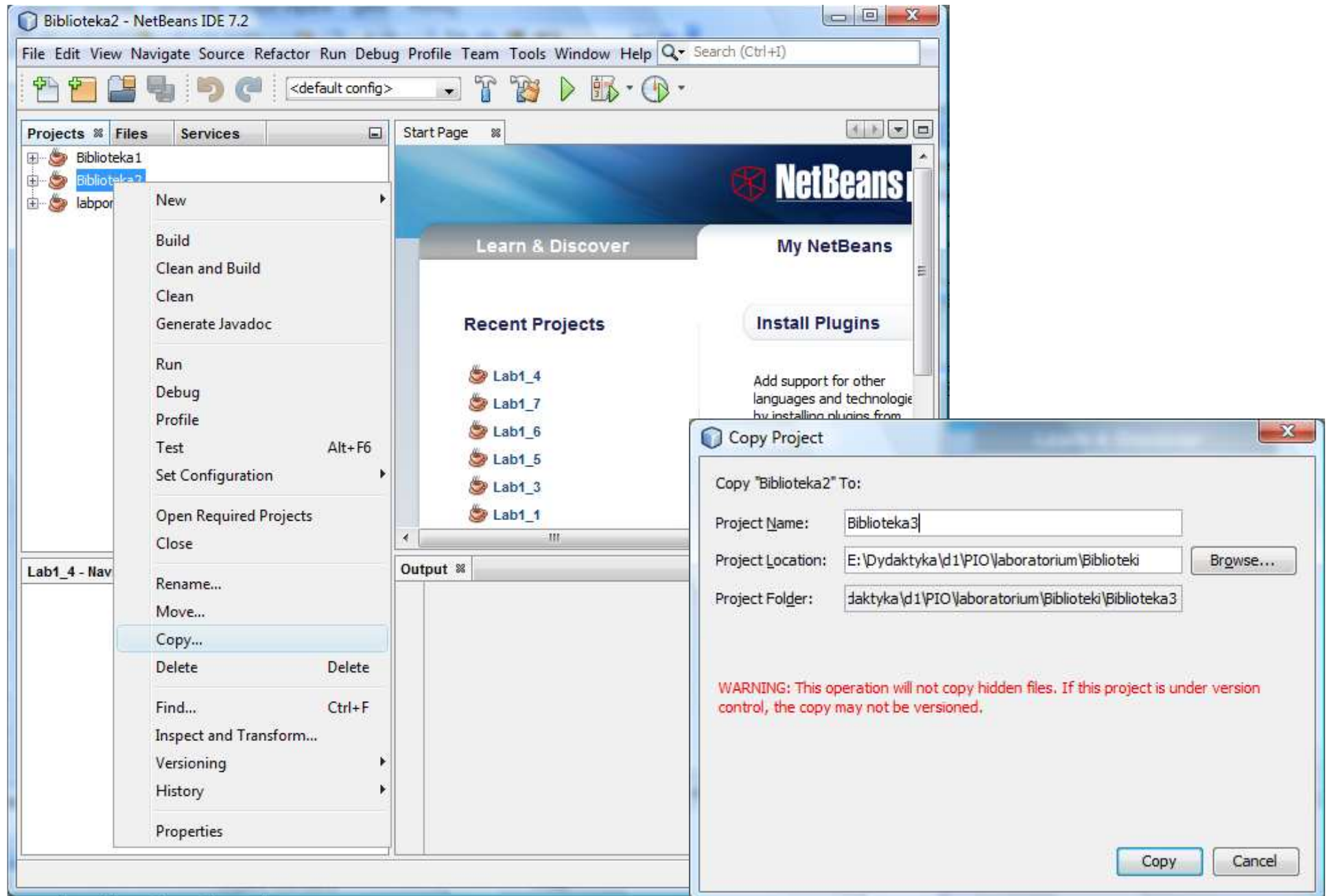
Dodanie nowego diagramu sekwencji reprezentującego metodę `addksiazka` w klasie `Tytul_książki` wywołaną w metodzie `dodaj_książke` klasy `Tytul_książki` (po kliknięciu prawym klawiszem na *Analysis Model* wybrać *Diagram/UML Diagrams/Sequence Diagram*)



