Instrukcja 1
Laboratorium z Podstaw Inżynierii Oprogramowania

Relacja 1 do 1..0– instrukcja z lab1

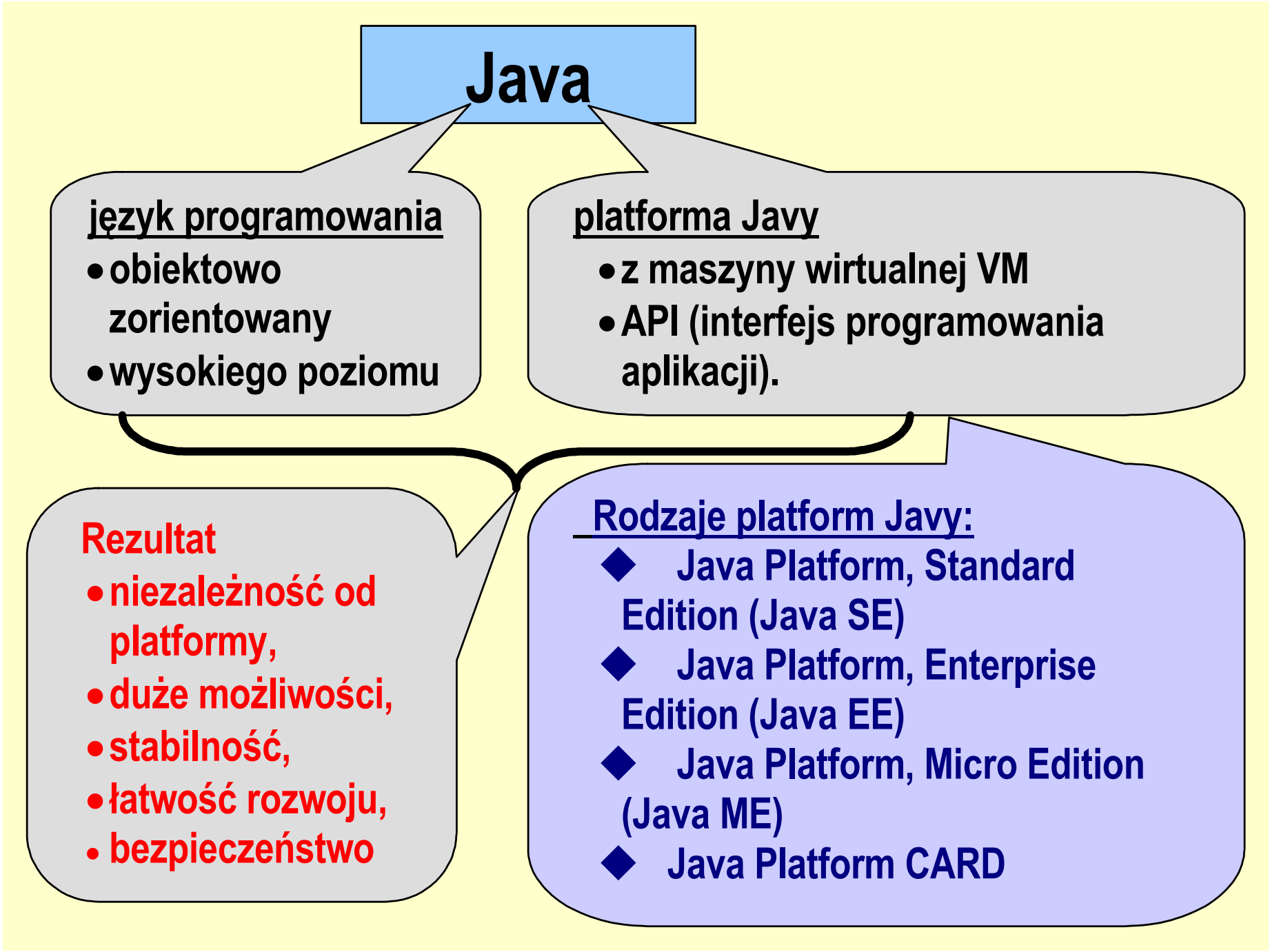
Cele laboratorium 1

Należy:

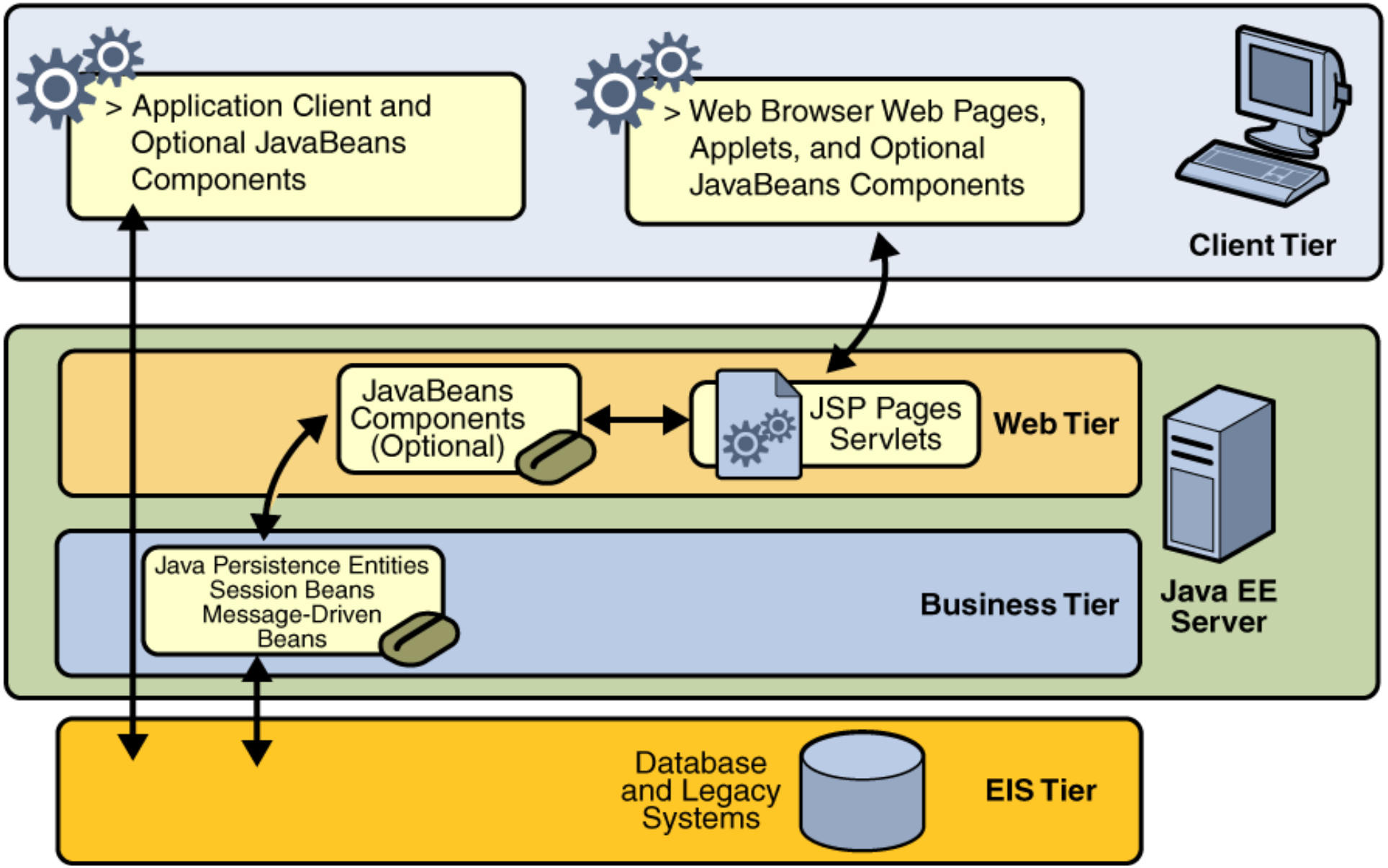
- wybrać projekt z podanej listy dostępnej za pomocą linku podanego w w laboratorium 1

- sformułować wymagania funkcjonalne i niefunkcjonalne dla wybranego projektu jako zadanie domowe. Zadanie domowe będzie stanowić podstawę do zaprojektowania przypadków użycia na kolejnych laboratorium.

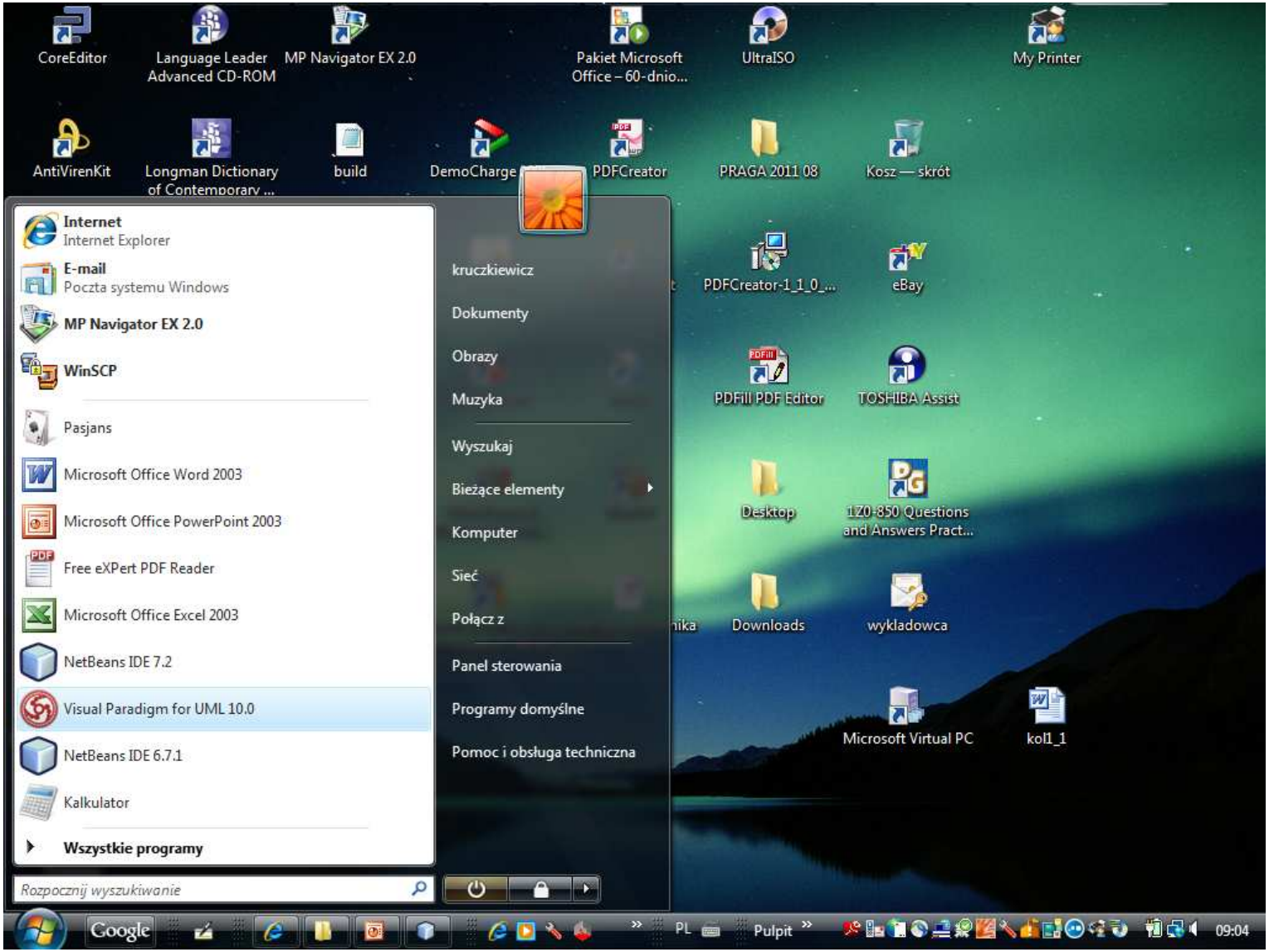
- wykonać projekt UML i wykonać prosty program stanowiący realizację projektu zgodnie z materiałem zawartym na slajdach 5-84. Jest to ćwiczenie, które pozwala poznać narzędzie Visulal Paradigm 10 for UML 2.0 wykorzystane w ramach zajęć laboratoryjnych z przedmiotu Podstawy Inżynierii Oprogramowania.


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

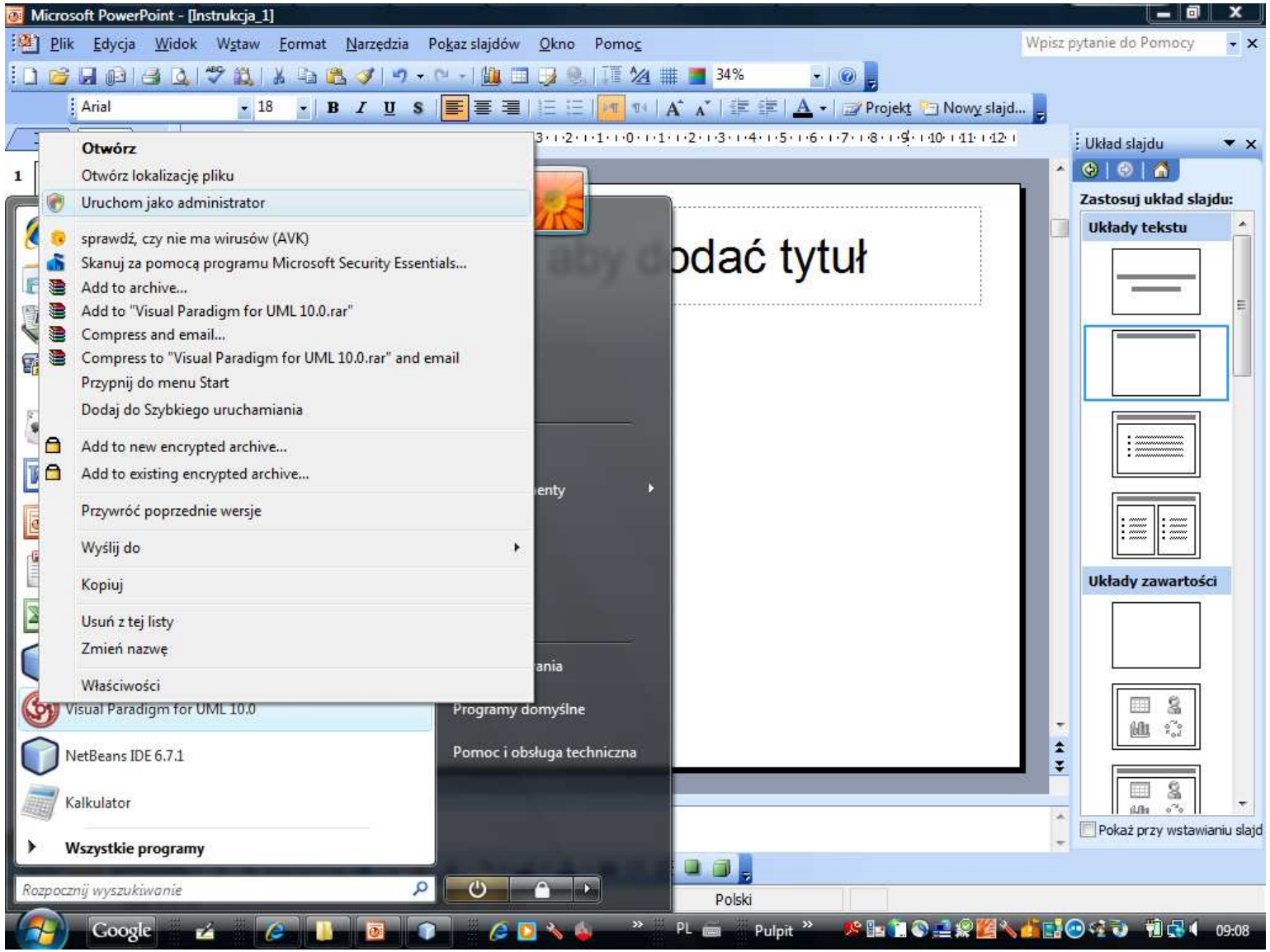
Warstwy aplikacji (Java EE)



