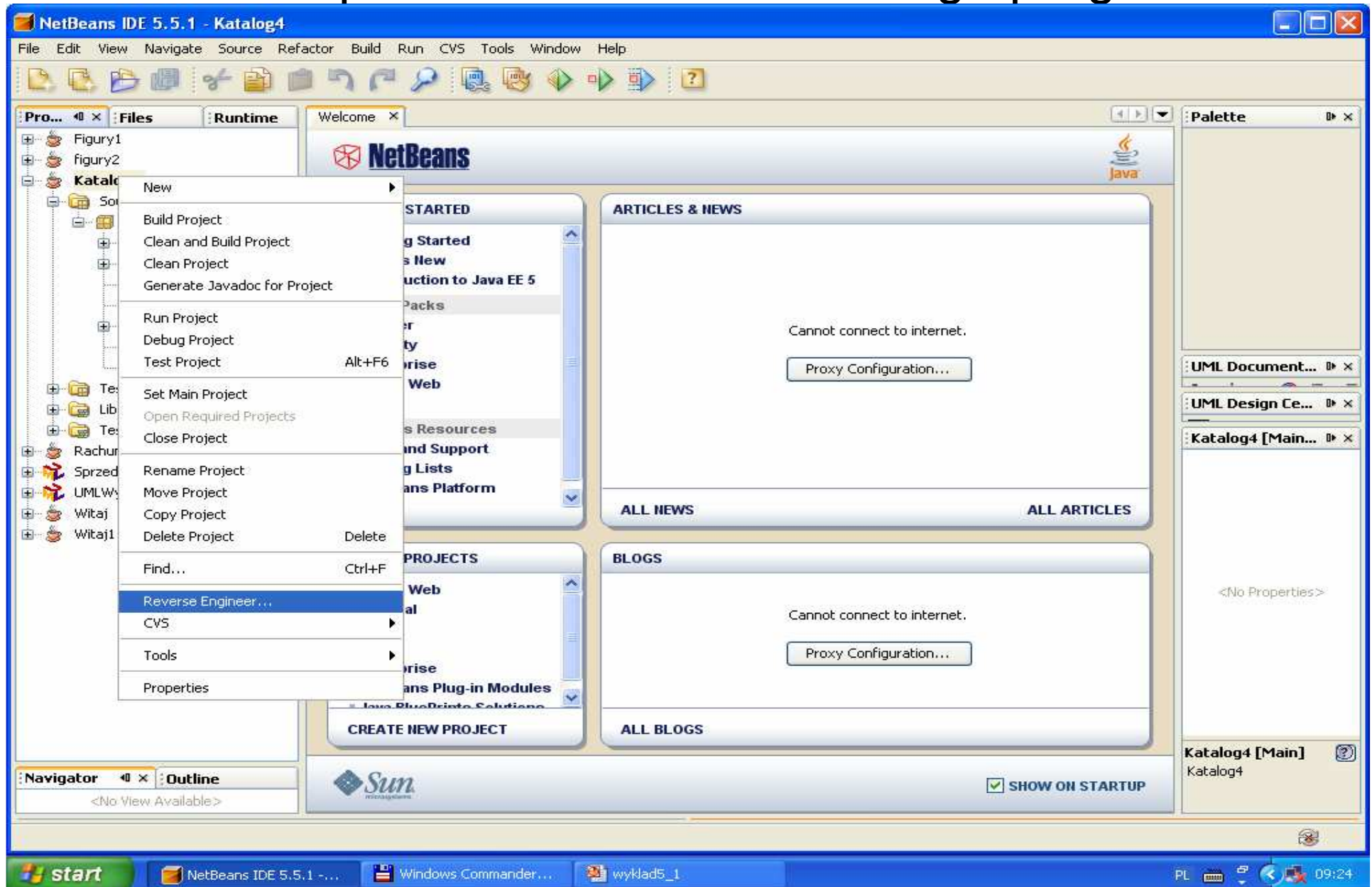


Inżynieria odwrotna – tworzenie diagramów klas i sekwencji na podstawie kodu źródłowego

Podstawy inżynierii oprogramowania

Zofia Kruczkiewicz

Wybór inżynierii odwrotnej – tworzenie diagramów UML na podstawie kodu źródłowego programu



Reverse Engineer



Selected Nodes: Katalog4

Java Project Source Package Folders

Reverse Engineer	Package Folder	Package Folder Label
<input checked="" type="checkbox"/>	src	Source Packages
<input checked="" type="checkbox"/>	test	Test Packages

Use Existing UML Project

Target Project: SprzedazUML

Create New UML Project

Project Name: Katalog4-Model

Project Location: C:\Settings\dydaktyka\PIO\wk5

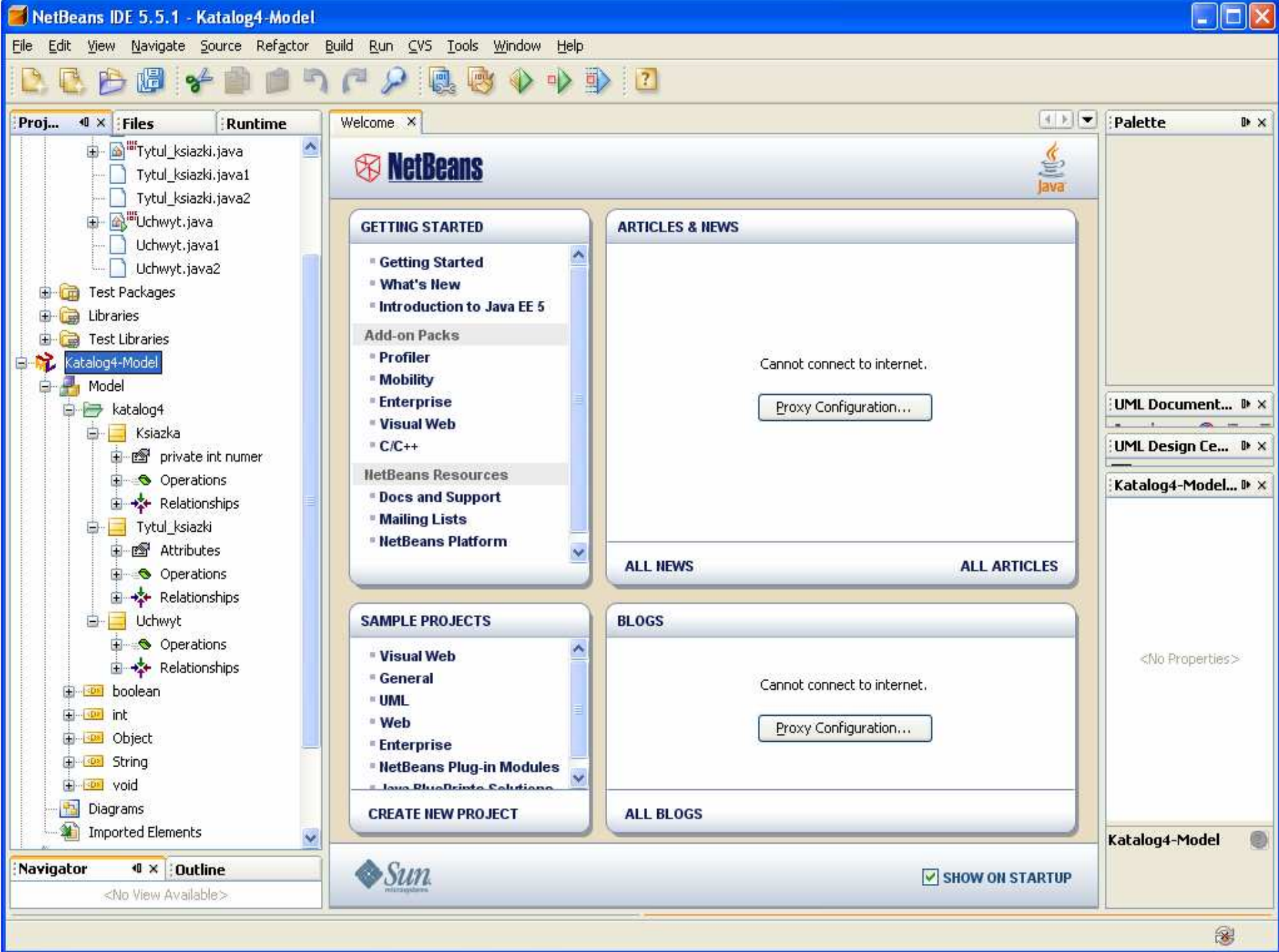
Browse...