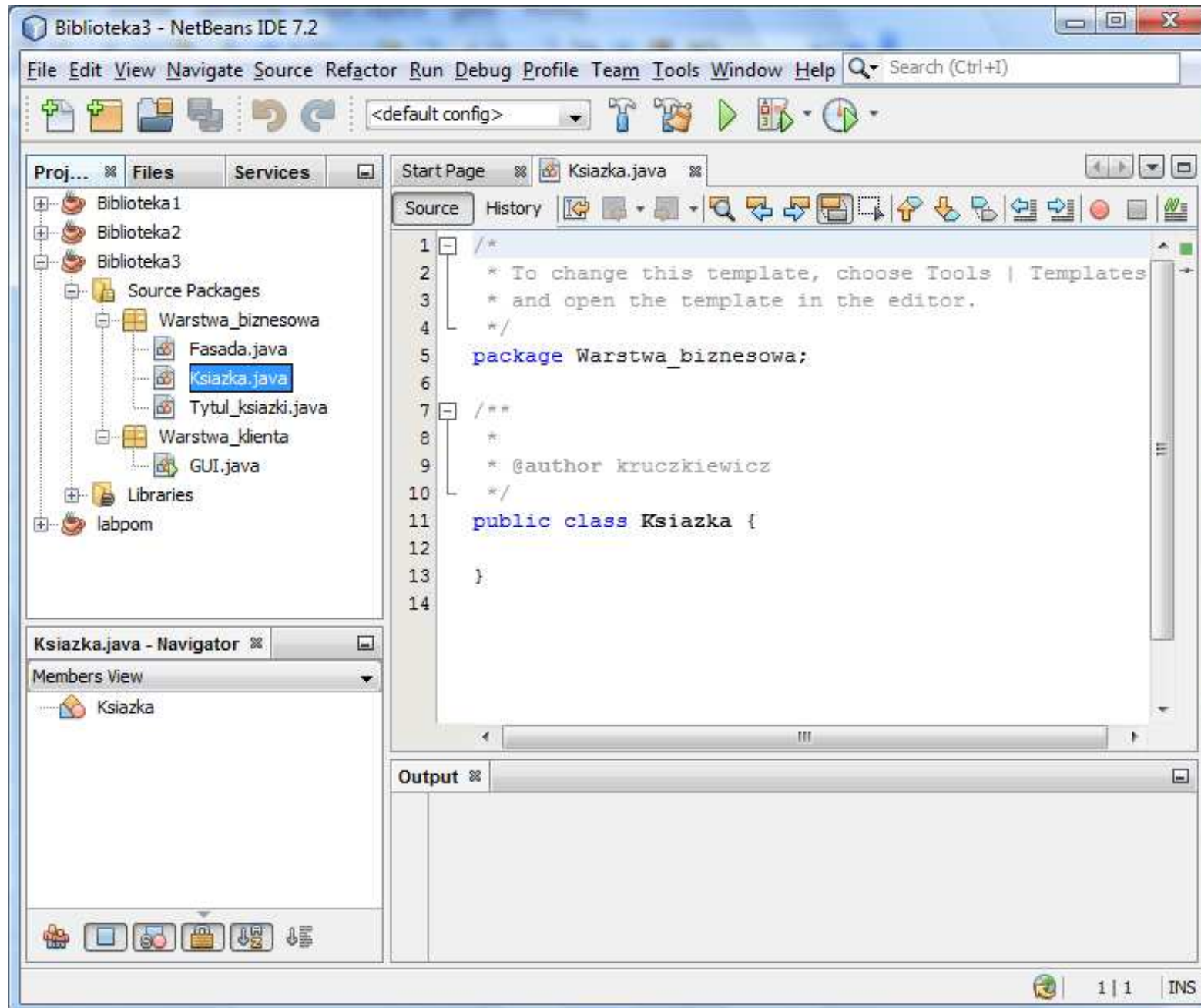
Po wykonaniu diagramu sekwencji metody `addksiazka` w klasie `Tytul_ksiazki` dodano metodę `equals` do klasy `Ksiazka`. Metoda ta jest użyta w metodzie `contains` klasy `ArrayList` w celu sprawdzenia czy nowa książka (obiekt typu `Ksiazka`) ma unikalny numer wśród książek należących do wybranego tytułu książki (obiekt typu `Tytul_ksiazki`)