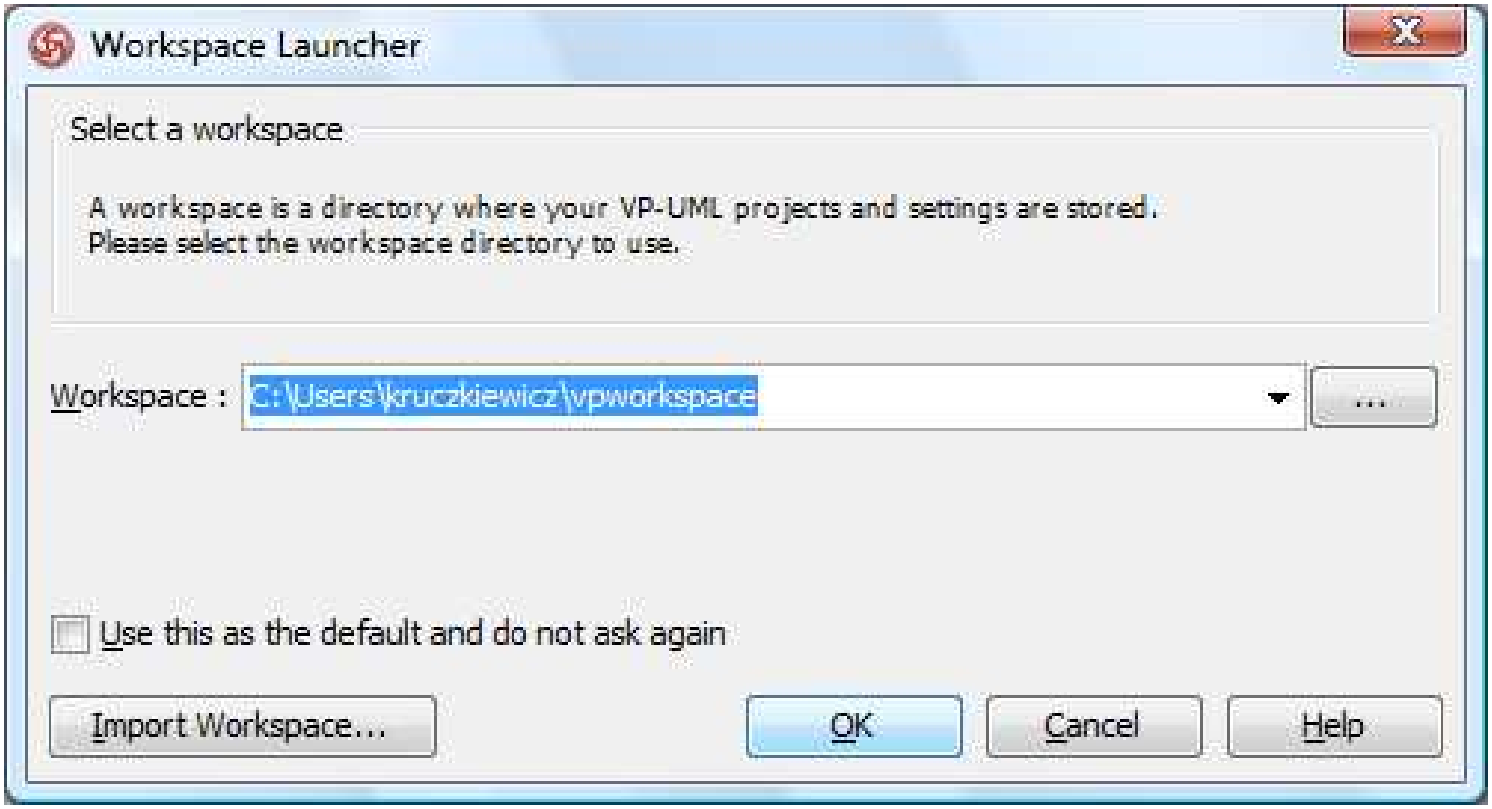
1.1. Uruchomienie programu programu *Visual Paradigm* z uprawnieniami administratora
- kliknięcie prawym klawiszem myszy na nazwę programu



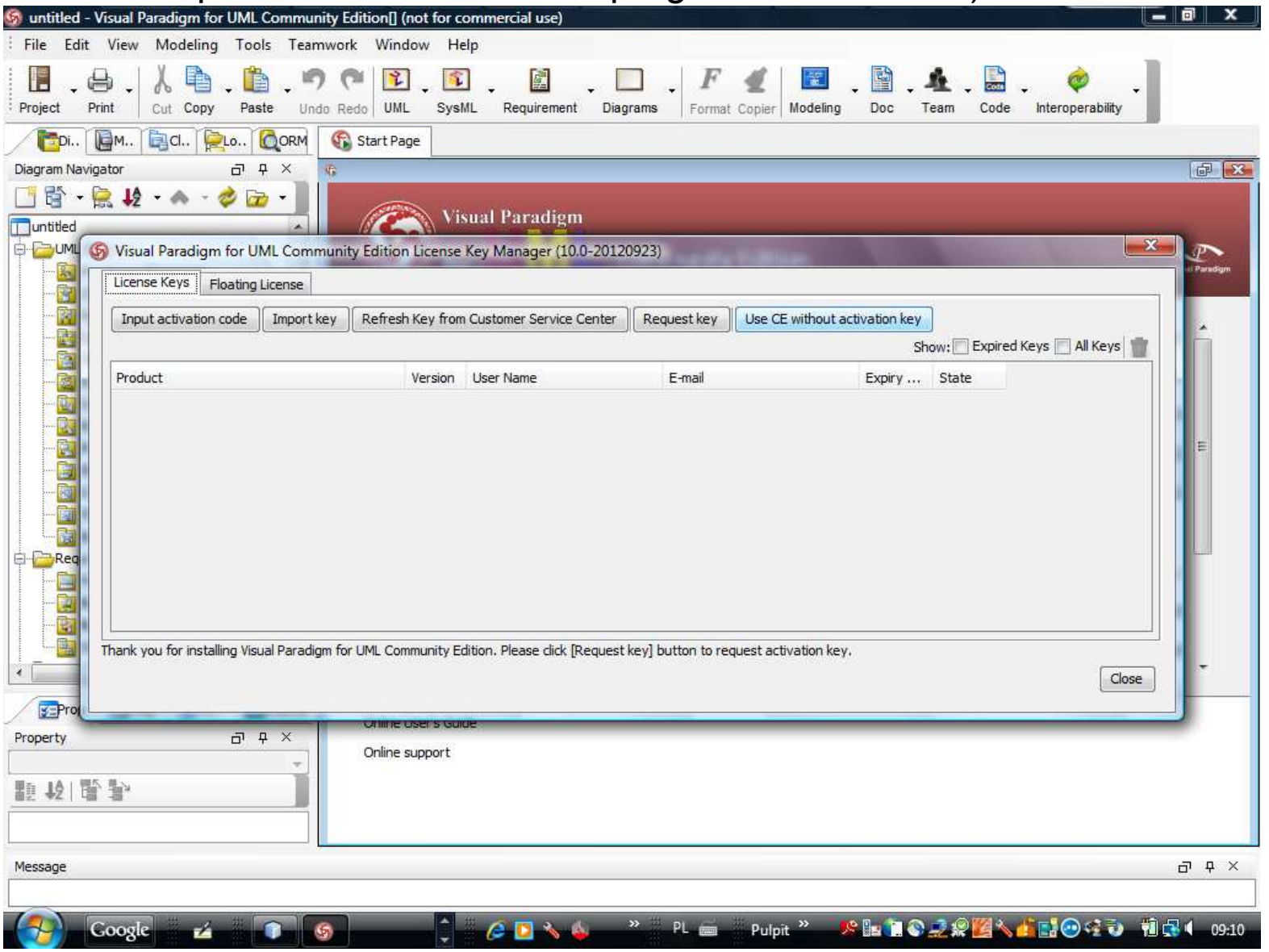
1.2. Uruchomienie programu programu *Visual Paradigm* z uprawnieniami administratora
- wybór z listy opcji *Uruchom jako administrator*



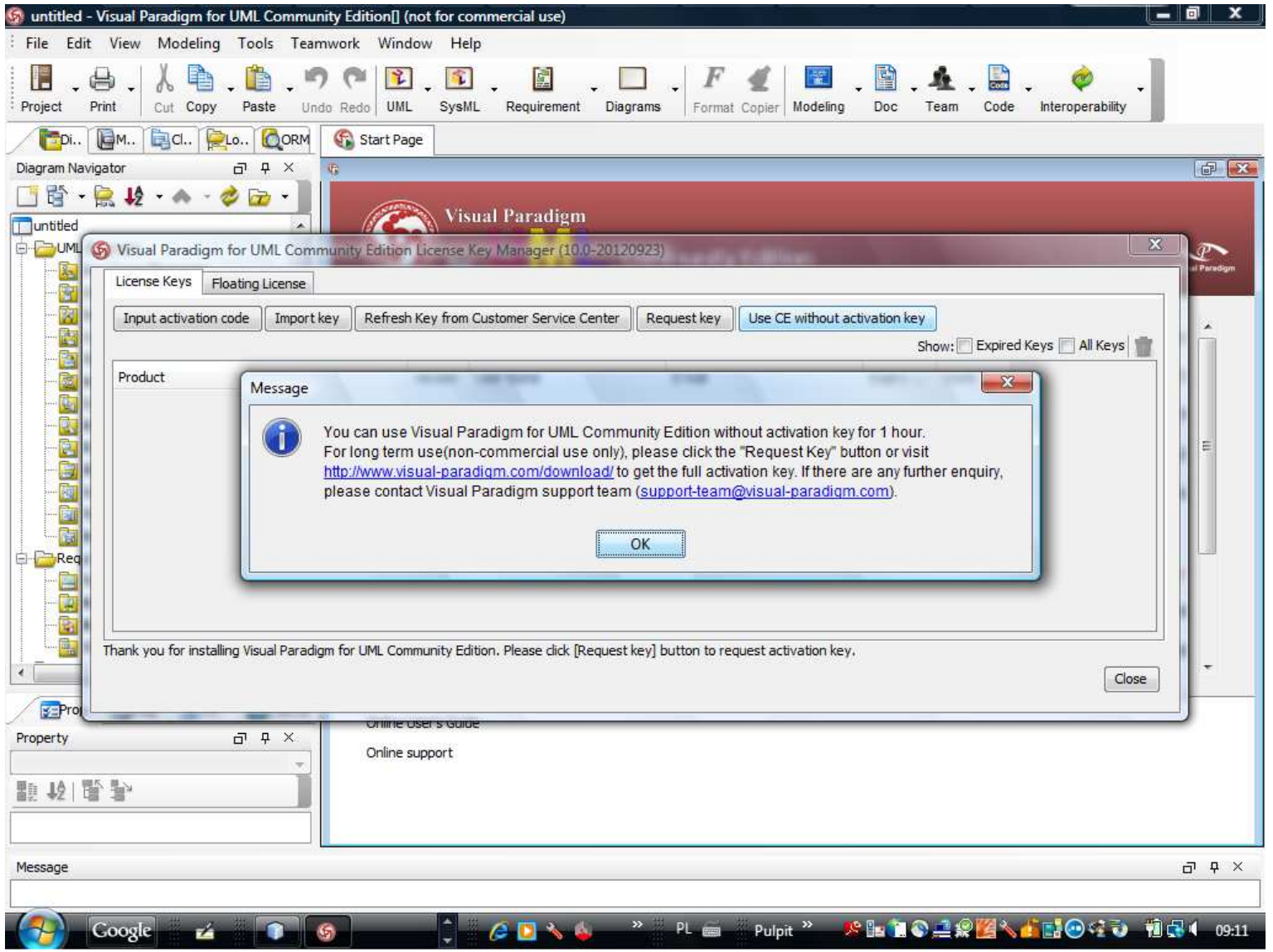
2. Wybór katalogu do tworzenia projektów



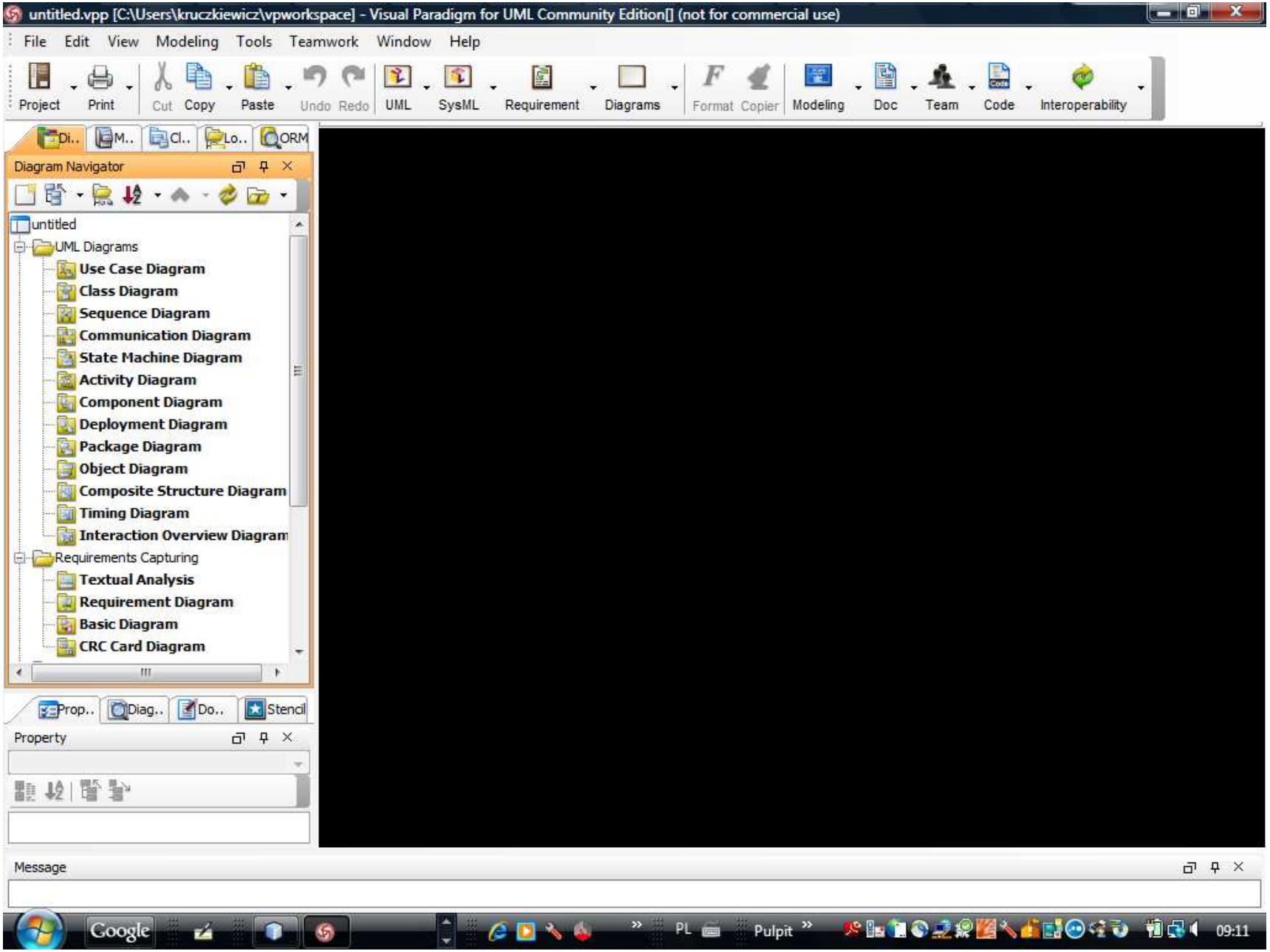
3.1. Uruchomienie wersji *Community* bez klucza
(co godzinę automatycznie pojawia się to okienko w celu kolejnego potwierdzenia działania programu bez klucza)