Project Folder: C:\Settings\dydaktyka\PIO\wk5\Katalog4-Model

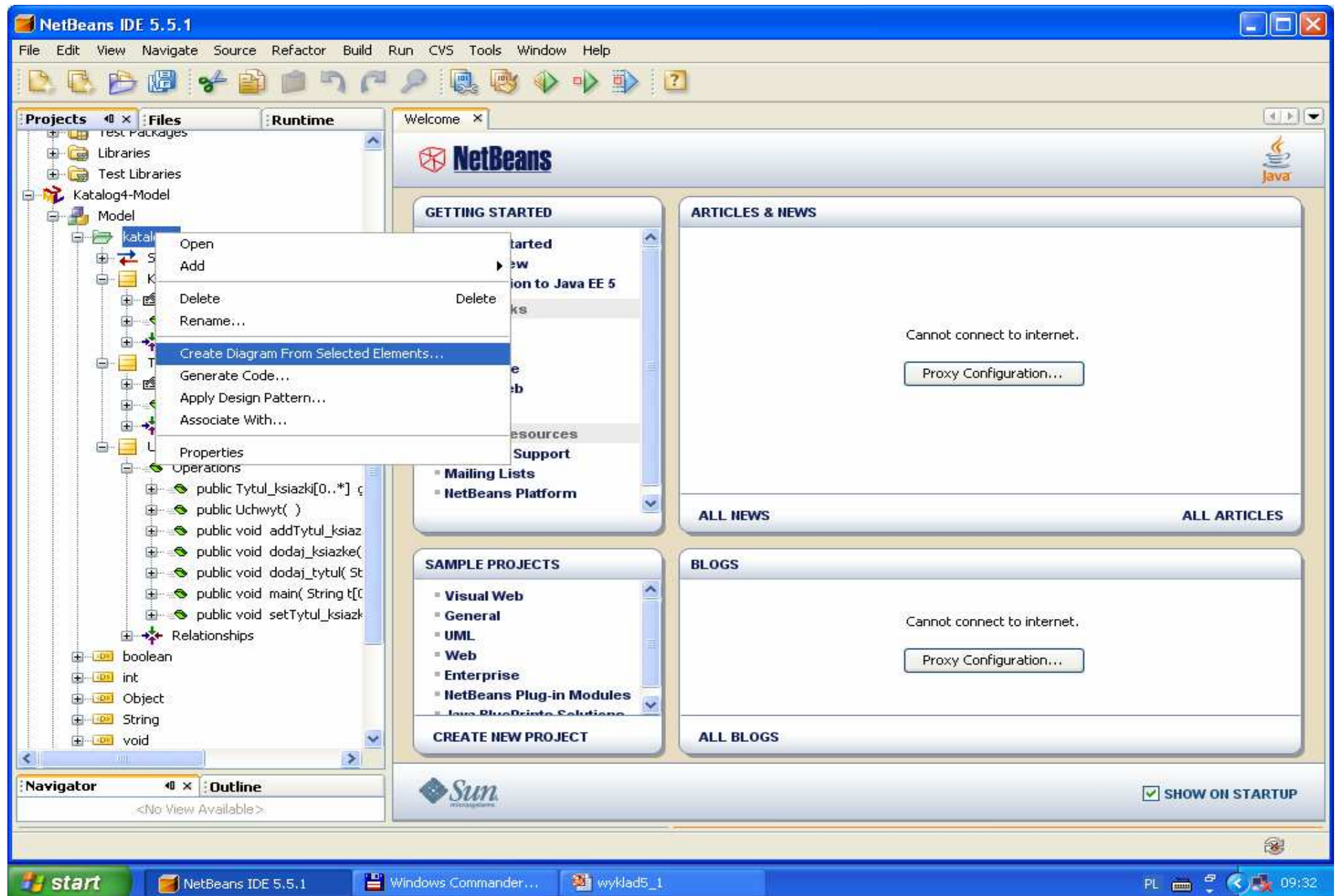
OK

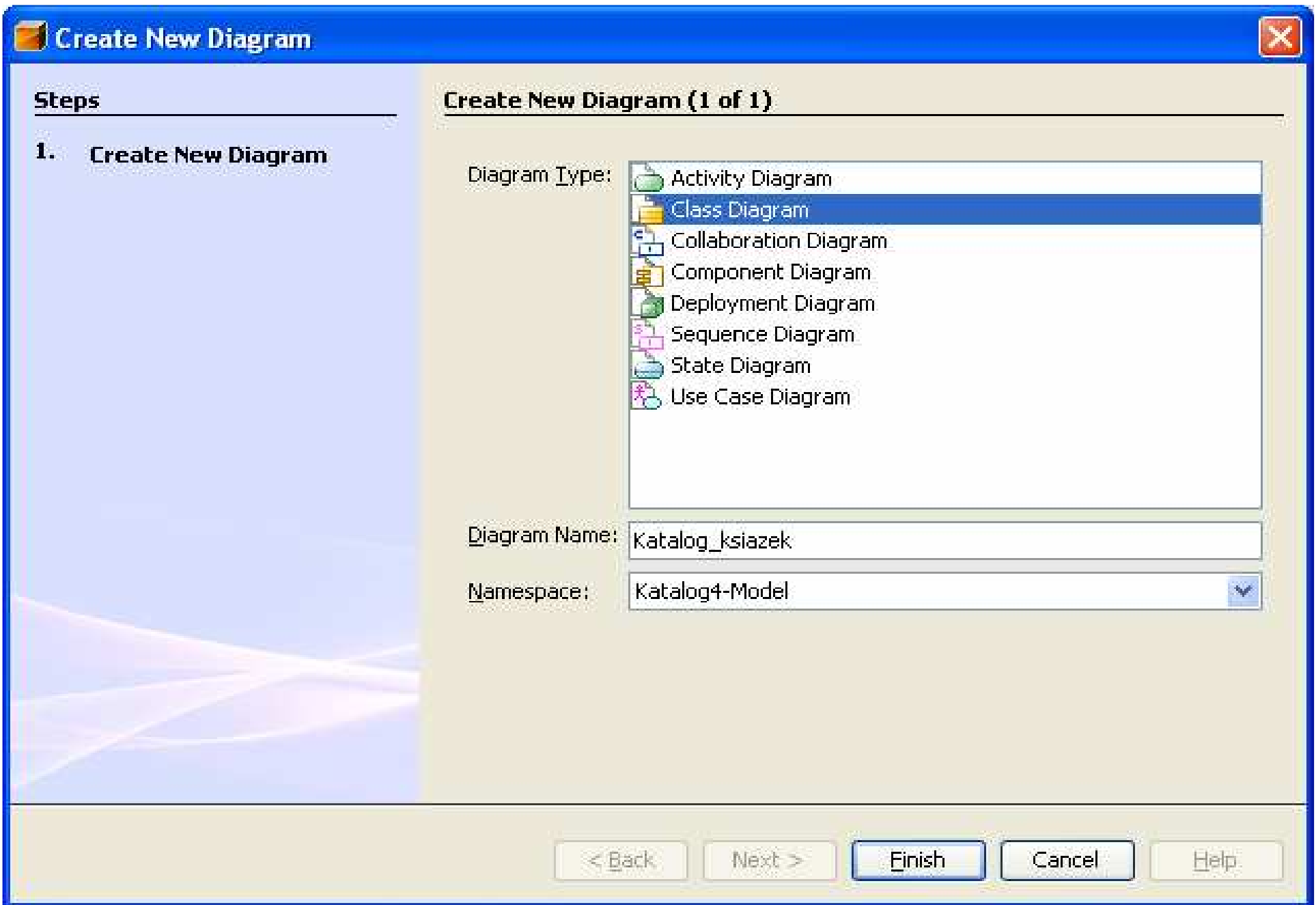
Cancel

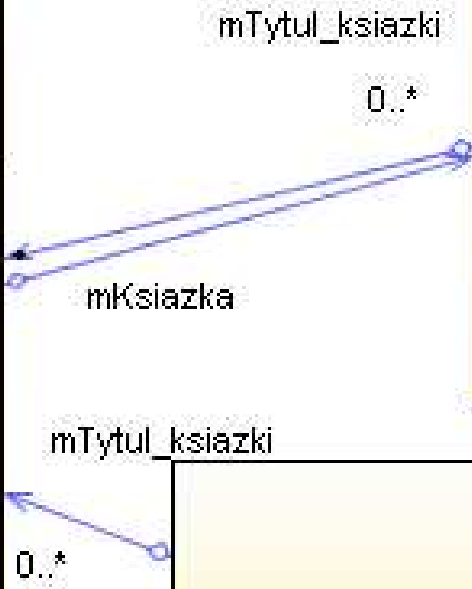
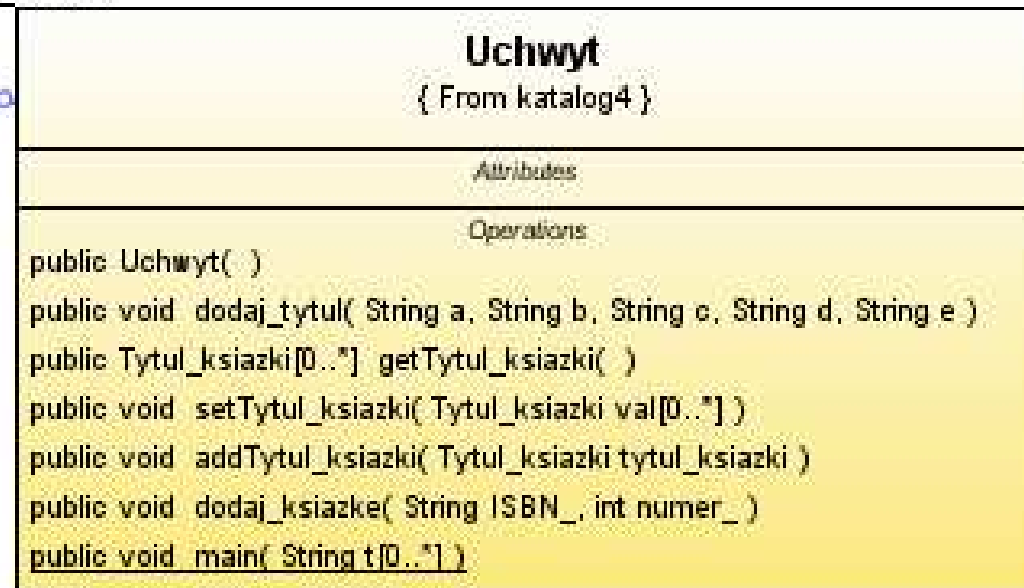
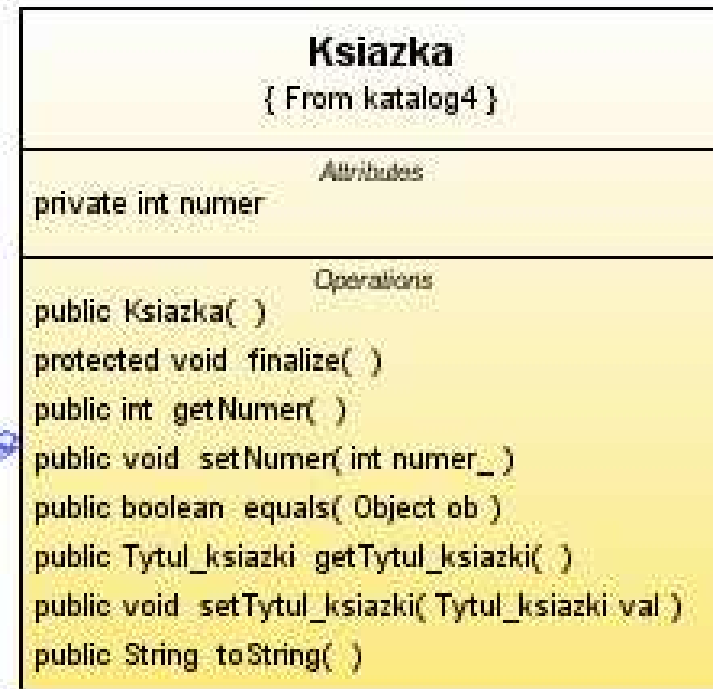
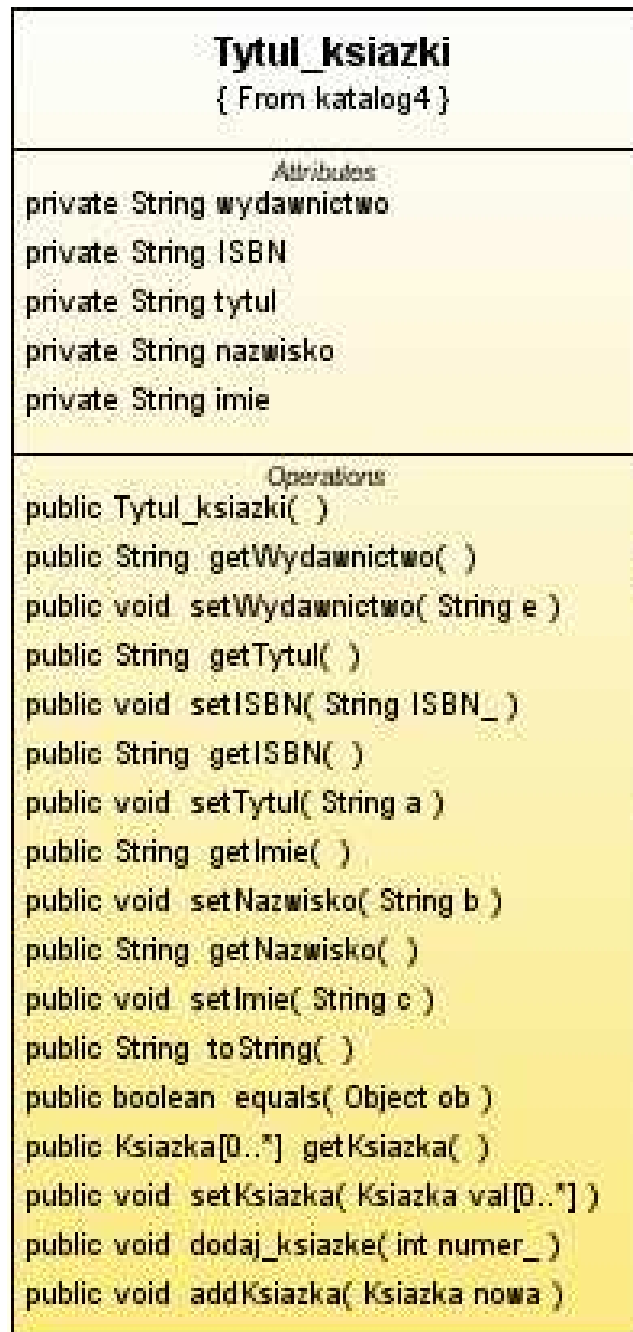
Help