Tworzenie kopii programu z etapu 1 (instrukcja do laboratorium 2)



Dodanie nowej klasy `Ksiazka` do pakietu `Warstwa_biznesowa`



Kod
klasy
Ksiazka

The screenshot shows an IDE window with the following components:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. Search (Ctrl+I).
- Toolbar:** Includes icons for file operations, undo, redo, and execution.
- Project Explorer (Left):** Shows a project structure with folders 'Biblioteka1', 'Biblioteka2', 'Biblioteka3', 'Source Packages', and 'Libraries'. Under 'Source Packages', there are sub-packages 'Warstwa_biznesowa' (containing 'Fasada.java', 'Ksiazka.java', 'Tytul_ksiazki.java') and 'Warstwa_klienta' (containing 'GUI.java').
- Members View (Bottom Left):** Shows the class 'Ksiazka' with members: 'Ksiazka()', 'equals(Object ob) : boolean', 'getNum() : int', 'getTytul_ksiazki() : Tytul_ksiazki', 'setNumer(int _numer)', 'setTytul_ksiazki(Tytul_ksiazki)', 'toString() : String', and 'mTytul_ksiazki : Tytul_ksiazki'.
- Source Editor (Right):** Displays the source code for 'Ksiazka.java'. The code is as follows:

```
1  /*...*/
5  package Warstwa_biznesowa;
6  /**...*/
10 public class Ksiazka {
11     private int numer;
12     private Tytul_ksiazki mTytul_ksiazki;
13
14     public Ksiazka() {
15     }
16
17     public int getNumer() {
18         return numer;
19     }
20
21     public void setNumer(int _numer) {
22         numer = _numer;
23     }
24
25     public Tytul_ksiazki getTytul_ksiazki() {
26         return mTytul_ksiazki;
27     }
28
29     public void setTytul_ksiazki(Tytul_ksiazki val)
30         mTytul_ksiazki = val;
31
32     public boolean equals(Object ob) // your code here
33     {
34         return numer == ((Ksiazka) ob).getNum();
35     }
36
37     public String toString() // your code here
38     {
39         String pom = mTytul_ksiazki.toString();
40         pom += " Numer: " + getNumer();
41         return pom;
42     }
43 }
```