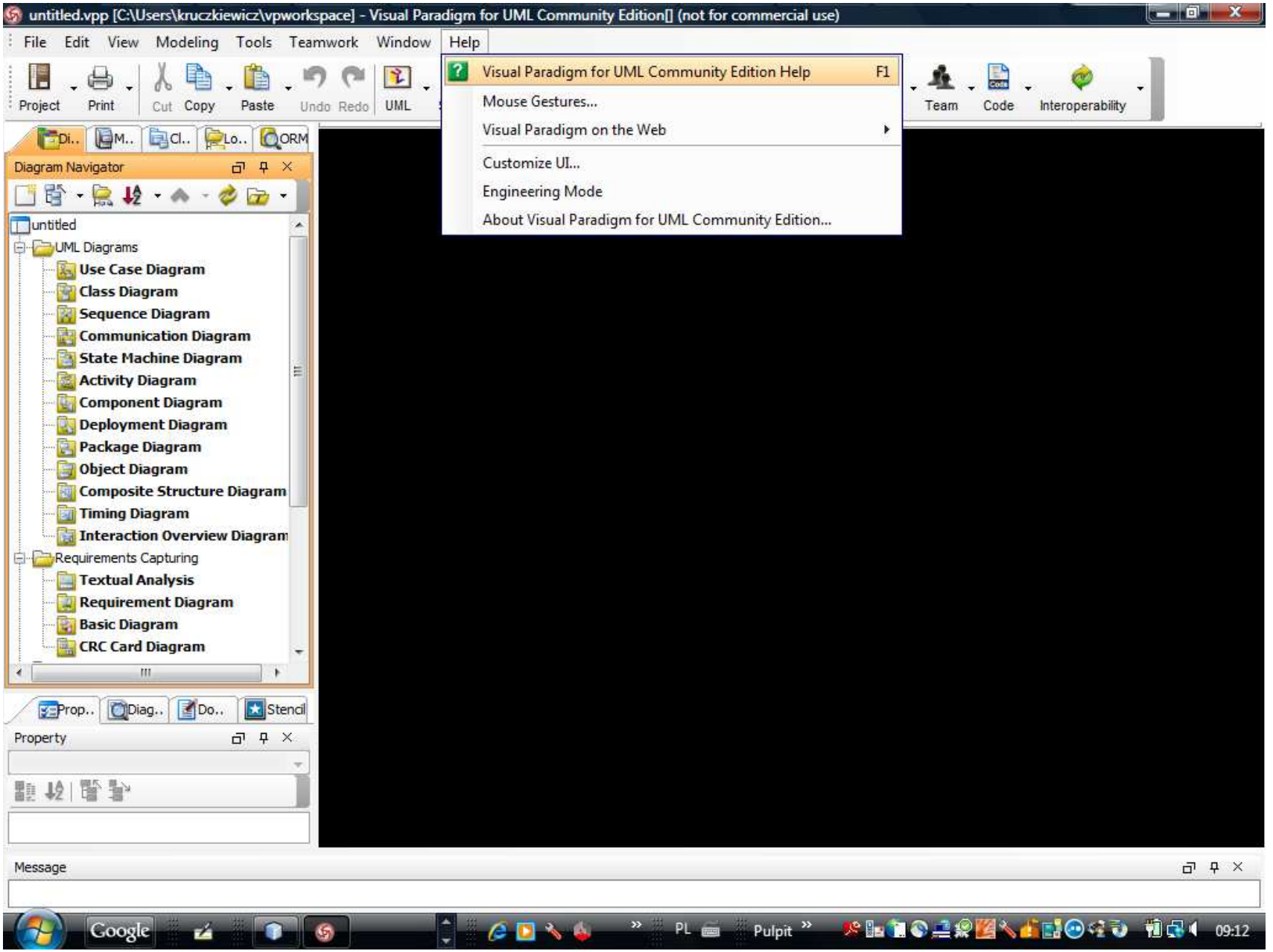
3.2. Uruchomienie wersji *Community* bez klucza (co godzinę automatycznie pojawia się to okienko w celu kolejnego potwierdzenia działania programu bez klucza)



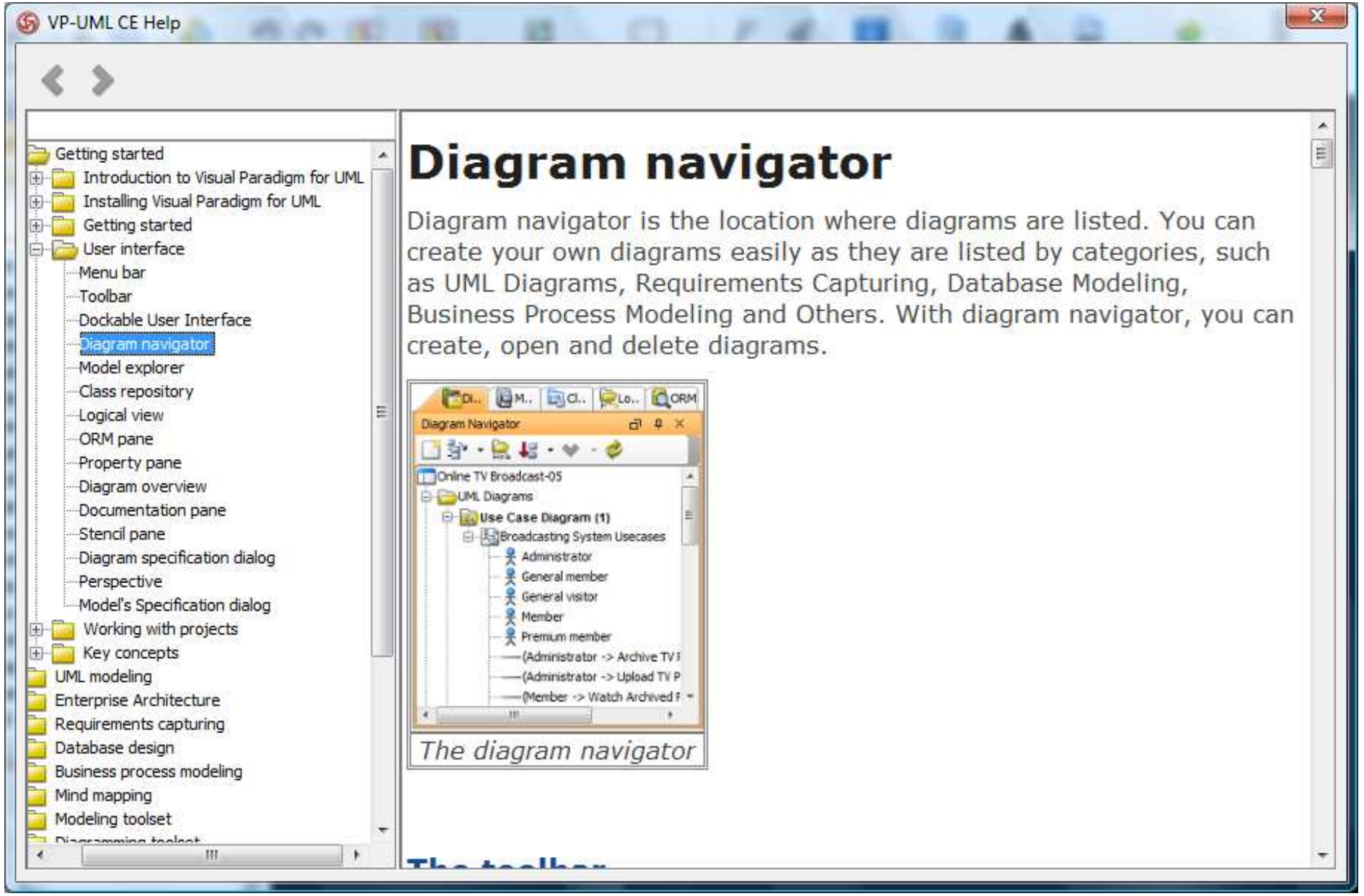
4. Start programu z pustym projektem bez nazwy



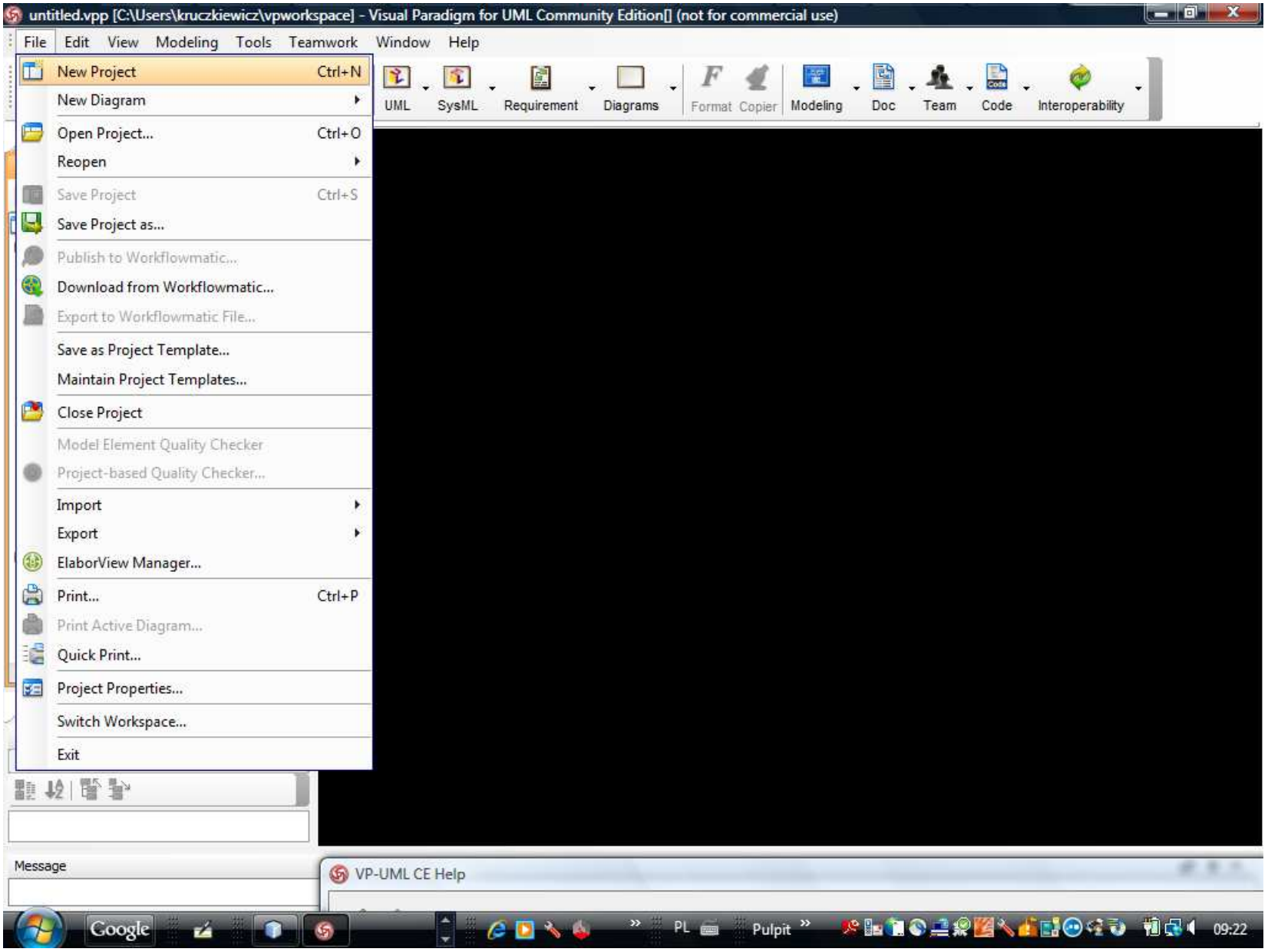
5. Ściągnięcie części *Help* ze strony producenta narzędzia



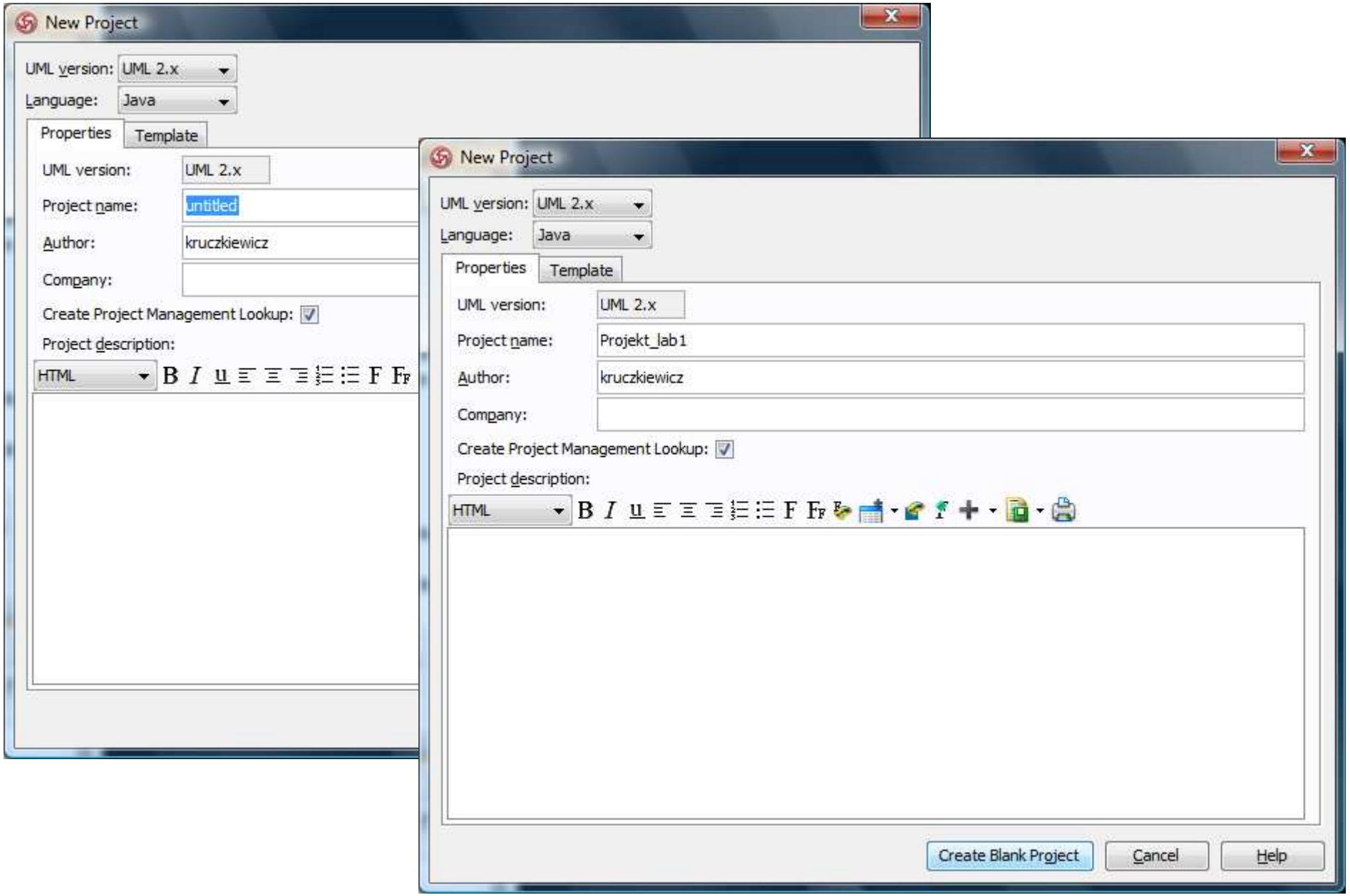
6. Przykładowe informacje dostarczone z części *Help*