Generowane diagramu klas dla całej aplikacji







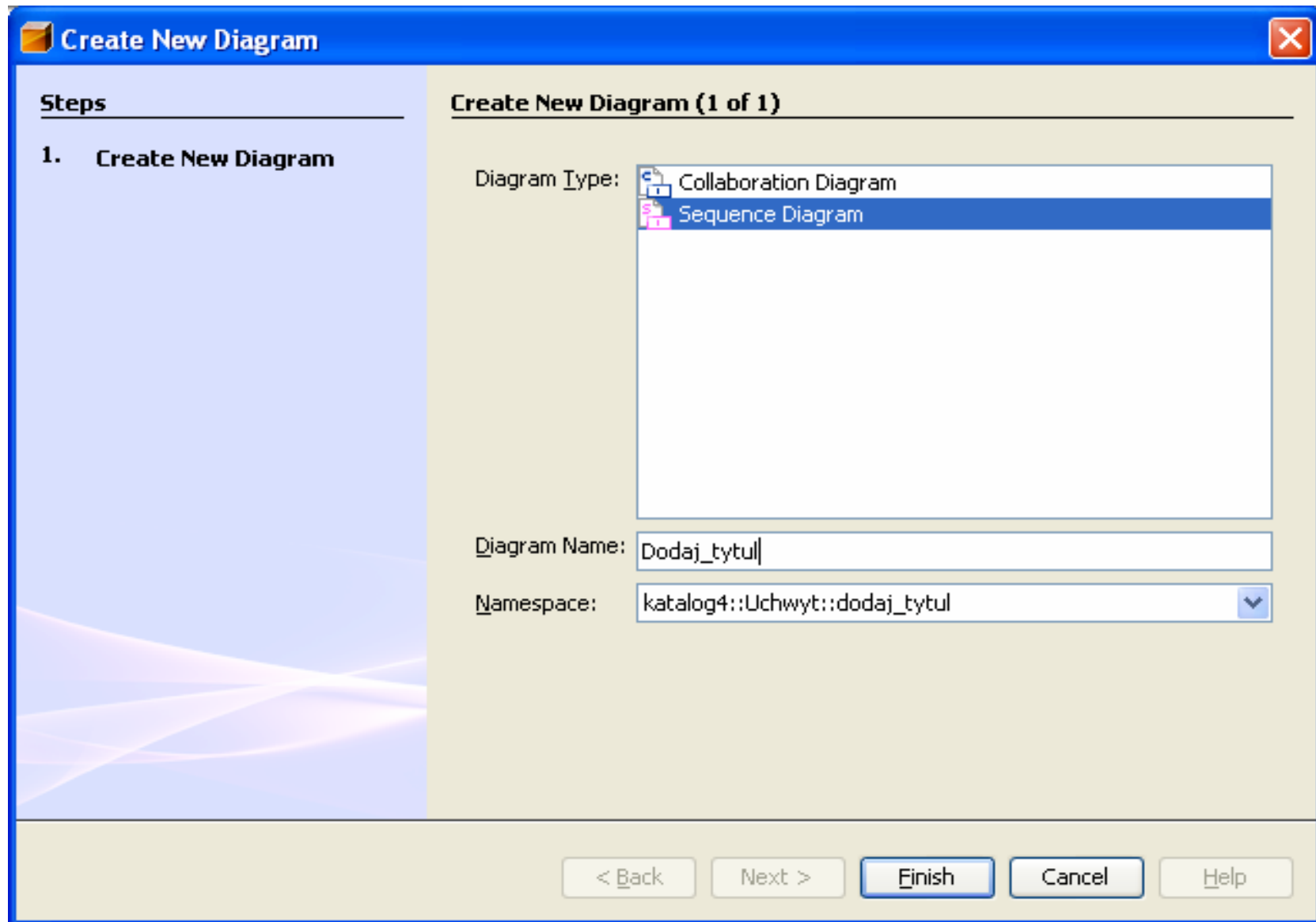
Generowanie diagramów sekwencji

The screenshot displays the NetBeans IDE 5.5.1 interface. The main window shows a UML class diagram for a book catalog system. The classes are:

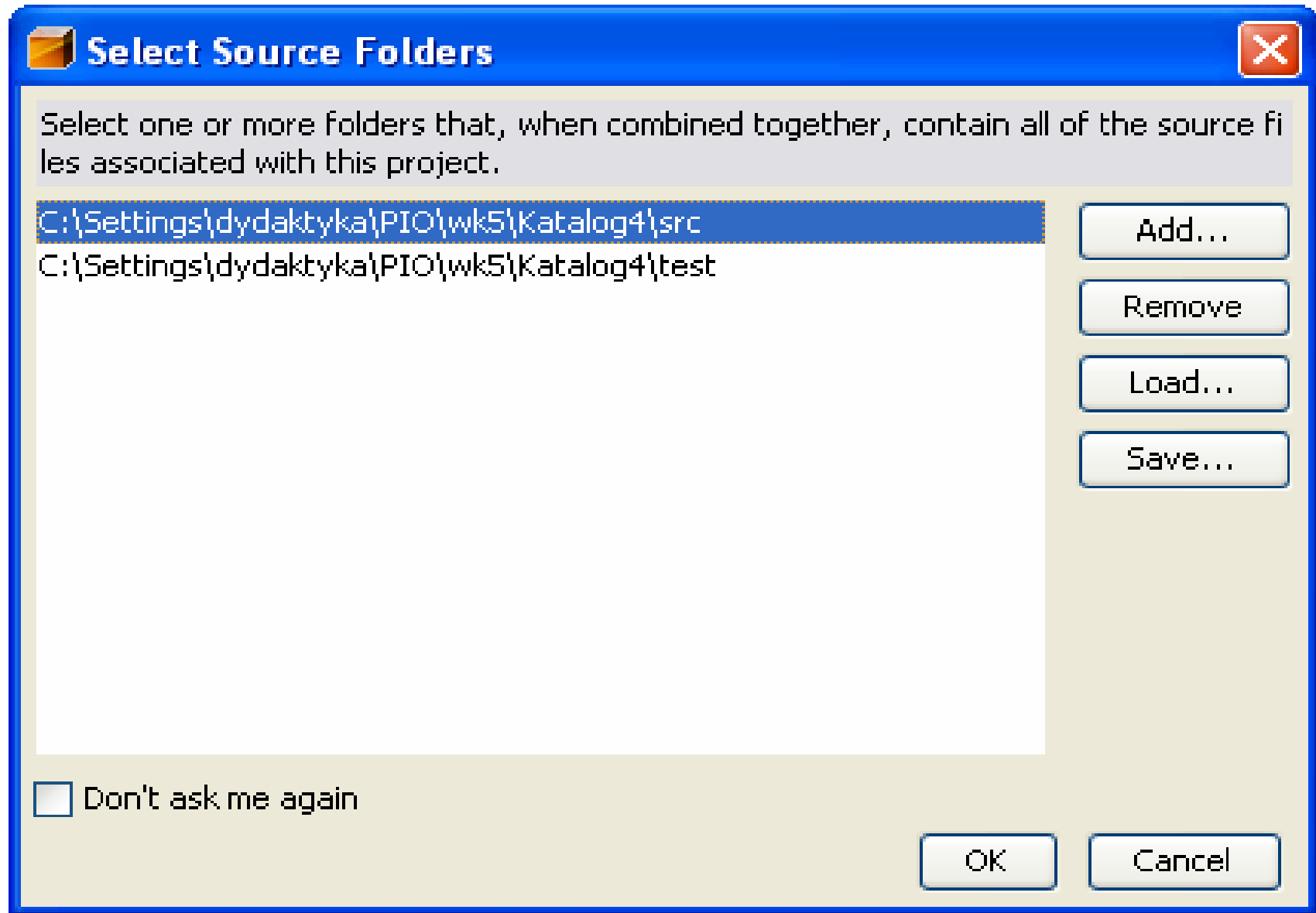
- Tytul_książki** (From katalog4):
 - Attributes: private String wydawnictwo, private String ISBN, private String tytul, private String nazwisko, private String imie
 - Operations: public Tytul_książki(), public String getWydawnictwo(), public void setWydawnictwo(String e), public String getTytul(), public void setISBN(String ISBN_), 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)
- Książka** (From katalog4):
 - Attributes: private int numer
 - Operations: public Książka(), protected void finalize(), public int getNumer(), public void setNumer(int numer_), public boolean equals(Object ob), public Tytul_książki getTytul_książki(), public void setTytul_książki(Tytul_książki val), public String toString()
- Uchwyt** (From katalog4):
 - Attributes: (None listed)
 - Operations: public Uchwyt(), public void dodaj_tytul(String a, String b, String c, String d, String e), public Tytul_książki[0..*] getTytul_książki(), public void setTytul_książki(Tytul_książki val[0..*]), public void addTytul_książki(Tytul_książki tytul_książki), public void dodaj_książke(String ISBN_, int numer_), public void main(String t[0..*])