Kod klasy `Ksiazka`

```
package Warstwa_biznesowa;

public class Ksiazka
{   private int numer;
    private Tytul_ksiazki mTytul_ksiazki;

    public Ksiazka() { }

    public int getNumer()                { return numer;}

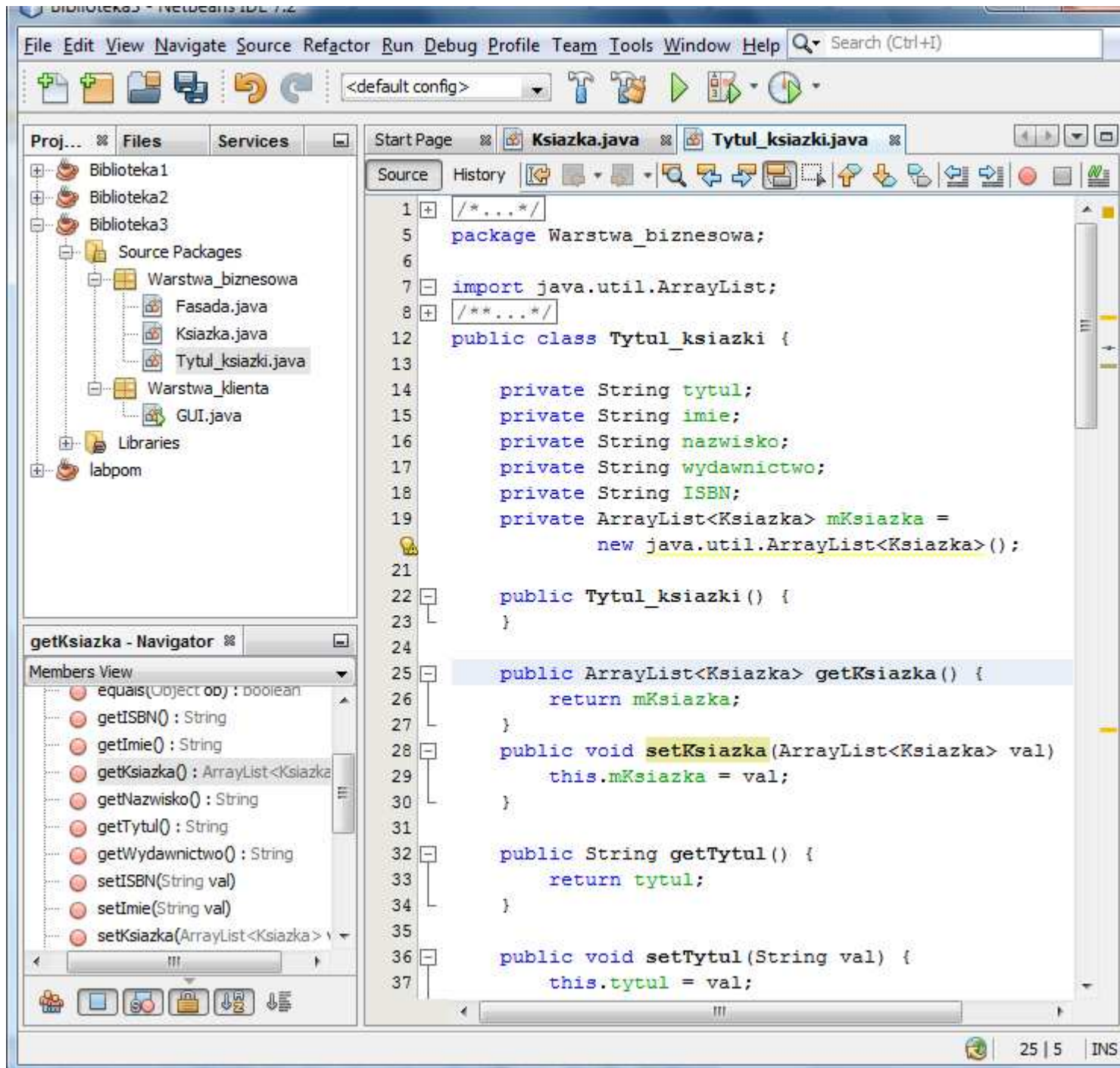
    public void setNumer(int _numer)     { numer = _numer; }

    public Tytul_ksiazki getTytul_ksiazki() { return mTytul_ksiazki;}

    public void setTytul_ksiazki(Tytul_ksiazki val) { mTytul_ksiazki = val; }

    public boolean equals(Object ob)
    { return numer==((Ksiazka)ob).getNumer(); }

    public String toString()
    { String pom= mTytul_ksiazki.toString();
      pom+=" Numer: "+getNumer();
      return pom;
    }
}
```



Uzupełniony
kod klasy
`Tytul_książki`

Uzupełniony kod klasy Tytul_książki

The screenshot displays the NetBeans IDE 7.2 interface for a project named 'Biblioteka3'. The main editor window shows the source code for the 'Tytul_książki.java' file. The code implements the 'equals()' method, which compares the ISBN of the current object with the ISBN of the provided object. It also includes a 'dodaj_książke()' method that creates a new 'Książka' object and adds it to the 'mKsiążka' list, and an 'addKsiążka()' method that adds a 'Książka' object to the list if it is not already present.

```
81  @Override
82  public boolean equals(Object ob) //your code here
83  {
84      String isbn1 = getISBN();
85      String isbn2 = ((Tytul_książki) ob).getISBN();
86      boolean a = isbn1.equals(isbn2);
87      return a;
88  }
89
90
91  public void dodaj_książke(String dane[]) // your code
92  {
93      Książka nowa= new Książka();
94      if (nowa != null) {
95          nowa.setNumer(Integer.parseInt(dane[1]));
96          addKsiążka(nowa);
97      }
98  }
99
100 public void addKsiążka(Książka nowa) {
101     if (!this.mKsiążka.contains(nowa)) {
102         this.mKsiążka.add(nowa);
103         nowa.setTytul_książki(this);
104     }
105 }
106 }
107
```