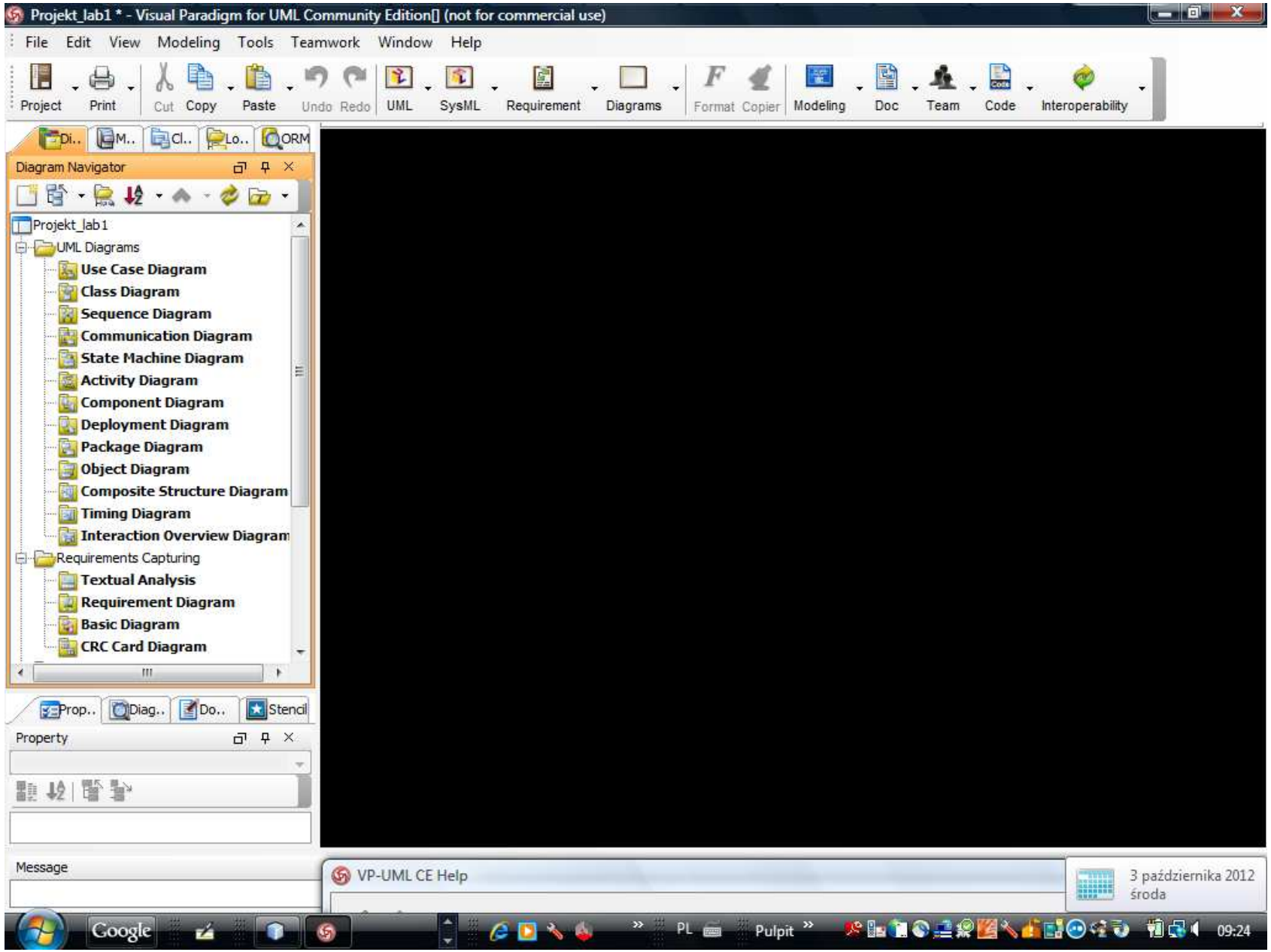
7.1. Utworzenie nowego projektu – *File/New Project*



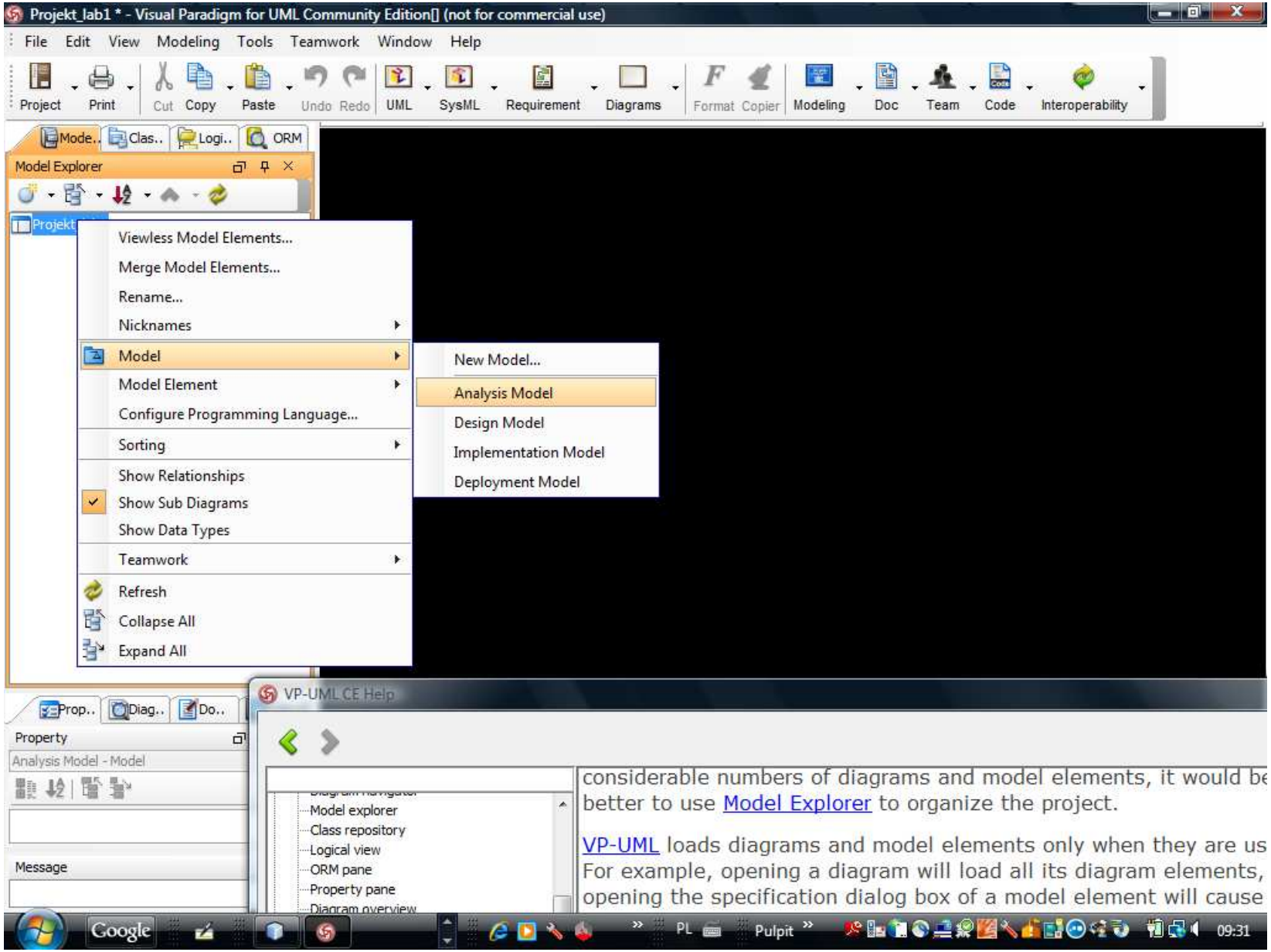
7.2. Utworzenie nowego projektu – nadanie nazwy nowemu projektowi w formularzu *New Project* w polu *Project name*



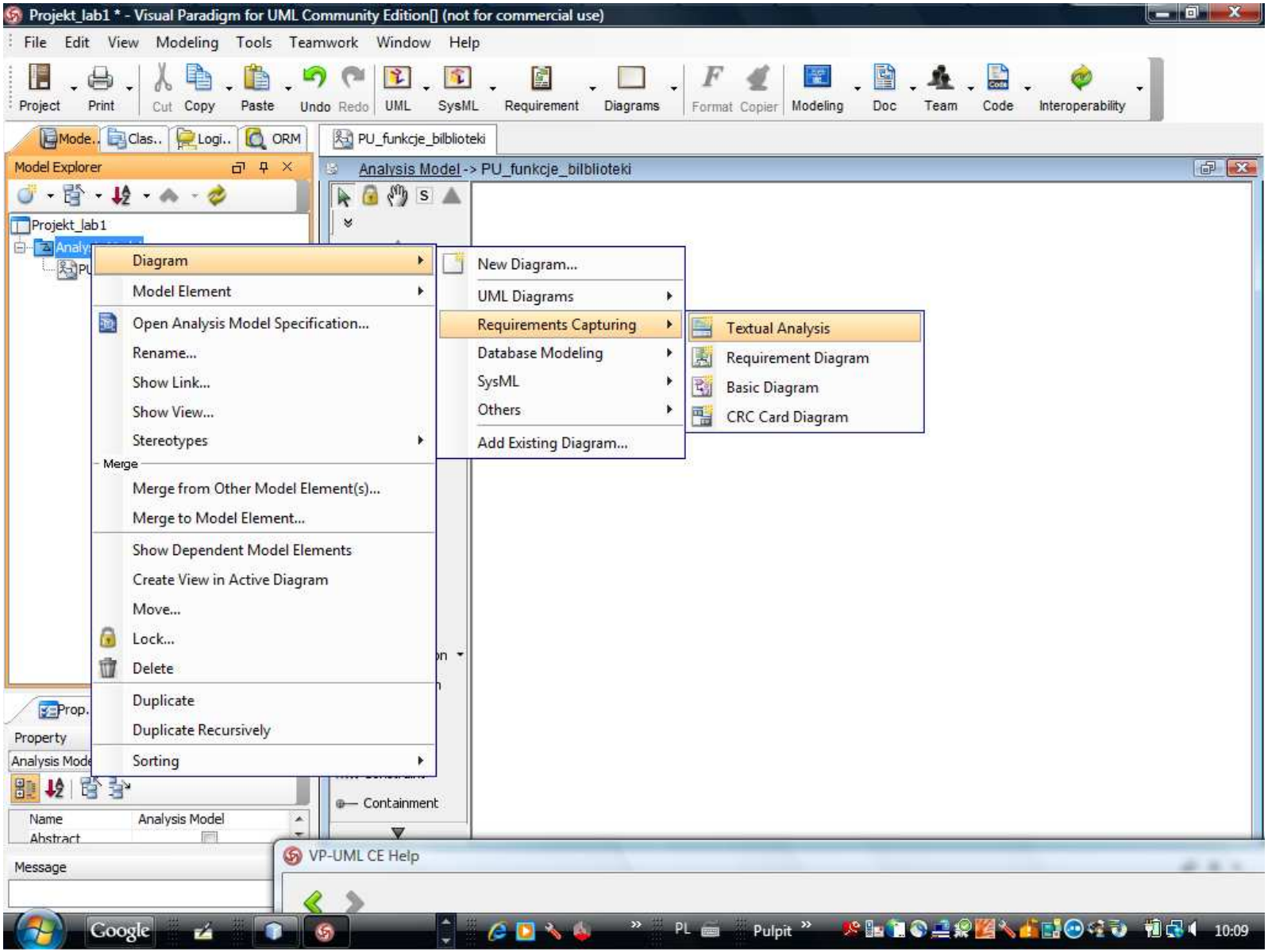
8. Widok nowego, pustego projektu – w stylu *Diagram Navigator*