Associations are shown between Tytul_książki and Książka, and Tytul_książki and Uchwyt, both with multiplicity 0..* at the Tytul_książki end. The diagram is zoomed to 75.56%. A context menu is open over the diagram, with the option 'Create Diagram From Selected Elements...' highlighted. The menu also includes options like 'Open', 'Add', 'Delete', 'Rename...', 'Navigate To Source', 'Apply Design Pattern...', 'Associate With...', 'Reverse Engineer Operation...', and 'Properties'.

Tworzenie diagramów sekwencji

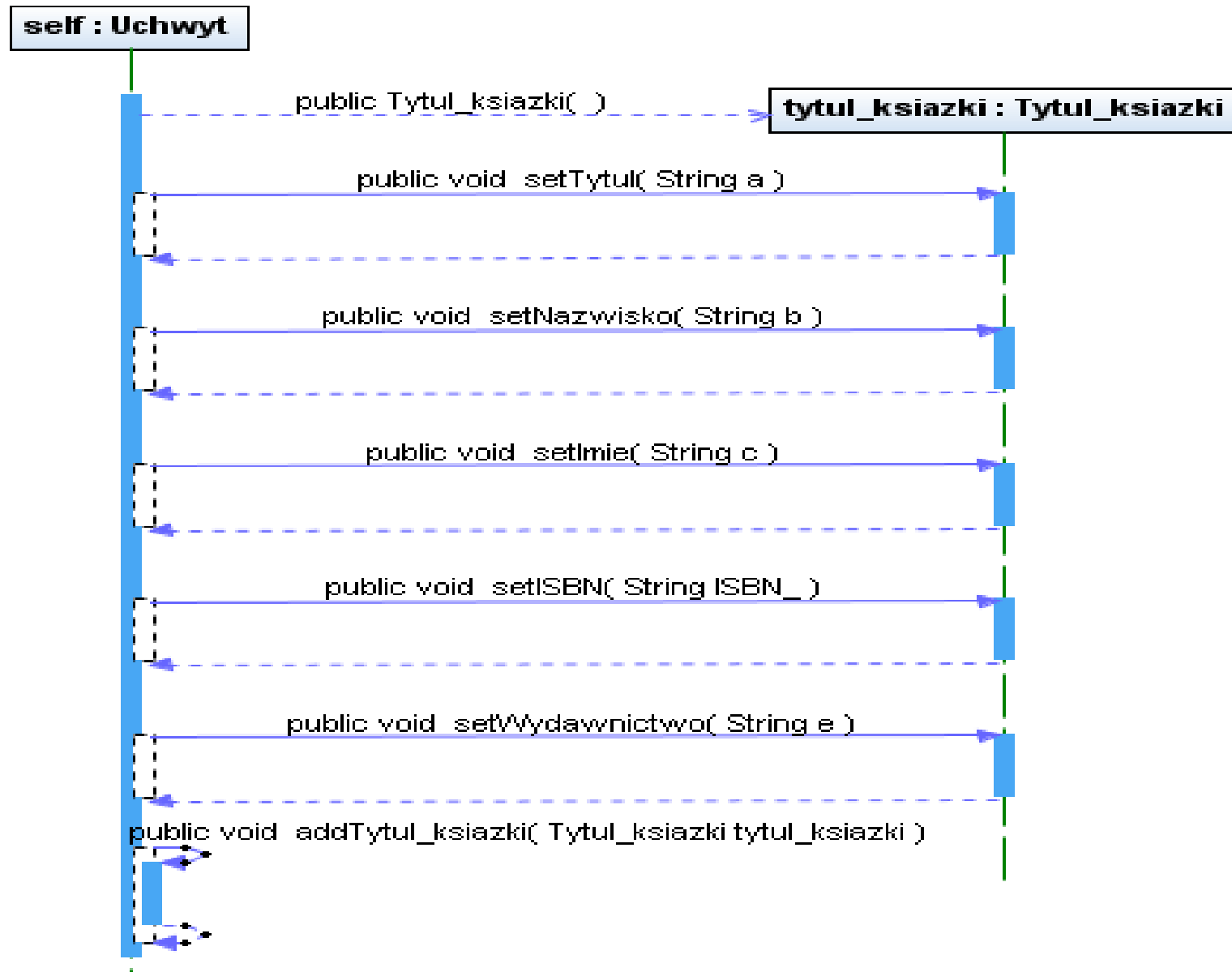


Zofia Kruczkiewicz, Podstawy
inżynierii oprogramowania

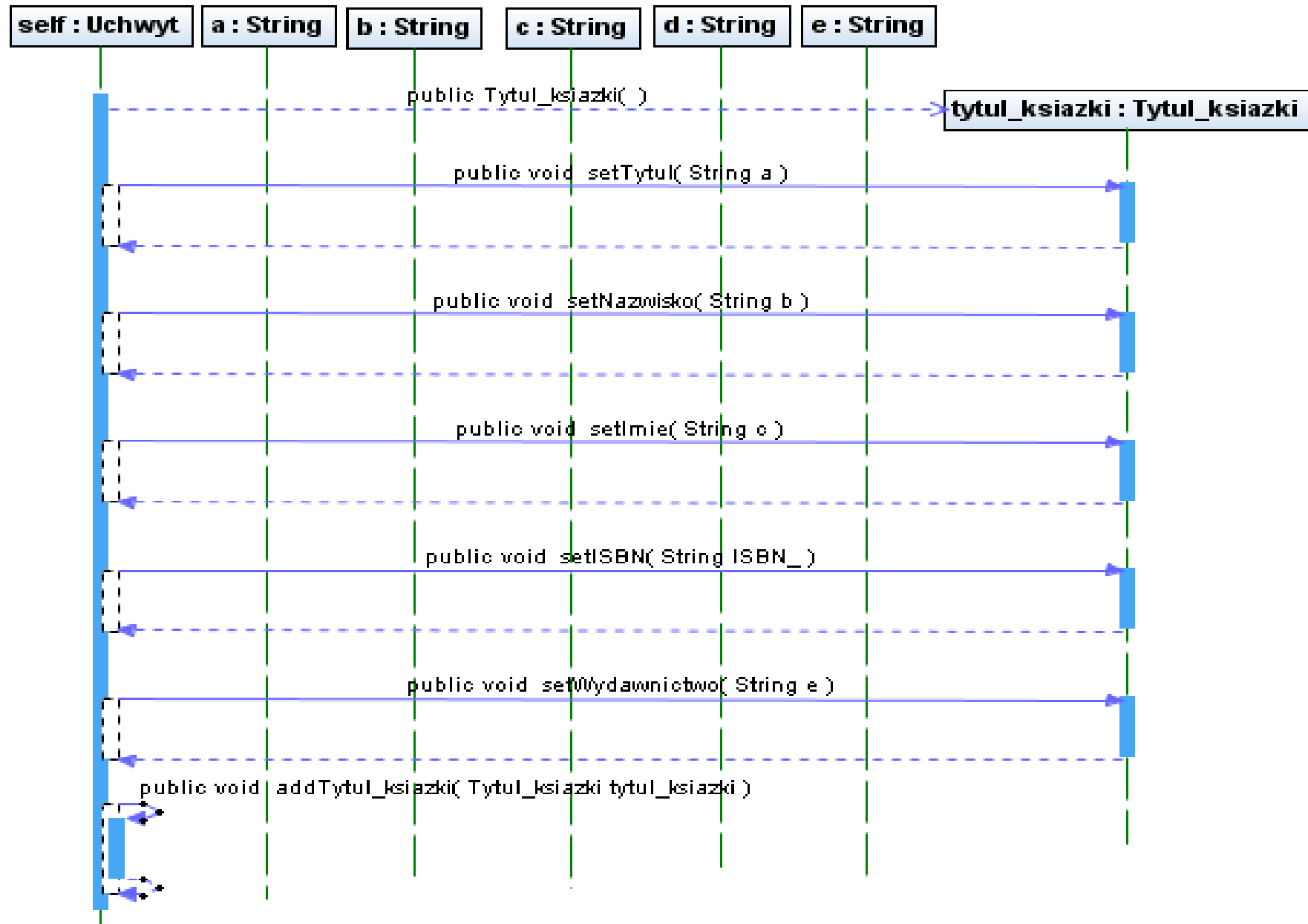