The left sidebar shows the project structure with 'Biblioteka3' expanded to show 'Source Packages' containing 'Warstwa_biznesowa' (with 'Fasada.java', 'Książka.java', and 'Tytul_książki.java') and 'Warstwa_klienta' (with 'GUI.java'). The 'Members View' for 'Tytul_książki' lists methods: 'Tytul_książki()', 'addKsiążka(Książka nowa)', 'dodaj_książke(String[] dane)', 'equals(Object ob) : boolean', 'getISBN() : String', 'getImie() : String', 'getKsiążka() : ArrayList<Książka>', and 'getNazwisko() : String'.

The status bar at the bottom right shows '107 | 1 | INS'.

Kod klasy Tytul_ksiazki

```
package Warstwa_biznesowa;

import java.util.ArrayList;

public class Tytul_ksiazki {

    private String tytul;
    private String imie;
    private String nazwisko;
    private String wydawnictwo;
    private String ISBN;
    private ArrayList<Ksiazka> mKsiazka = new java.util.ArrayList<Ksiazka>();

    public Tytul_ksiazki() { }

    public ArrayList<Ksiazka> getKsiazka() {
        return mKsiazka;
    }
    public void setKsiazka(ArrayList<Ksiazka> val) {
        this.mKsiazka = val;
    }
    public String getTytul() {
        return tytul;
    }
    public void setTytul(String val) {
        this.tytul = val;
    }
}
```

```
public String getImie() {
    return imie;
}
public void setImie(String val) {
    this.imie = val;
}
public String getNazwisko() {
    return nazwisko;
}
public void setNazwisko(String val) {
    this.nazwisko = val;
}
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;
}
```

@Override

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

@Override

```
public boolean equals(Object ob)
{   String isbn1 = getISBN();
    String isbn2 = ((Tytul_ksiazki) ob).getISBN();
    boolean a = isbn1.equals(isbn2);
    return a;
}
```

```
public void dodaj_ksiazke(String dane[])
{   Ksiazka nowa = new Ksiazka();
    if (nowa != null) {
        nowa.setNumer(Integer.parseInt(dane[1]));
        addKsiazka(nowa);
    }
}
```

```
public void addKsiazka(Ksiazka nowa) {
    if (!this.mKsiazka.contains(nowa)) {
        this.mKsiazka.add(nowa);
        nowa.setTytul_ksiazki(this);
    }
}
}
```


Uzupełniony kod klasy Fasada

The screenshot displays the NetBeans IDE 7.2 interface for a project named 'Biblioteka3'. The main editor window shows the code for the 'Fasada.java' class. The code is as follows:

```
1 package Warstwa_biznesowa;
2 /*...*/
6 import java.util.ArrayList;
7 /*...*/
11 public class Fasada {
12
13     private ArrayList<Tytul_książki> tytuly_książek =
14         new ArrayList<Tytul_książki>();
15
16     public Fasada() {
17     }
18
19     public ArrayList<Tytul_książki> getTytuly_książek() {
20         return tytuly_książek;
21     }
22
23     public void setTytuly_książek(ArrayList<Tytul_książki> val)
24         this.tytuly_książek = val;
25     }
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar with various icons, and a project explorer on the left showing the directory structure: Biblioteka1, Biblioteka2, Biblioteka3, Source Packages, Warstwa_biznesowa (containing Fasada.java, Książka.java, Tytul_książki.java), Warstwa_klienta, and GUI.java. A 'tytuly_książek - Navigator' window is also visible, showing a 'Members View' with methods like Dodaj_książke, addtytul_książki, dodaj_tytul, getTytuly_książek, and setTytuly_książek.

Uzupełniony kod klasy Fasada

The screenshot shows the NetBeans IDE 7.2 interface. The main editor window displays the code for the `Fasada` class. The code is as follows:

```
27 public void dodaj_tytul(String dane_tytul[]) {
28     Tytul_książki tytul_książki = new Tytul_książki();
29     tytul_książki.setTytul(dane_tytul[0]);
30     tytul_książki.setNazwisko(dane_tytul[1]);
31     tytul_książki.setImie(dane_tytul[2]);
32     tytul_książki.setISBN(dane_tytul[3]);
33     tytul_książki.setWydawnictwo(dane_tytul[4]);
34     addtytul_książki(tytul_książki);
35 }
36
37 public void addtytul_książki(Tytul_książki val) {
38     boolean czy_jest = tytuly_książek.contains(val);
39     if (!czy_jest) {
40         tytuly_książek.add(val);
41     }
42 }
43
44 public void dodaj_książke(String dane[]) // your code here
45 {
46     Tytul_książki pom = new Tytul_książki();
47     pom.setISBN(dane[0]);
48     int idx = tytuly_książek.indexOf(pom);
49     if (idx != -1) {
50         Tytul_książki pom1 = tytuly_książek.get(idx);
51         pom1.dodaj_książke(dane);
52         System.out.println(pom1.getKsiążka().toString());
53     } //linia tymczasowa
54 }
55 }
```

The left sidebar shows the project structure for `Biblioteka3`. The `Members View` for the `Fasada` class is visible, listing the following methods and attributes:

- `Fasada()`
- `addtytul_książki(Tytul_książki val)`
- `dodaj_książke(String[] dane)`
- `dodaj_tytul(String[] dane_tytul)`
- `getTytuly_książek(): ArrayList<Tytul_książki>`
- `setTytuly_książek(ArrayList<Tytul_książki> tytuly_książek)`
- `tytuly_książek: ArrayList<Tytul_książki>`

The status bar at the bottom right shows the page number 51, the slide number 36, and the text INS.

Kod klasy Fasada

```
package Warstwa_biznesowa;
```

```
import java.util.ArrayList;
```

```
public class Fasada {
```

```
    private ArrayList<Tytul_ksiazki> tytuly_ksiazek = new ArrayList<Tytul_ksiazki>();
```

```
    public Fasada() {  
    }
```

```
    public ArrayList<Tytul_ksiazki> getTytuly_ksiazek() {  
        return tytuly_ksiazek;  
    }
```

```
    public void setTytuly_ksiazek(ArrayList<Tytul_ksiazki> val) {  
        this.tytuly_ksiazek = val;  
    }
```



```
public void dodaj_tytul(String dane_tytul[]) {  
    Tytul_książki tytul_książki = new Tytul_książki();  
    tytul_książki.setTytul(dane_tytul[0]);  
    tytul_książki.setNazwisko(dane_tytul[1]);  
    tytul_książki.setImie(dane_tytul[2]);  
    tytul_książki.setISBN(dane_tytul[3]);  
    tytul_książki.setWydawnictwo(dane_tytul[4]);  
    addtytul_książki(tytul_książki);  
}
```

```
public void addtytul_książki(Tytul_książki val) {  
    boolean czy_jest = tytuly_książek.contains(val);  
    if (!czy_jest) {  
        tytuly_książek.add(val);  
    }  
}
```

```
public void dodaj_książke(String dane[]) /  
{  
    Tytul_książki pom = new Tytul_książki();  
    pom.setISBN(dane[0]);  
    int idx = tytuly_książek.indexOf(pom);  
    if (idx != -1) {  
        Tytul_książki pom1 = tytuly_książek.get(idx);  
        pom1.dodaj_książke(dane);  
        System.out.println(pom1.getKsiążka().toString());  
    } //linia tymczasowa  
}
```

Wynik działania programu – wstawianie tytułów

The screenshot displays the NetBeans IDE interface for a project named "Biblioteka3". The main editor window shows the source code for `Tytul_ksiazki.java`. The code defines a `dodawanie_tytulow()` method that adds two book titles to a list. The first title is "tytul1" by Jan Kowalski (ISBN: 12345, Publisher: W1), and the second is "tytul2" by Piotr Nowak (ISBN: 67891, Publisher: W2). The method uses `ap.dodaj_tytul()` to add the titles and `ap.getTytuly_ksiazek().toString()` to retrieve the list. The output is printed to the console and shown in a message dialog.