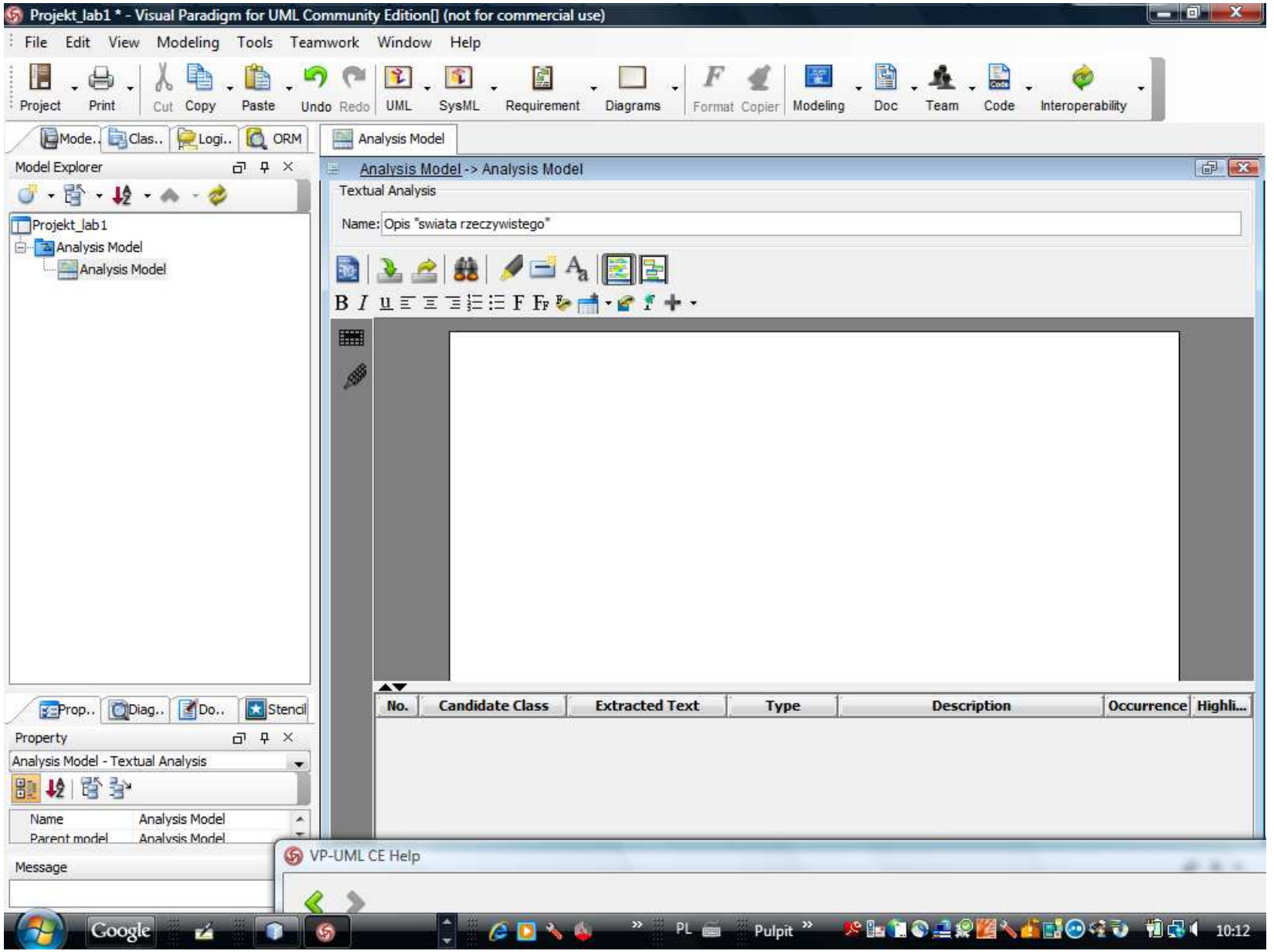
9. Utworzenie modelu analizy w projekcie – po kliknięciu prawym klawiszem na nazwę projektu należy wybrać z list: *Model/Analysis Model*



10.1. Dodanie diagramu reprezentującego opis „świata rzeczywistego” – po kliknięciu prawym klawiszem myszy na nazwę modelu wybór z list: *Diagram/Requirements Caturing/Textual Analysis*



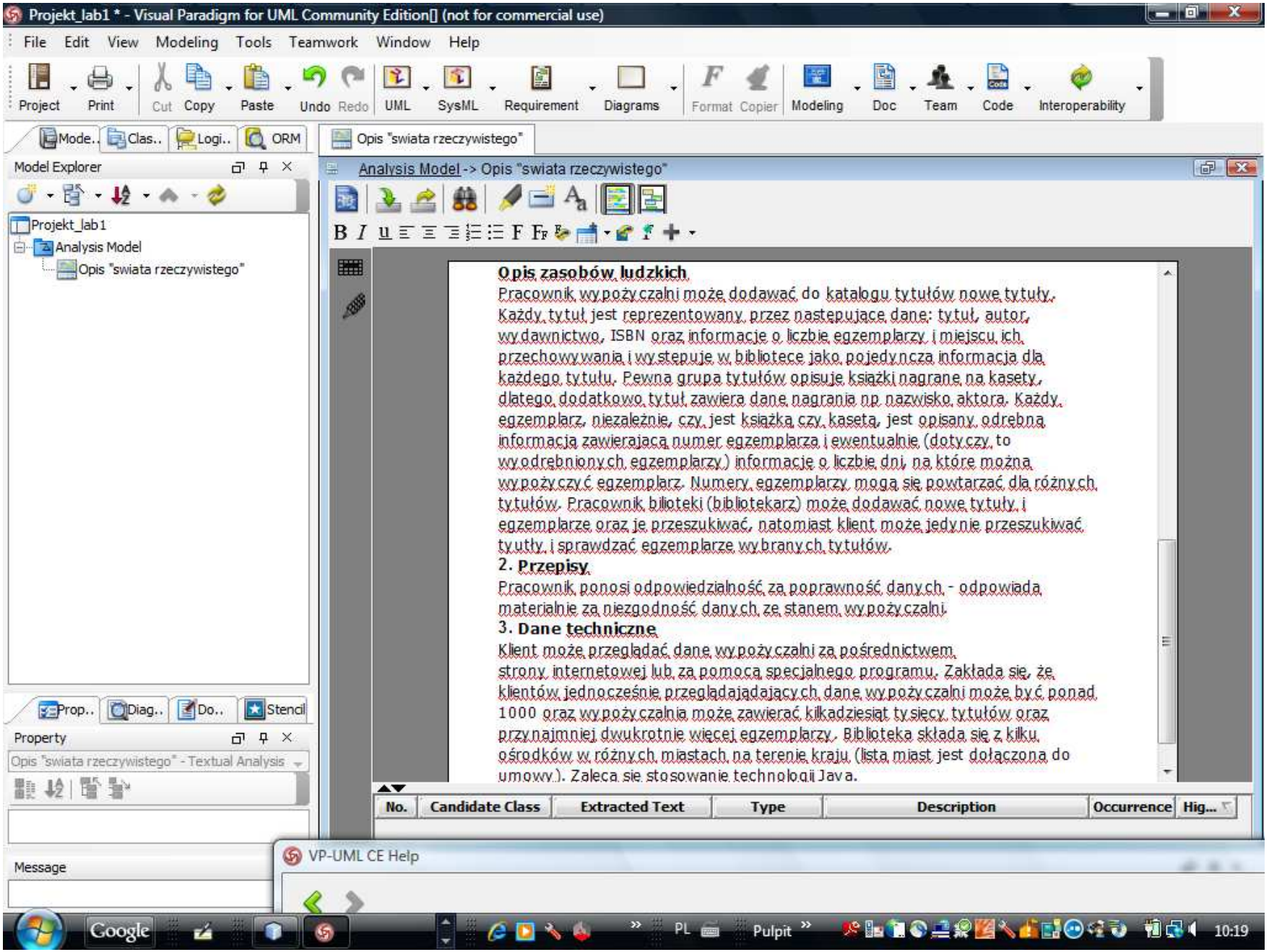
10.2. Okno diagramu – należy nadać nazwę digramowi np. Opis „świata rzeczywistego” i należy wprowadzić tekst reprezentujący opis „świata rzeczywistego” projektu (następny slajd zawiera przykład opisu)



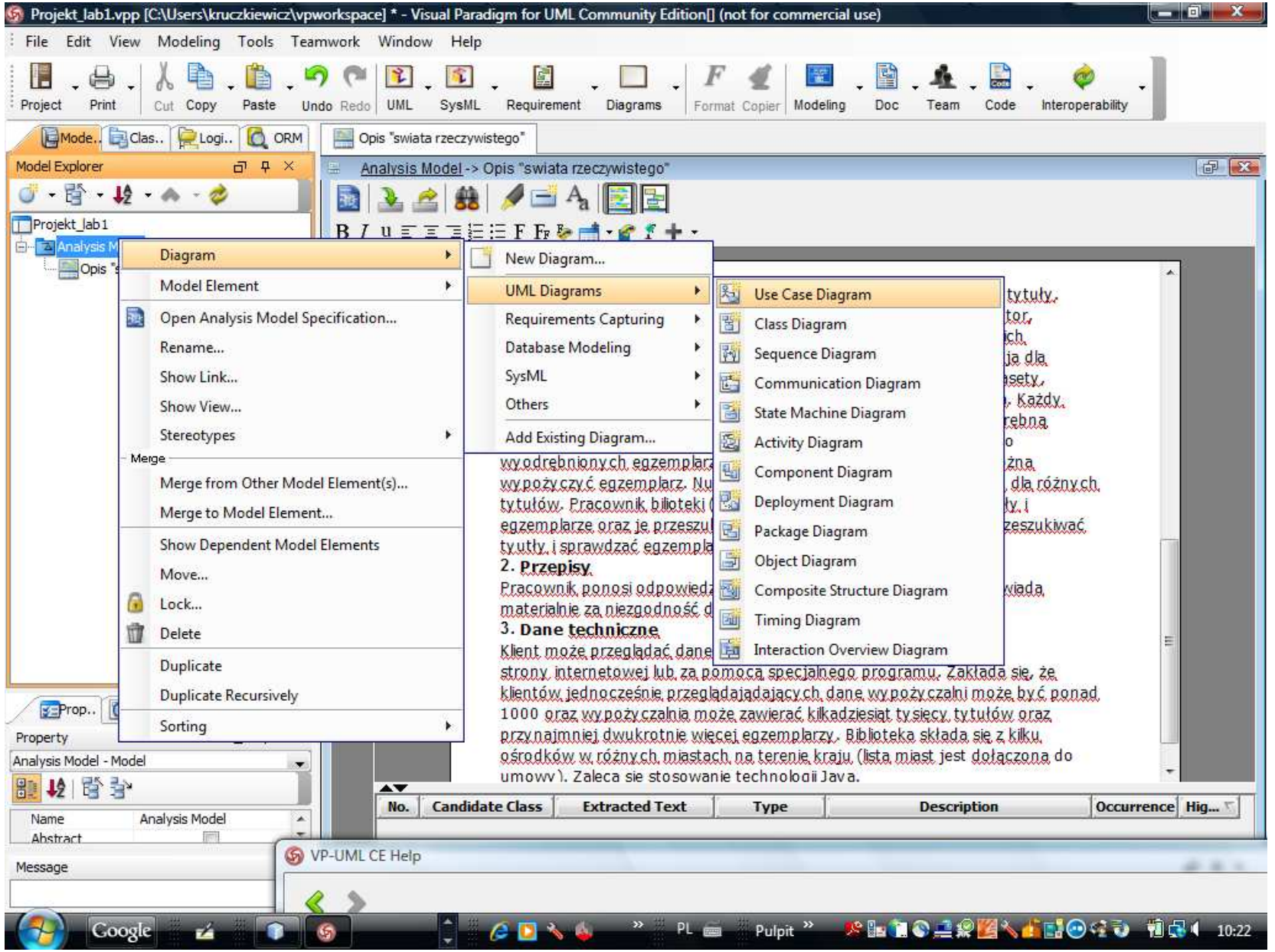
10.3. Należy wykonać opis biznesowy „świata rzeczywistego” – Katalog tytułów i książek w bibliotece

1. **Opis zasobów ludzkich**
   Pracownik wypożyczalni może dodawać do katalogu tytułów nowe tytuły. Każdy tytuł jest reprezentowany przez następujące dane: tytuł, autor, wydawnictwo, ISBN oraz informacje o liczbie egzemplarzy i miejscu ich przechowywania i występuje w bibliotece jako pojedyncza informacja dla każdego tytułu. Pewna grupa tytułów opisuje książki nagrane na kasety, dlatego dodatkowo tytuł zawiera dane nagrania np nazwisko aktora. Każdy egzemplarz, niezależnie, czy jest książką czy kasetą, jest opisany odrębną informacją zawierajacą numer egzemplarza i ewentualnie (dotyczy to wyodrębnionych egzemplarzy) informację o liczbie dni, na które można wypożyczyć egzemplarz. Numery egzemplarzy mogą się powtarzać dla różnych tytułów. Pracownik bilioteki (bibliotekarz) może dodawać nowe tytuły i egzemplarze oraz je przeszukiwać, natomiast klient może jedynie przeszukiwać tyutły i sprawdzać egzemplarze wybranych tytułów.
2. **Przepisy**
   Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem wypożyczalni.
3. **Dane techniczne**
   Klient może przeglądać dane wypożyczalni za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu. Zakłada się, że klientów jednocześnie przeglądającdających dane wypożyczalni może być ponad 1000 oraz wypożyczalnia może zawierać kilkadziesiąt tysięcy tytułów oraz przynajmniej dwukrotnie więcej egzemplarzy. Biblioteka składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