Diagramy sekwencji dla klasy Uchwyt

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



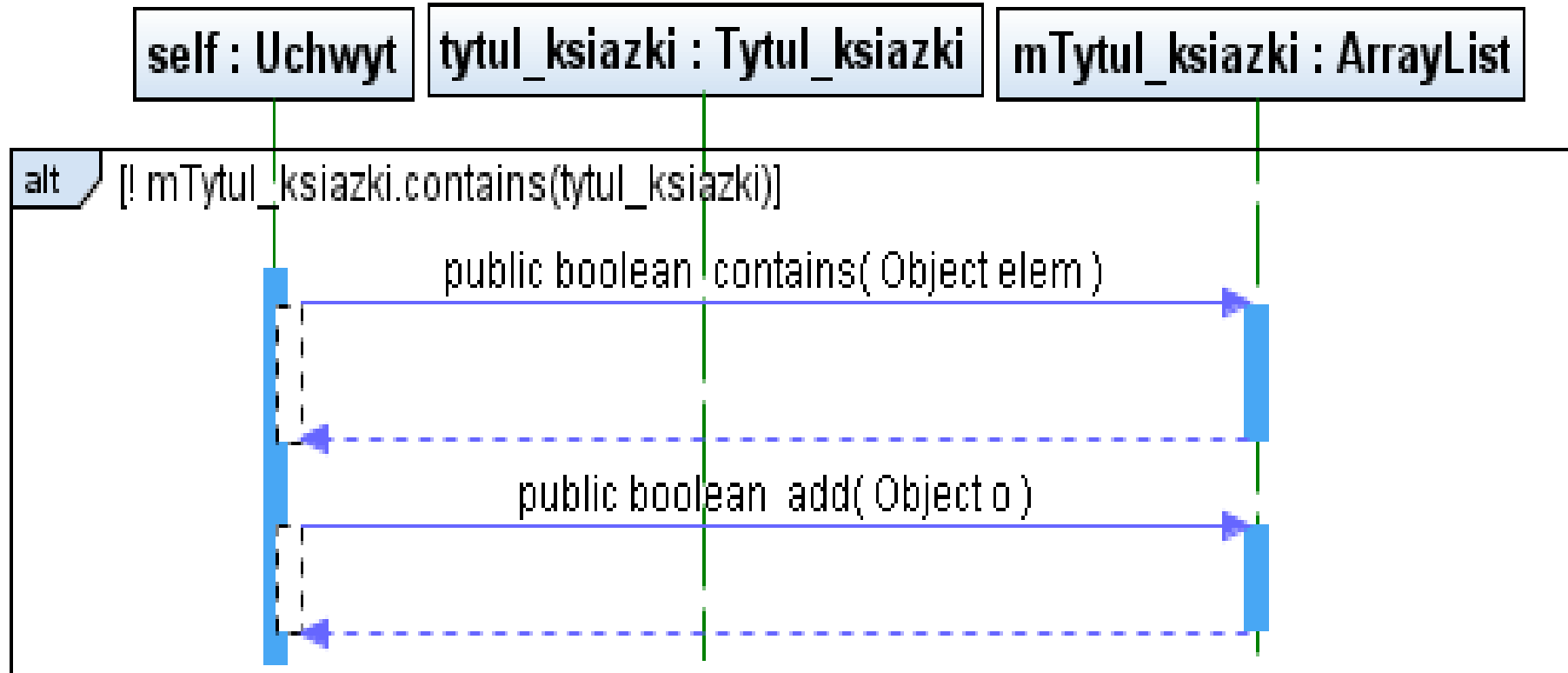
Bez poprawek – po procesie Reverse Engineering



Kod do procesu Reverse Engineering

```
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);  
}
```

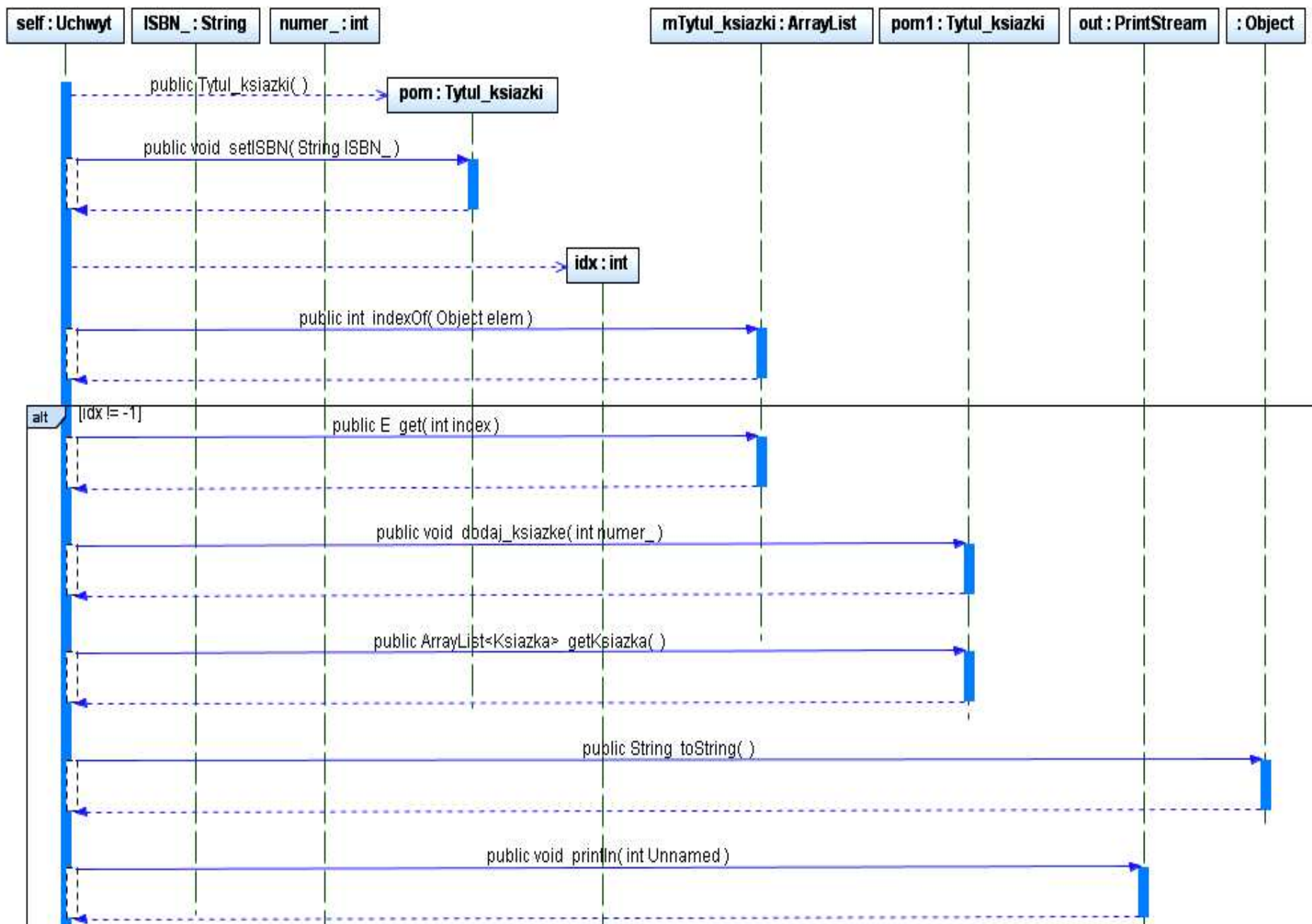
public void addTytul_książki(Tytul_książki tytul_książki) – po procesie Reverse Engineering



Kod do procesu Reverse Engineering

```
public void addTytul_książki(Tytul_książki tytul_książki)
{
    if (! mTytul_książki.contains(tytul_książki))
        mTytul_książki.add(tytul_książki);
}
```

public void dodaj_ksiazke(String ISBN_, int numer_)