```
17  
18     public void dodawanie_tytulow() {  
19         String dane_tytulu1[] = {"tytul1", "Jan", "Kowalski", "12345", "W1"};  
20         String dane_tytulu2[] = {"tytul2", "Piotr", "Nowak", "67891", "W2"};  
21  
22         ap.dodaj_tytul(dane_tytulu1);  
23         ap.dodaj_tytul(dane_tytulu1);  
24         ap.dodaj_tytul(dane_tytulu2);  
25  
26         String lan = ap.getTytuly_ksiazek().toString();  
27  
28         System.out.println(lan);  
29         JOptionPane.showMessageDialog(null, lan);  
30     }
```

The Output window shows the following text:

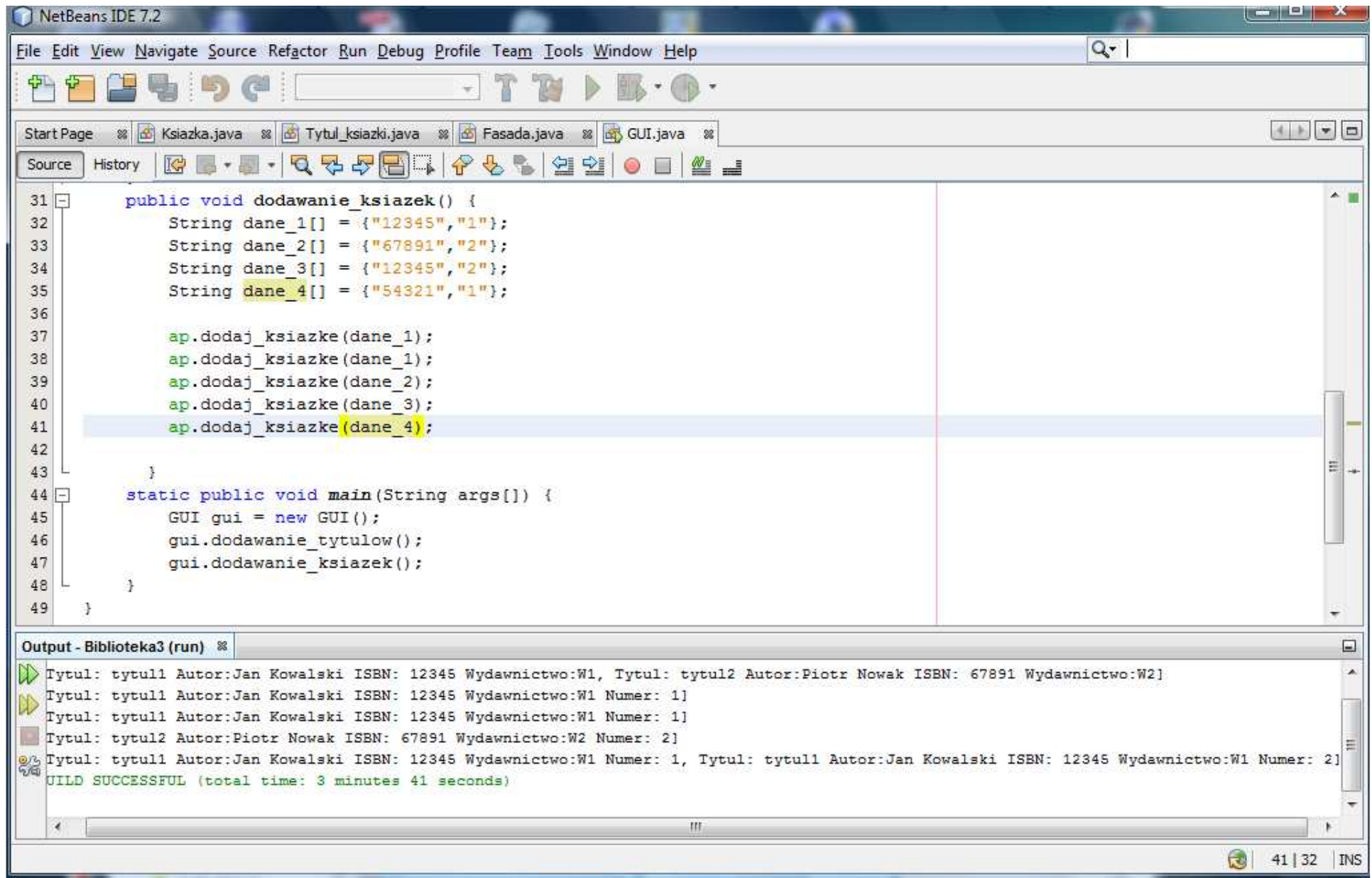
```
run:  
[Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1, Tytul: tytul2 Autor:Piotr Nowak ISBN: 67891 Wydawnictwo:W
```

A message dialog box is displayed with the following text:

```
Message  
[Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1, Tytul: tytul2 Autor:Piotr Nowak ISBN: 67891 Wydawnictwo:W2]
```

The dialog box has an "OK" button.

Wynik działania programu – wstawianie książek



The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows the source code for a Java application. The code defines a `dodawanie_ksiazek()` method that adds four books to a list, and a `main()` method that initializes a GUI and calls the `dodawanie_ksiazek()` method. The output window shows the results of the program's execution, including the titles, authors, ISBNs, and publisher information for the books added.

```
31 public void dodawanie_ksiazek() {
32     String dane_1[] = {"12345", "1"};
33     String dane_2[] = {"67891", "2"};
34     String dane_3[] = {"12345", "2"};
35     String dane_4[] = {"54321", "1"};
36
37     ap.dodaj_ksiazke(dane_1);
38     ap.dodaj_ksiazke(dane_1);
39     ap.dodaj_ksiazke(dane_2);
40     ap.dodaj_ksiazke(dane_3);
41     ap.dodaj_ksiazke(dane_4);
42
43 }
44 static public void main(String args[]) {
45     GUI gui = new GUI();
46     gui.dodawanie_tytulow();
47     gui.dodawanie_ksiazek();
48 }
49 }
```

Output - Biblioteka3 (run) %

```
Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1, Tytul: tytul2 Autor:Piotr Nowak ISBN: 67891 Wydawnictwo:W2]
Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1 Numer: 1]
Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1 Numer: 1]
Tytul: tytul2 Autor:Piotr Nowak ISBN: 67891 Wydawnictwo:W2 Numer: 2]
Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1 Numer: 1, Tytul: tytul1 Autor:Jan Kowalski ISBN: 12345 Wydawnictwo:W1 Numer: 2]
BUILD SUCCESSFUL (total time: 3 minutes 41 seconds)
```

41 | 32 | INS

Kod klasy GUI

```
package Warstwa_klienta;

import Warstwa_biznesowa.Fasada;
import javax.swing.JOptionPane;

public class GUI {

    Fasada ap = new Fasada();

    public void dodawanie_tytulow() {
        String dane_tytulu1[] = {"tytul1", "Jan", "Kowalski", "12345", "W1"};
        String dane_tytulu2[] = {"tytul2", "Piotr", "Nowak", "67891", "W2"};

        ap.dodaj_tytul(dane_tytulu1);
        ap.dodaj_tytul(dane_tytulu1);
        ap.dodaj_tytul(dane_tytulu2);

        String lan = ap.getTytuly_ksiazek().toString();

        System.out.println(lan);
        JOptionPane.showMessageDialog(null, lan);
    }
}
```

```
public void dodawanie_ksiazek() {  
    String dane_1[] = {"12345","1"};  
    String dane_2[] = {"67891","2"};  
    String dane_3[] = {"12345","2"};  
    String dane_4[] = {"54321","1"};  
  
    ap.dodaj_ksiazke(dane_1);  
    ap.dodaj_ksiazke(dane_1);  
    ap.dodaj_ksiazke(dane_2);  
    ap.dodaj_ksiazke(dane_3);  
    ap.dodaj_ksiazke(dane_4);  
  
}  
  
static public void main(String args[]) {  
    GUI gui = new GUI();  
    gui.dodawanie_tytulow();  
    gui.dodawanie_ksiazek();  
}  
}
```