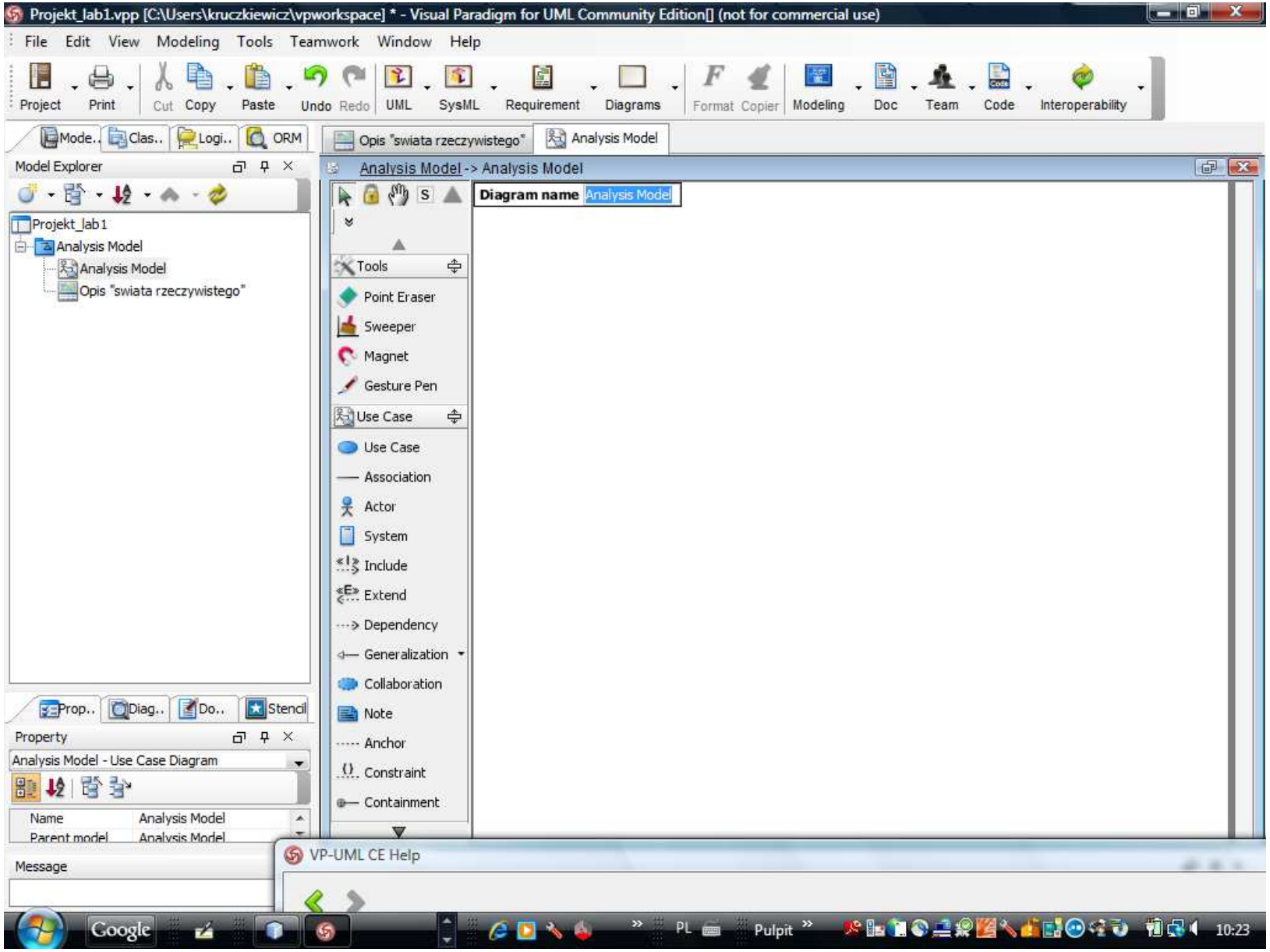
10.4. Okno diagramu – wykonany opis



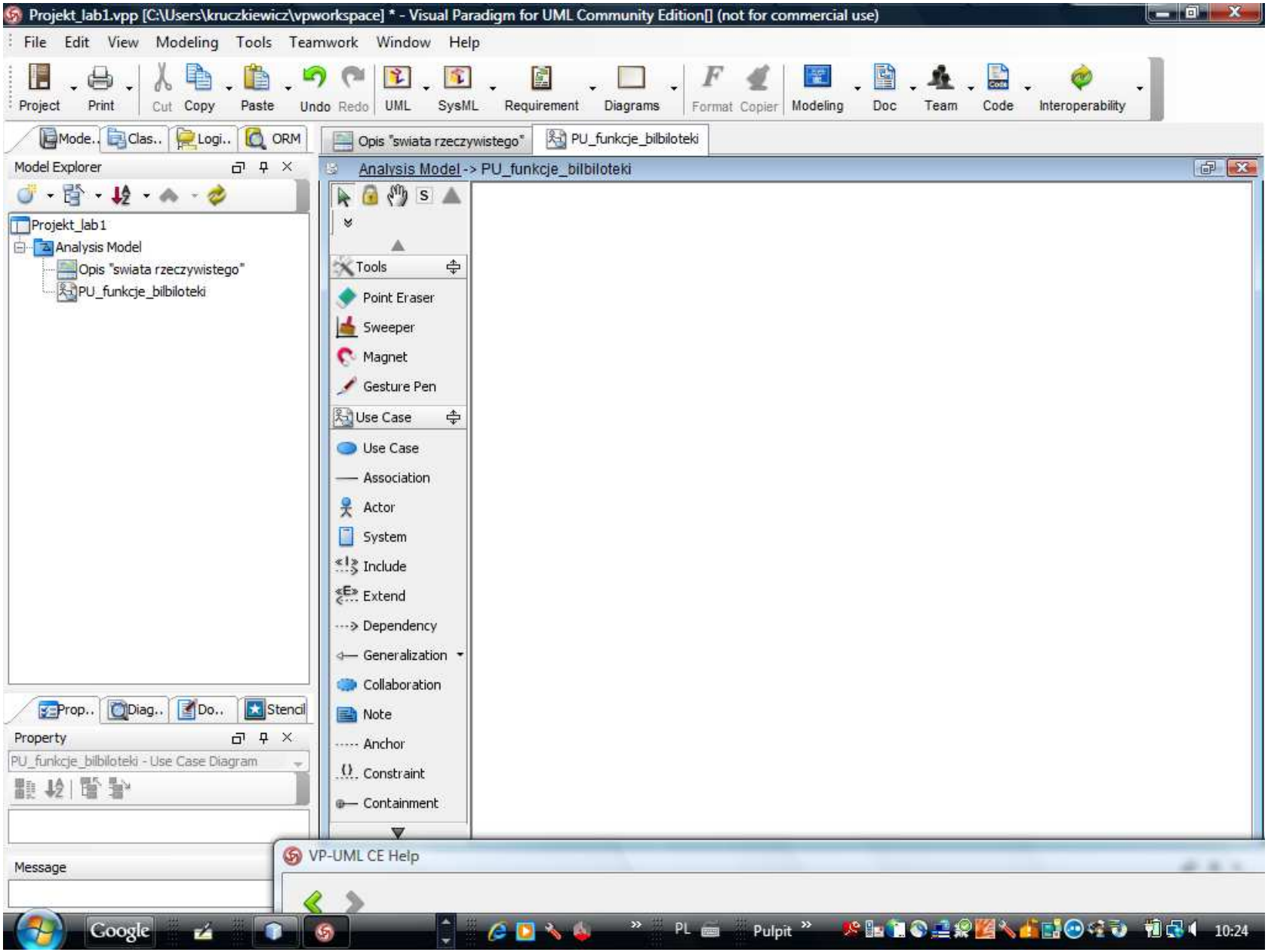
11.1. Wstawianie do projektu diagramu typu *Use Case* – po kliknięciu na nazwę modelu wybór z list: *Diagram/UML Diagram/ Use Case Diagram*



11.2. Wstawianie do projektu diagramu typu *Use Case* – nadanie nazwy diagramowi typu *Use Case*



11.3. Wstawianie do projektu diagramu typu *Use Case* – nadanie nazwy diagramowi typu *Use Case* np..**PU_funkcje_biblioteki**



12. Należy zdefiniować wymagania aplikacji

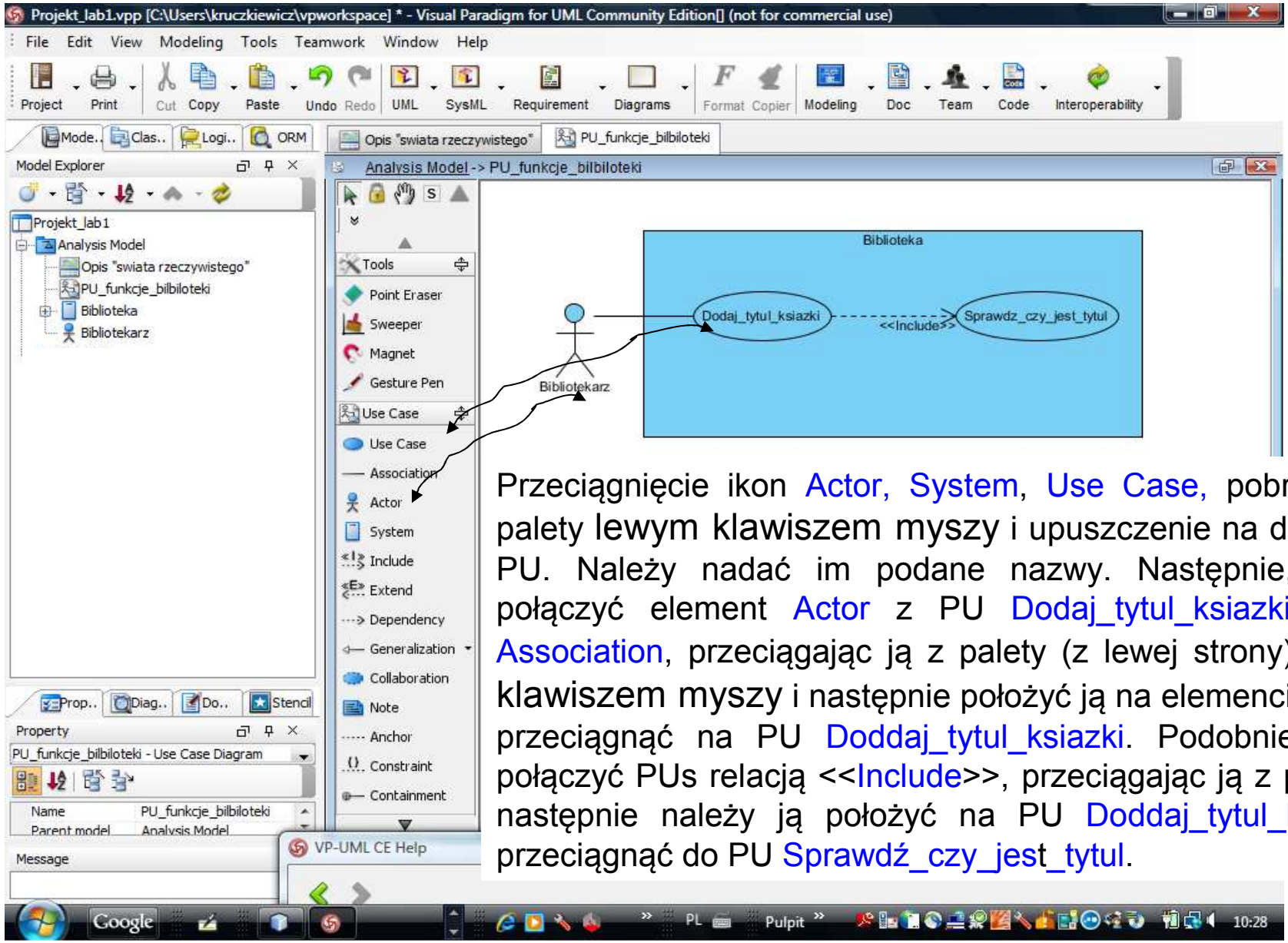
Wymagania funkcjonalne

- Biblioteka wypożycza podane książki i czasopisma osobom zarejestrowanym, o ile je posiada
- Biblioteka dokonuje zakupu nowych książek, przy czym popularne książki kupuje w kilku egzemplarzach. Usuwa zniszczone książki i czasopisma.
- Bibliotekarz jest pracownikiem biblioteki, komunikuje się z wypożyczającym. Jego praca jest wspierana za pomocą systemu
- 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.
- Biblioteka może łatwo utworzyć, zmienić i usunąć informację o tytułach, wypożyczających, wypożyczeniach i rezerwacjach

Wymagania niefunkcjonalne

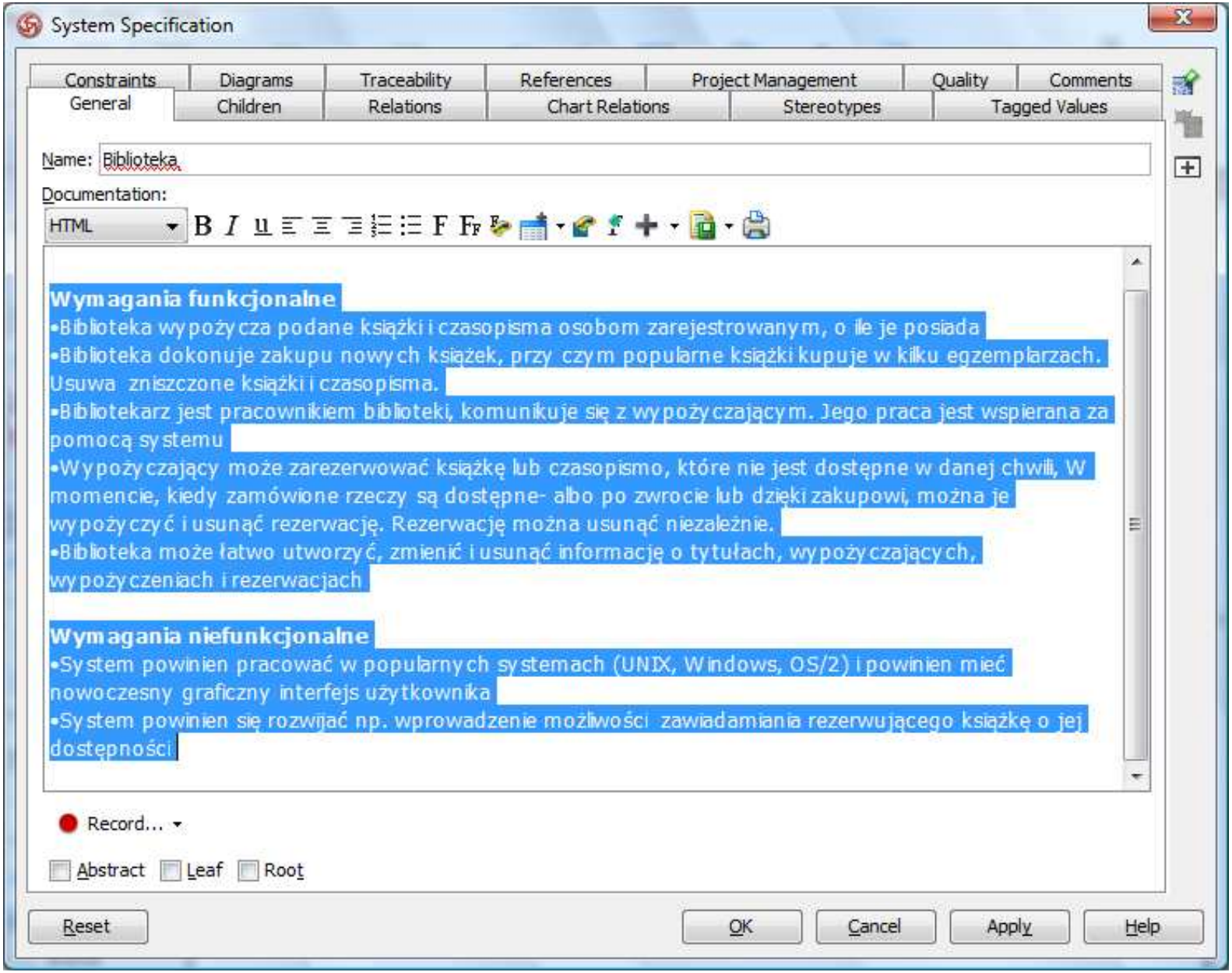
- System powinien pracować w popularnych systemach (UNIX, Windows, OS/2) i powinien mieć nowoczesny graficzny interfejs użytkownika
- System powinien się rozwijać np. wprowadzenie możliwości zawiadamiania rezerwującego książkę o jej dostępności

13. Wstawianie do projektu diagramu typu *Use Case* – definicja przypadków użycia specyfikujących wymagania stawiane aplikacji (p.12) w zakresie wstawiania zasobów biblioteki