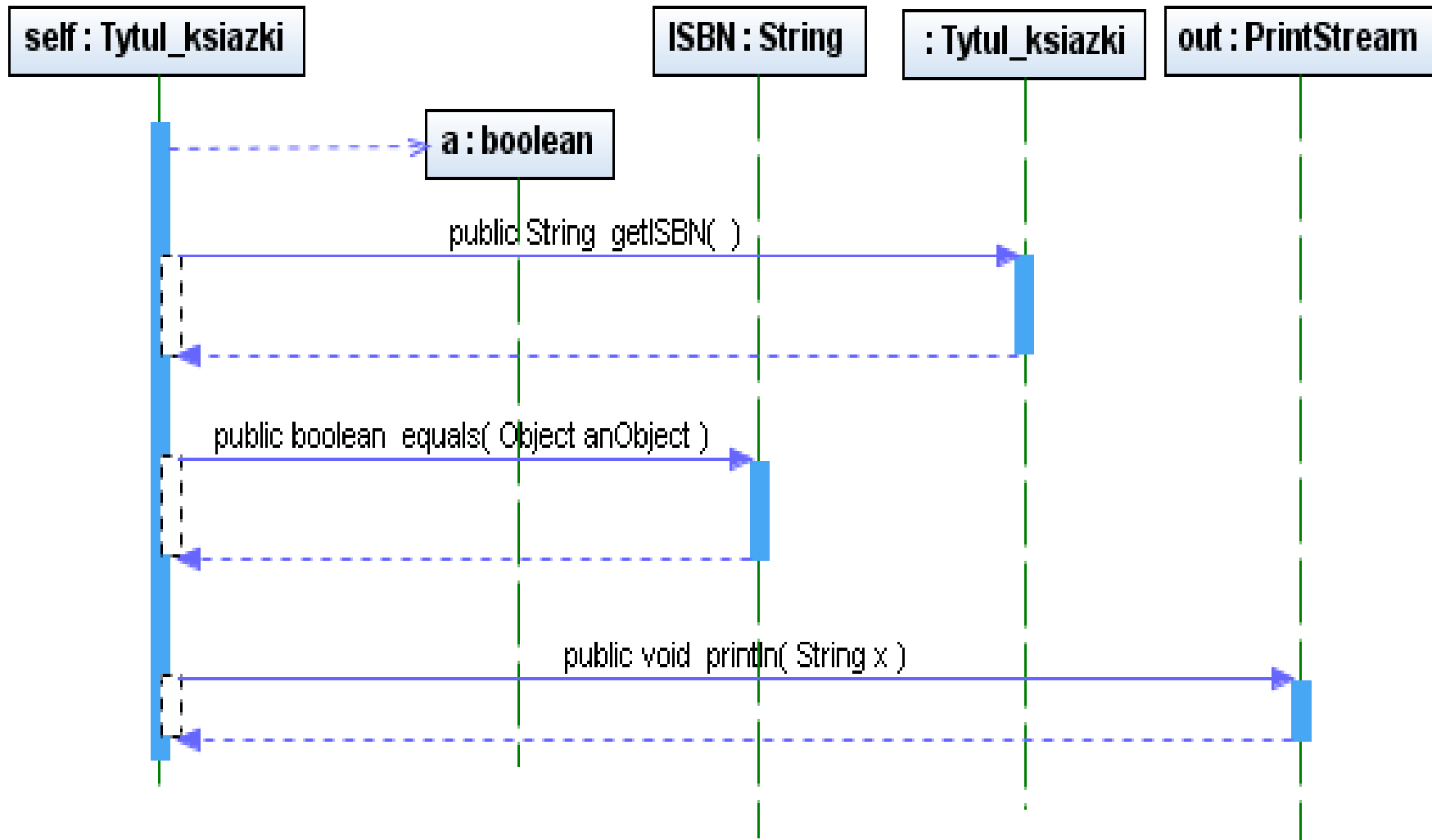


Kod do procesu Reverse Engineering

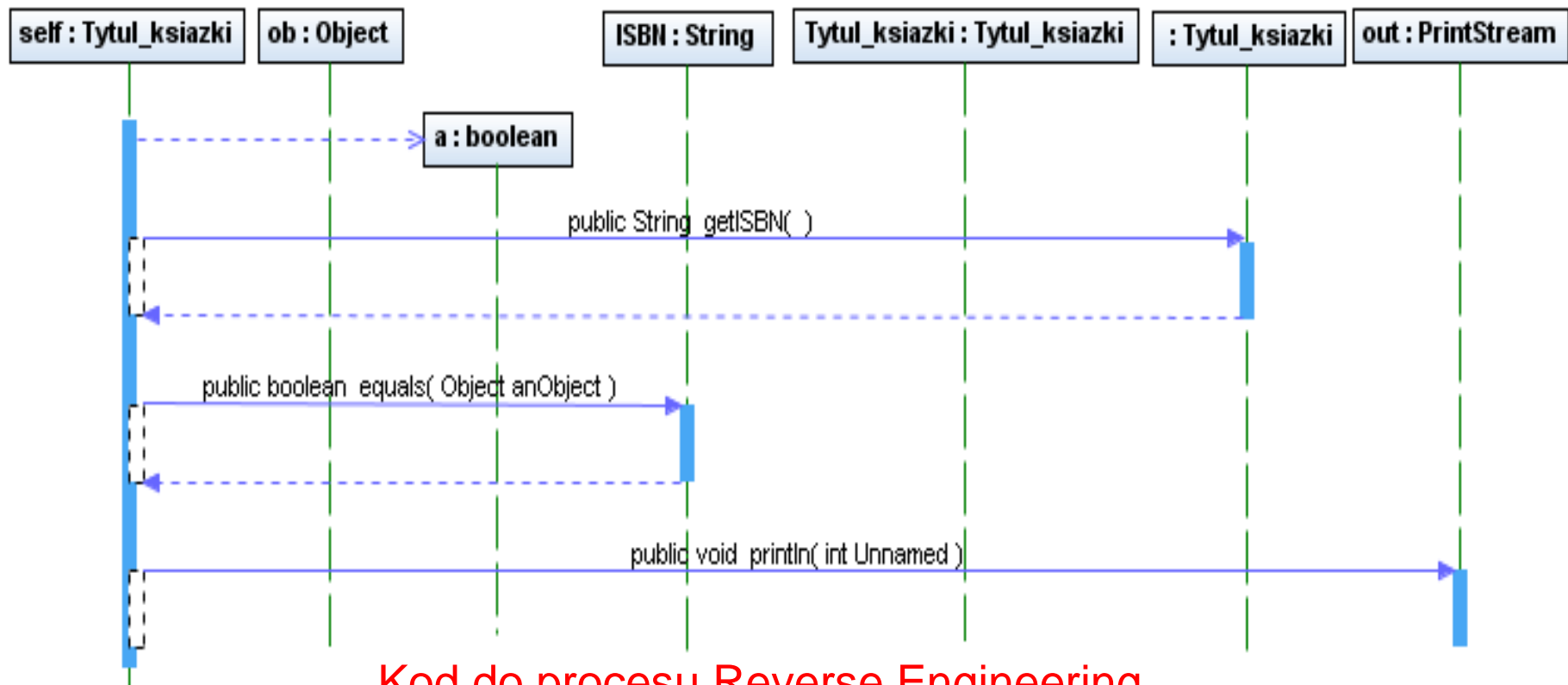
```
public void dodaj_ksiazke(String ISBN_, int numer_)
{
    Tytul_ksiazki pom=new Tytul_ksiazki();
    pom.setISBN(ISBN_);
    int idx=mTytul_ksiazki.indexOf(pom);
    if (idx!=-1)
    {
        Tytul_ksiazki pom1=mTytul_ksiazki.get(idx);
        pom1.dodaj_ksiazke(numer_);
        System.out.println(pom1.getKsiazka().toString());
    } //linia tymczasowa
}
```

Diagramy sekwencji dla klasy Tytul_ksiazki

public boolean equals(Object ob)



Bez poprawek – po procesie Reverse Engineering



Kod do procesu Reverse Engineering

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

```
{
```

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

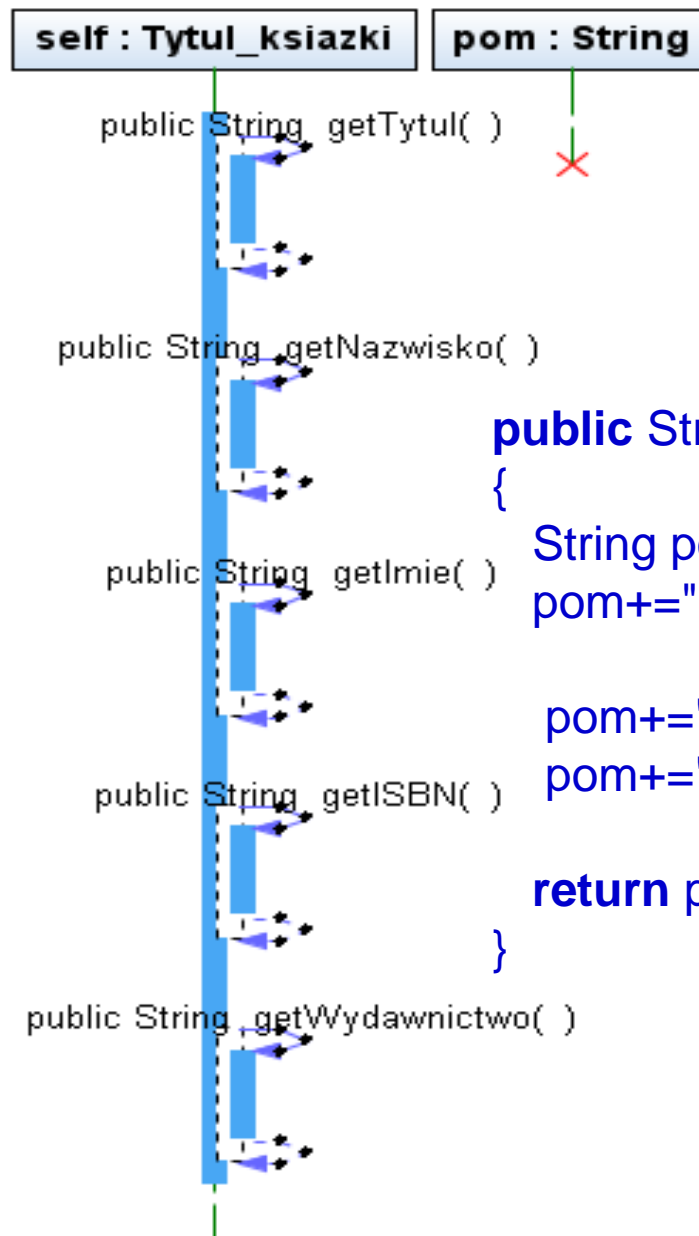
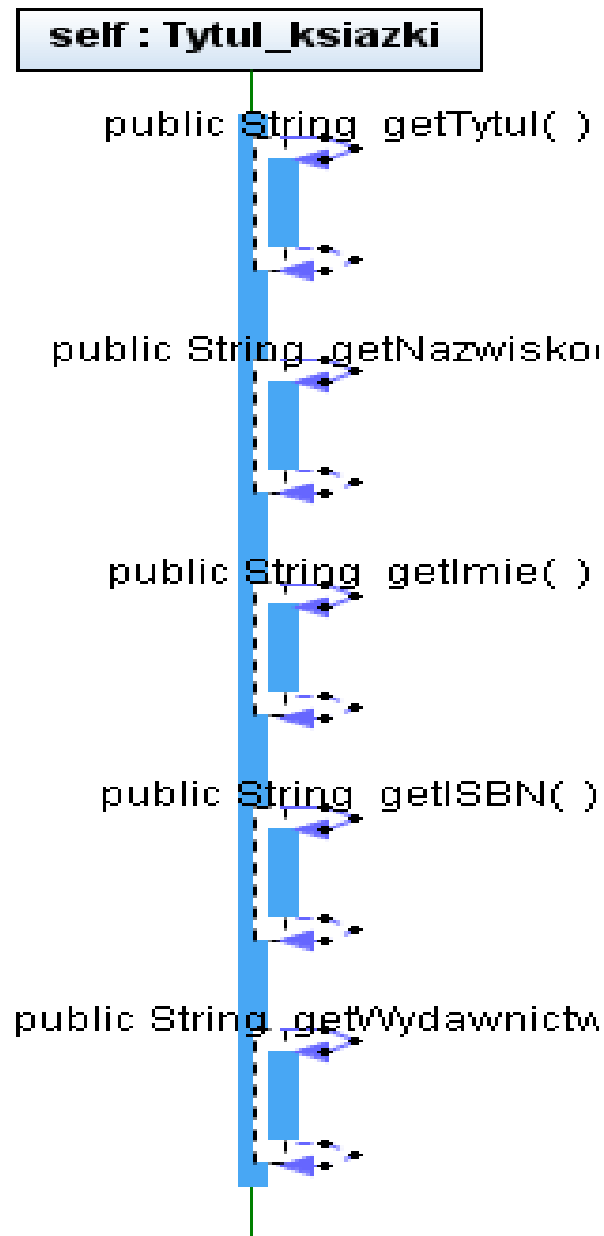
```
    System.out.println(a);           //linia tymczasowa
```

```
    return a;
```

```
}
```