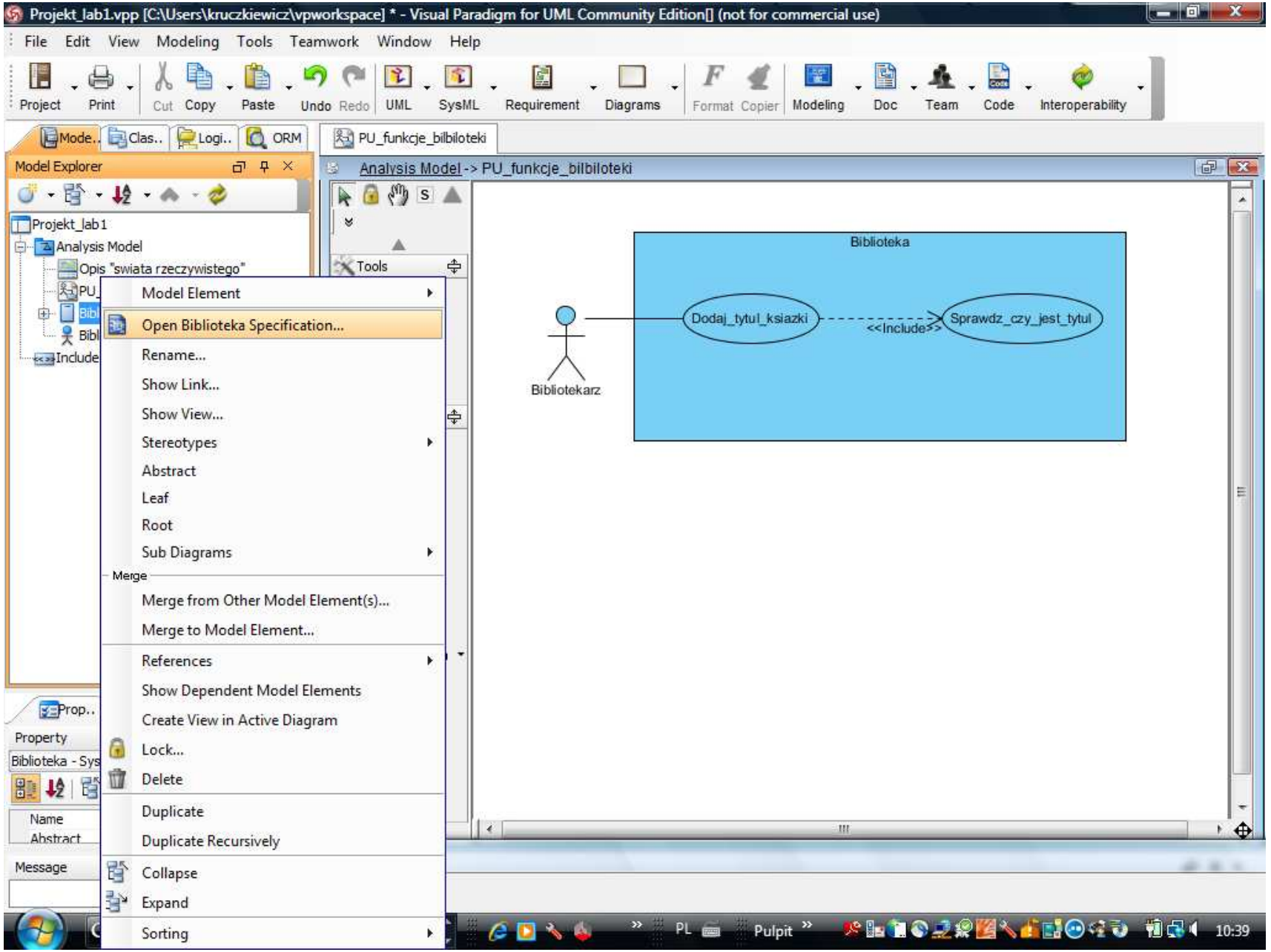


Przeciągnięcie ikon Actor, System, Use Case, pobranych z palety lewym klawiszem myszy i upuszczenie na diagramie PU. Należy nadać im podane nazwy. Następnie, należy połączyć element Actor z PU Dodaj_tytul_ksiazki relacją Association, przeciągając ją z palety (z lewej strony) lewym klawiszem myszy i następnie położyć ją na elemencie Actor i przeciągnąć na PU Doddaj_tytul_ksiazki. Podobnie należy połączyć PUs relacją <<Include>>, przeciągając ją z palety - i następnie należy ją położyć na PU Doddaj_tytul_ksiazki i przeciągnąć do PU Sprawdź_czy_jest_tytul.

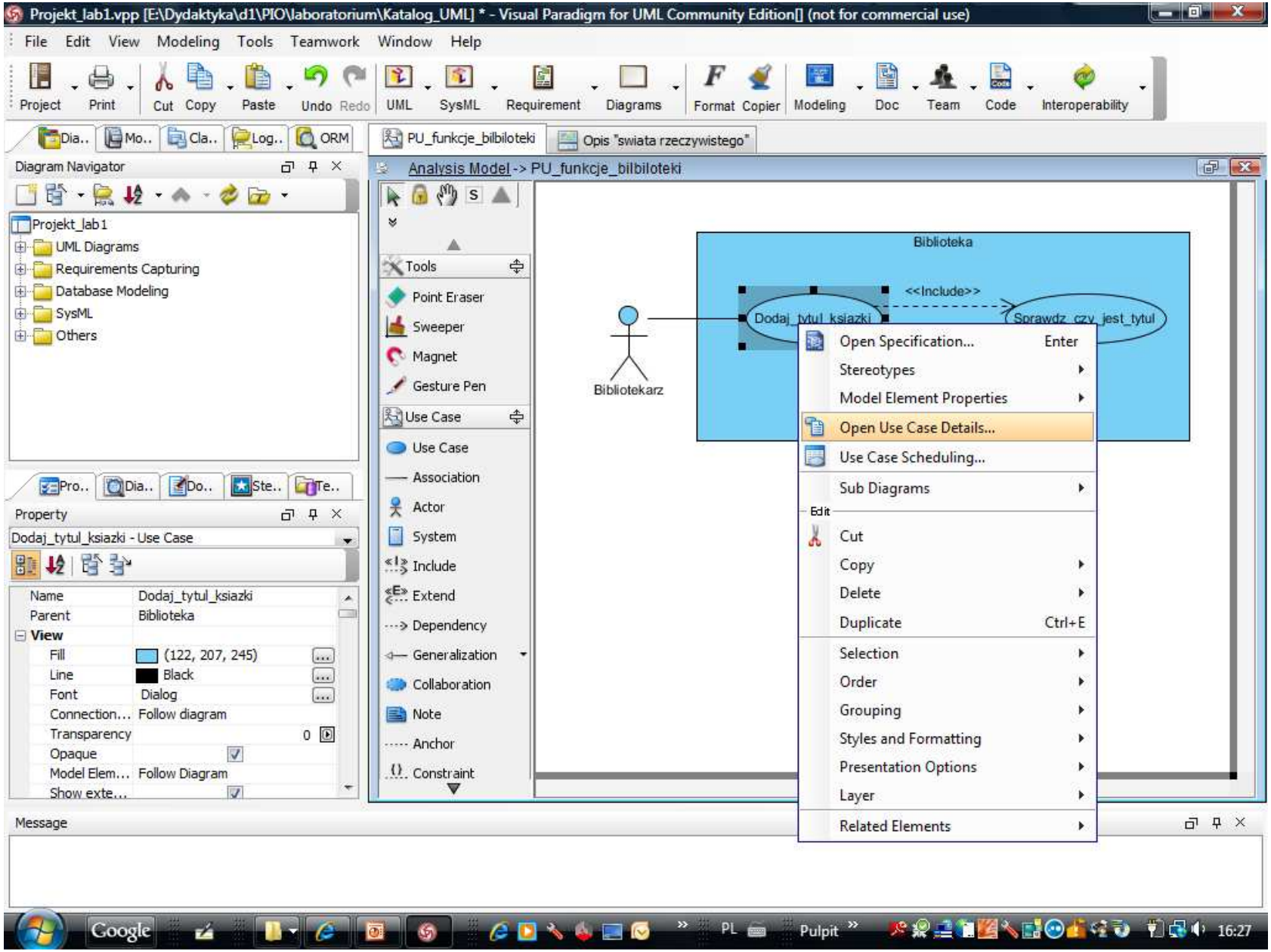
14.1. Po kliknięciu prawym klawiszem na element *System* (albo na diagramie albo w okienku *Model Explorer* – następny slajd), należy wybrać z listy *Open Specification*. Nastepnie, należy wprowadzić tekst wymagań stawianych aplikacji (p. 12)



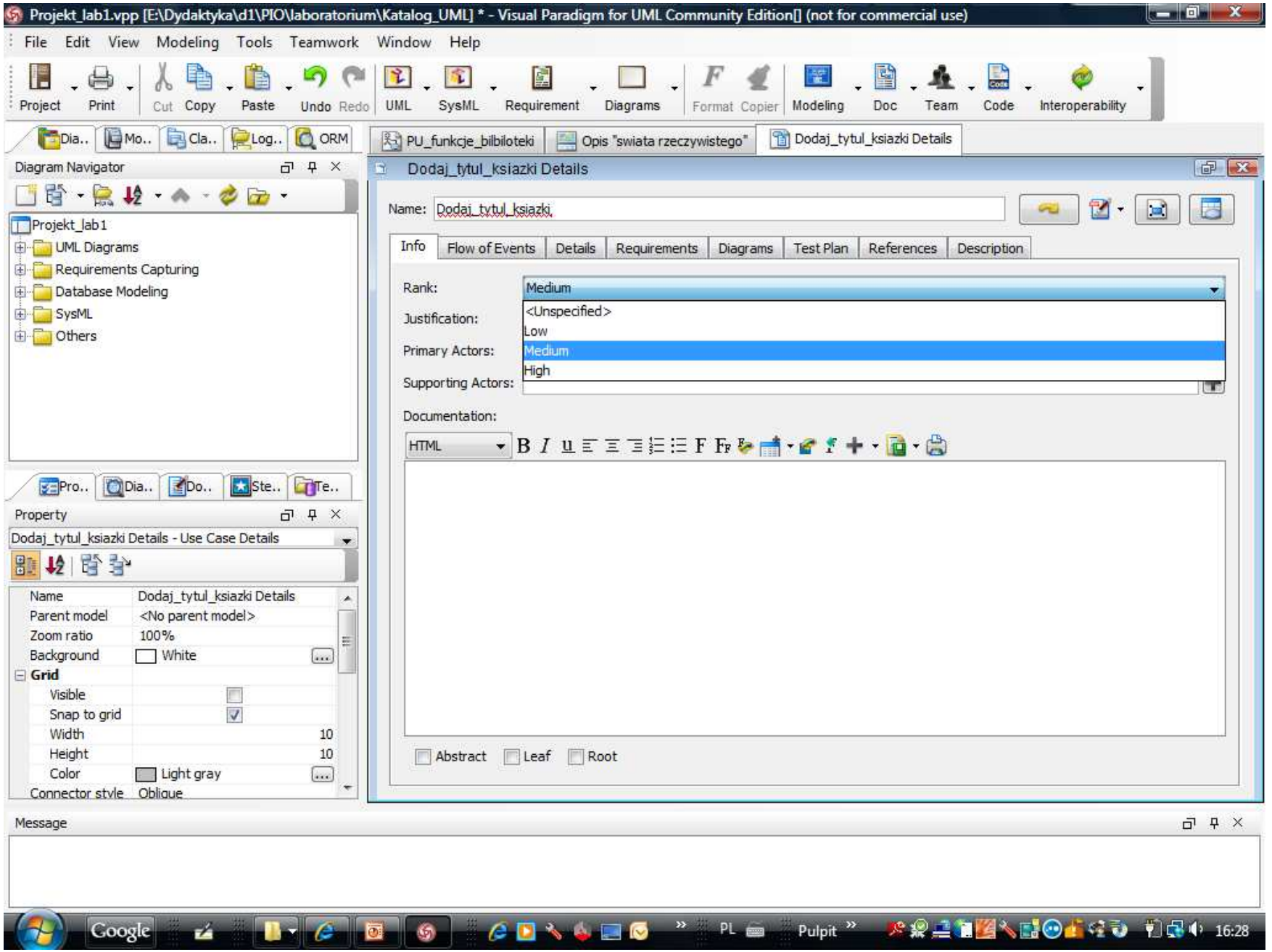
14.2. Wybór okna specyfikacji elementu *System* po kliknięciu prawym klawiszem na nazwę *System* w okienku *Model Explorer*



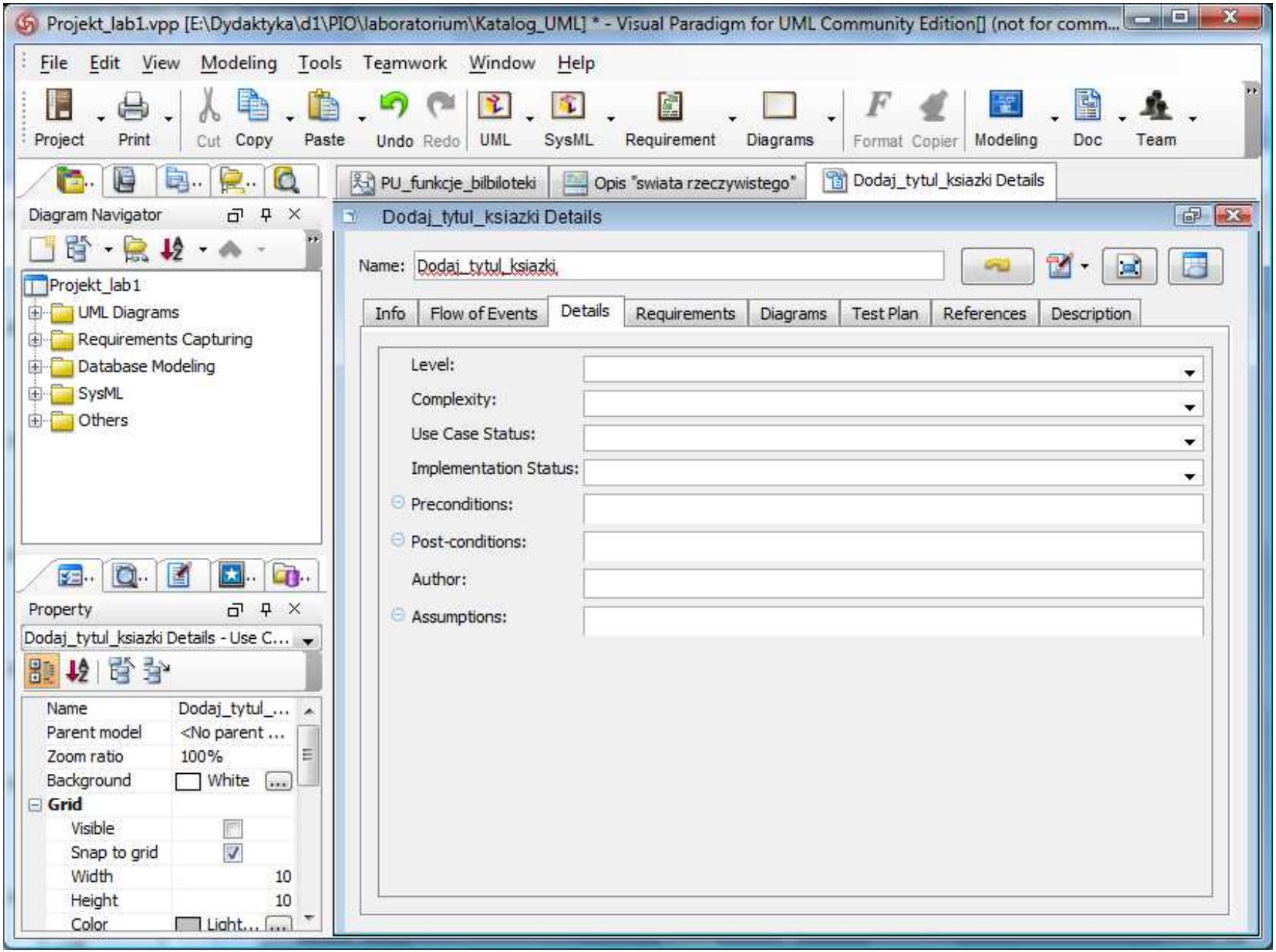
15.1. Definiowanie elementów typu *Use Case* – po kliknięciu prawym klawiszem myszy na PU *Dodaj_tytul_ksiazki* należy dokonać wyboru z listy opcji *Open Use Case Details*