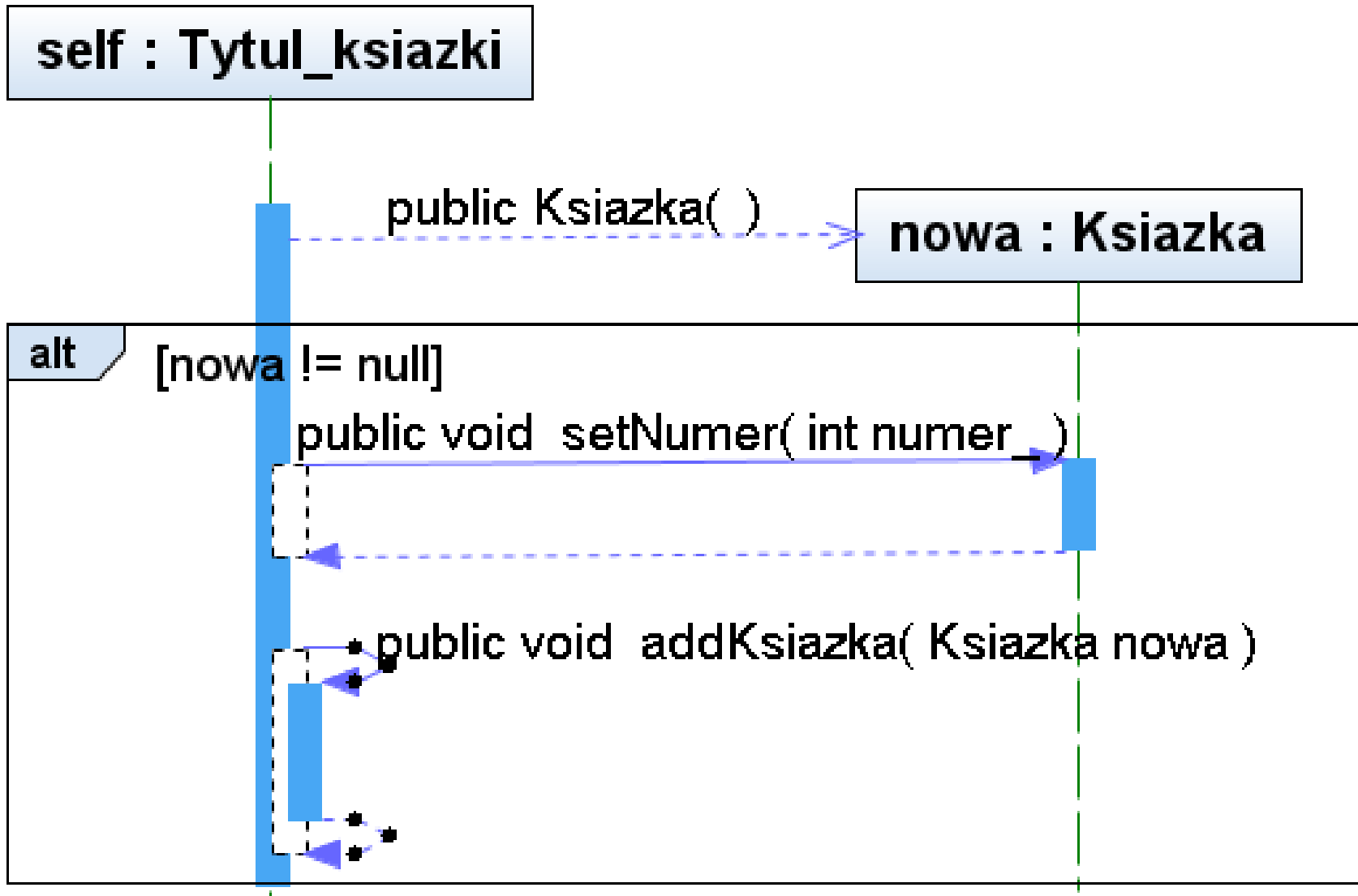
Zofia Kruczkiewicz, Podstawy
inżynierii oprogramowania

public String toString()

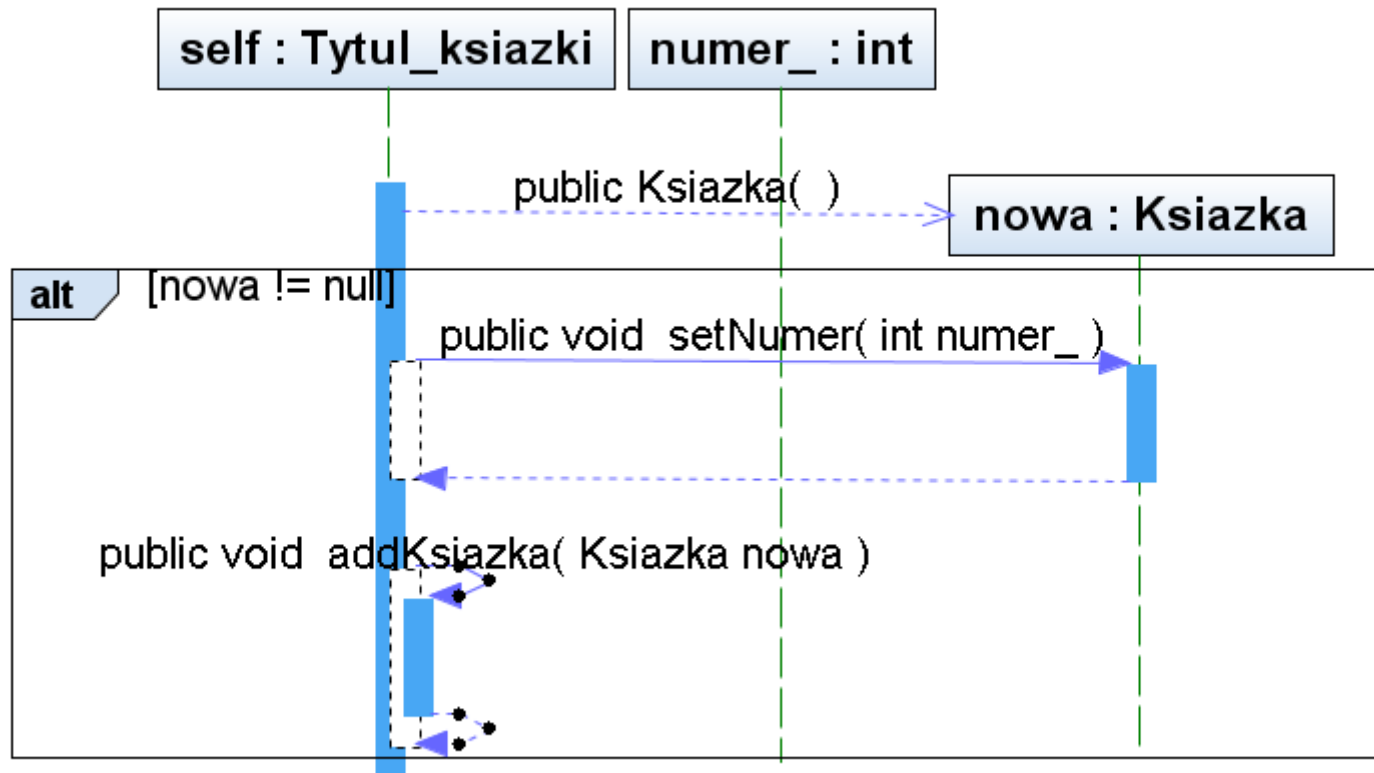


```
public String toString()
{
    String pom="Tytul: "+getTytul();
    pom+=" Autor:"+getNazwisko()
        +" "+getImie();
    pom+=" ISBN: "+getISBN();
    pom+=" Wydawnictwo:„
        +getWydawnictwo();
    return pom;
}
```

public void dodaj_ksiazke(int numer_)



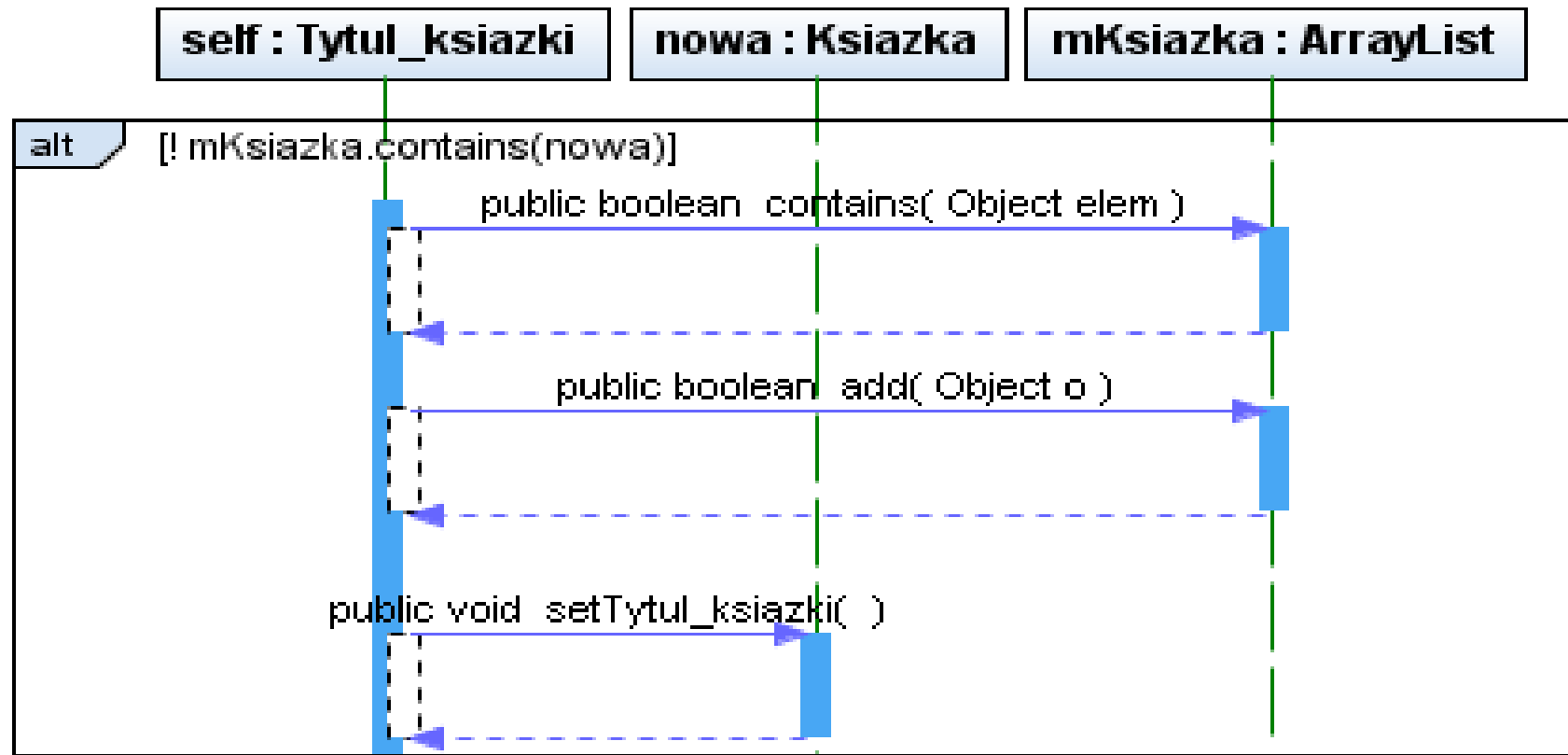
Bez poprawek – po procesie Reverse Engineering



Kod do procesu Reverse Engineering

```
public void dodaj_ksiazke(int numer_)  
{  
    Ksiazka nowa= new Ksiazka();  
    if (nowa != null)  
    { nowa.setNumer(numer_);  
      addKsiazka(nowa);}  
}
```

public void addKsiazka(Ksiazka nowa) – po procesie Reverse Engineering

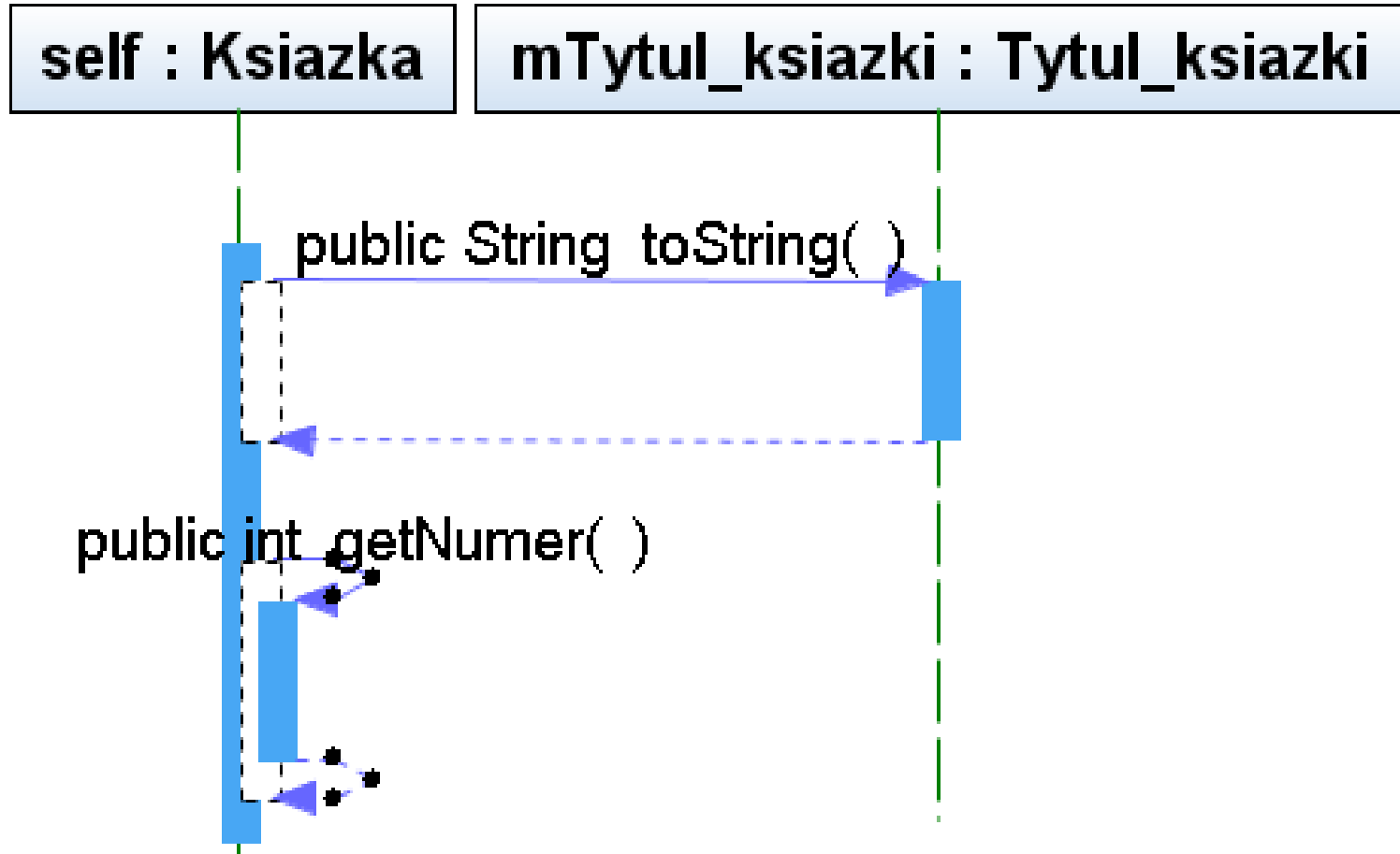


Kod do procesu Reverse Engineering

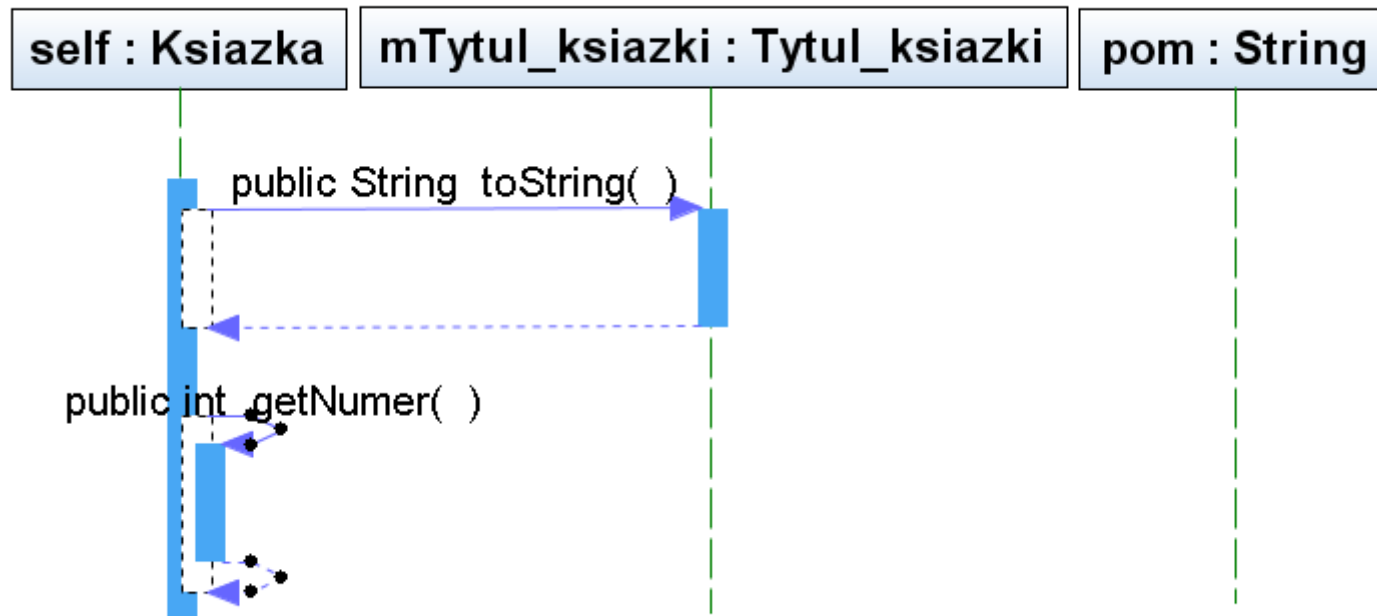
```
public void addKsiazka(Ksiazka nowa)
{
    if (!mKsiazka.contains(nowa))
    {
        mKsiazka.add(nowa);
        nowa.setTytul_ksiazki(this);
    }
}
```


Diagramy sekwencji dla klasy Ksiazka

public String Ksiazka::toString()



Bez poprawek – po procesie Reverse Engineering



Kod do procesu Reverse Engineering

```
public String toString()           // your code here
{
    String pom = mTytul_ksiazki.toString();
    pom+=" Numer: "+getNum();
    return pom;
}
```

mizymem programowania