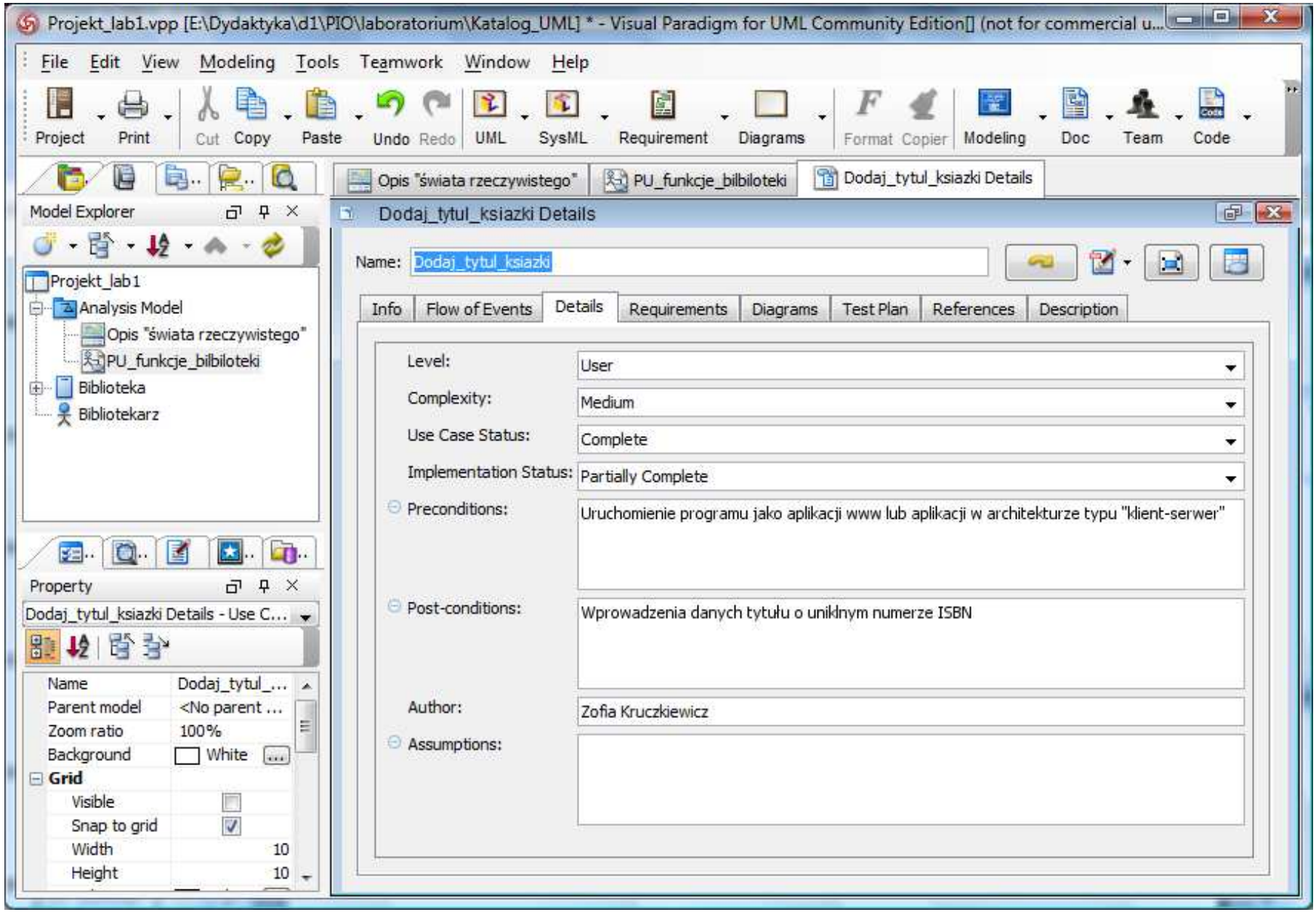
15.2. Nadanie wagi diagramowi – wybór wartości z listy w polu *Rank*



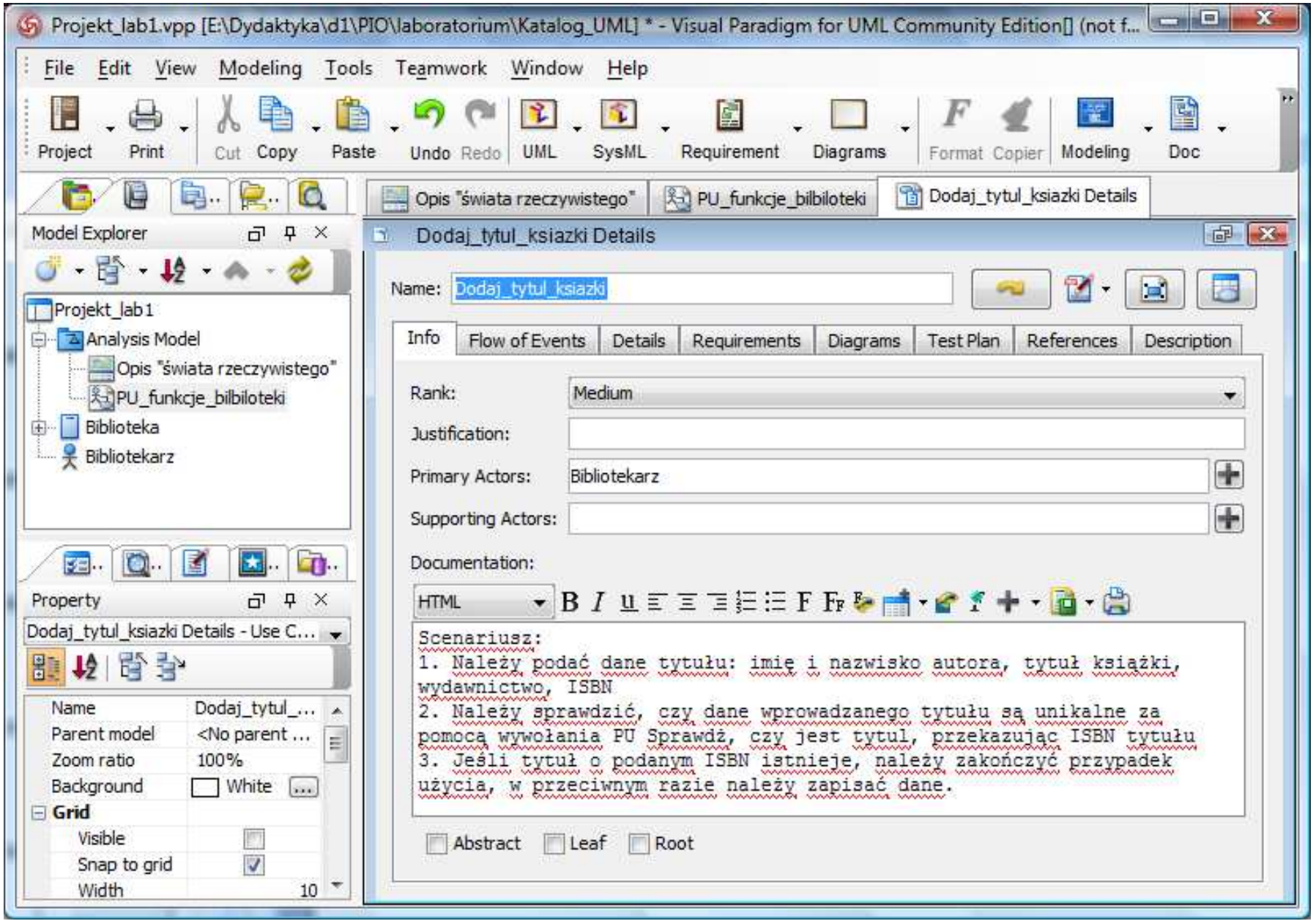
15.3. Wybór podformularza *Details* związanego z wybranym wcześniej PU



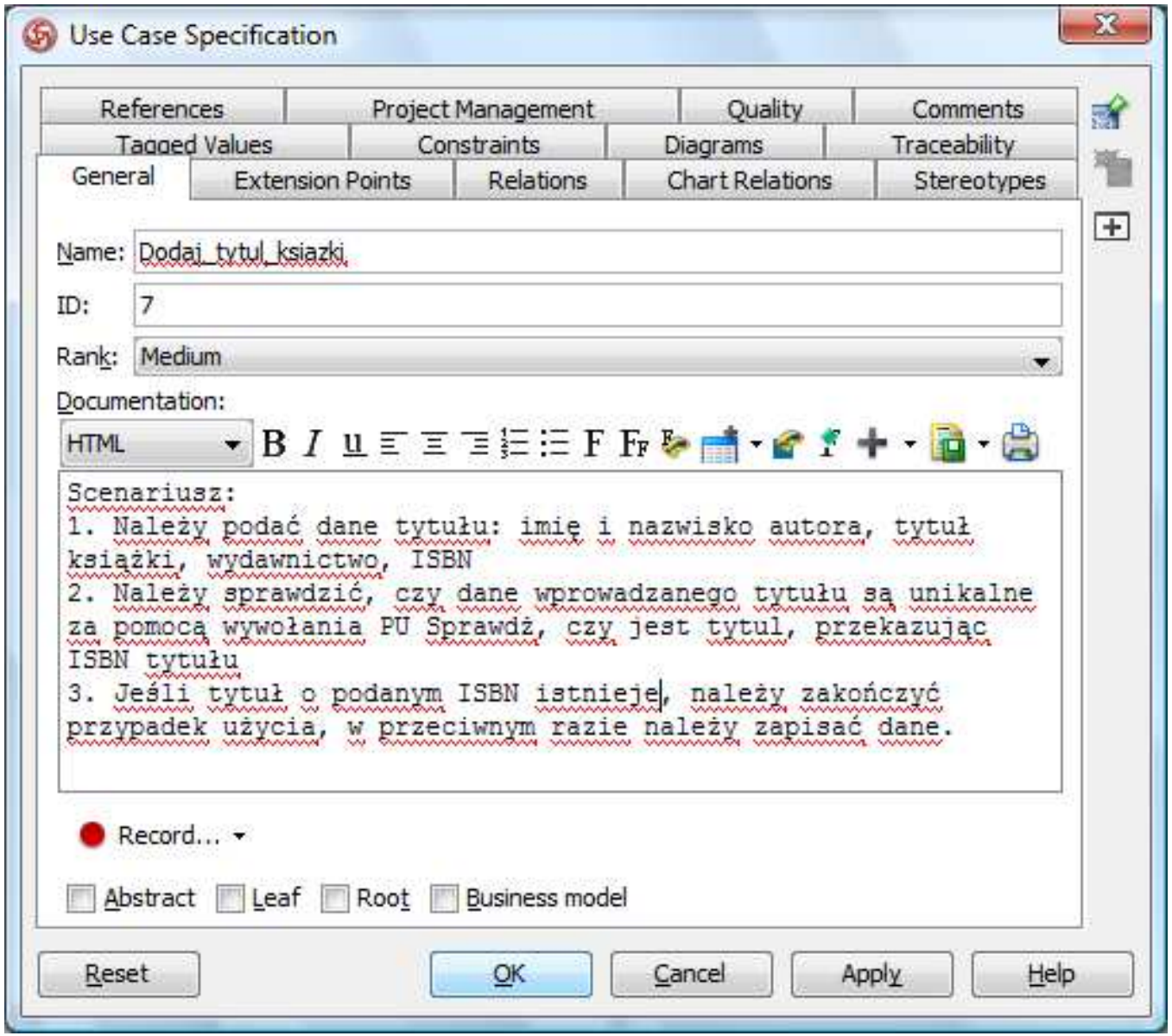
15.4. Wybór podformularza *Details* związanego z wybranym wcześniej PU – nadanie wartości poszczególnym polom formularza przez wybór z listy lub wprowadzenie tekstu



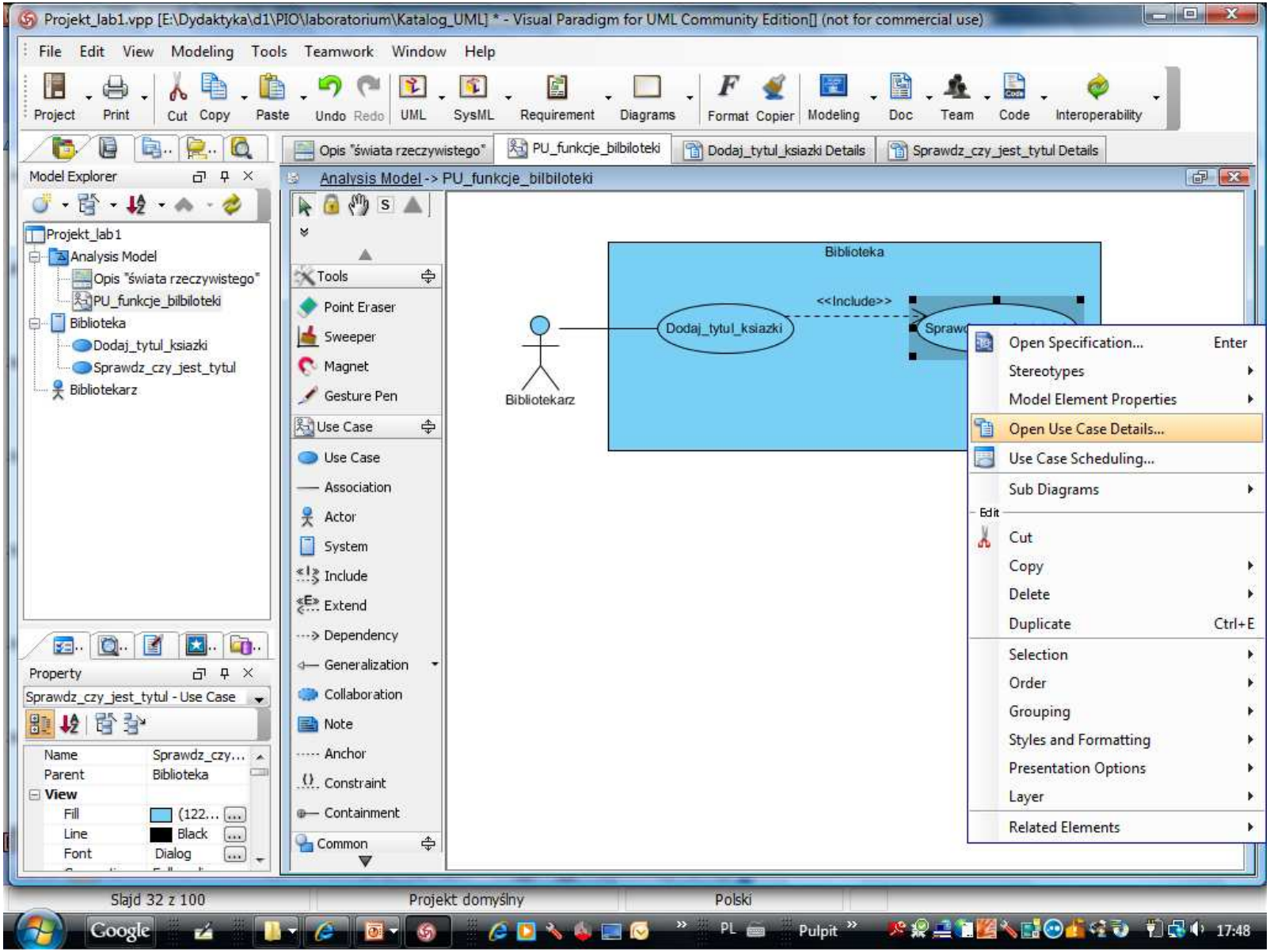
15.5. Wpisanie do podformularza *Info* w części *Documentation* scenariusza przypadku użycia *Dodaj_tytul*



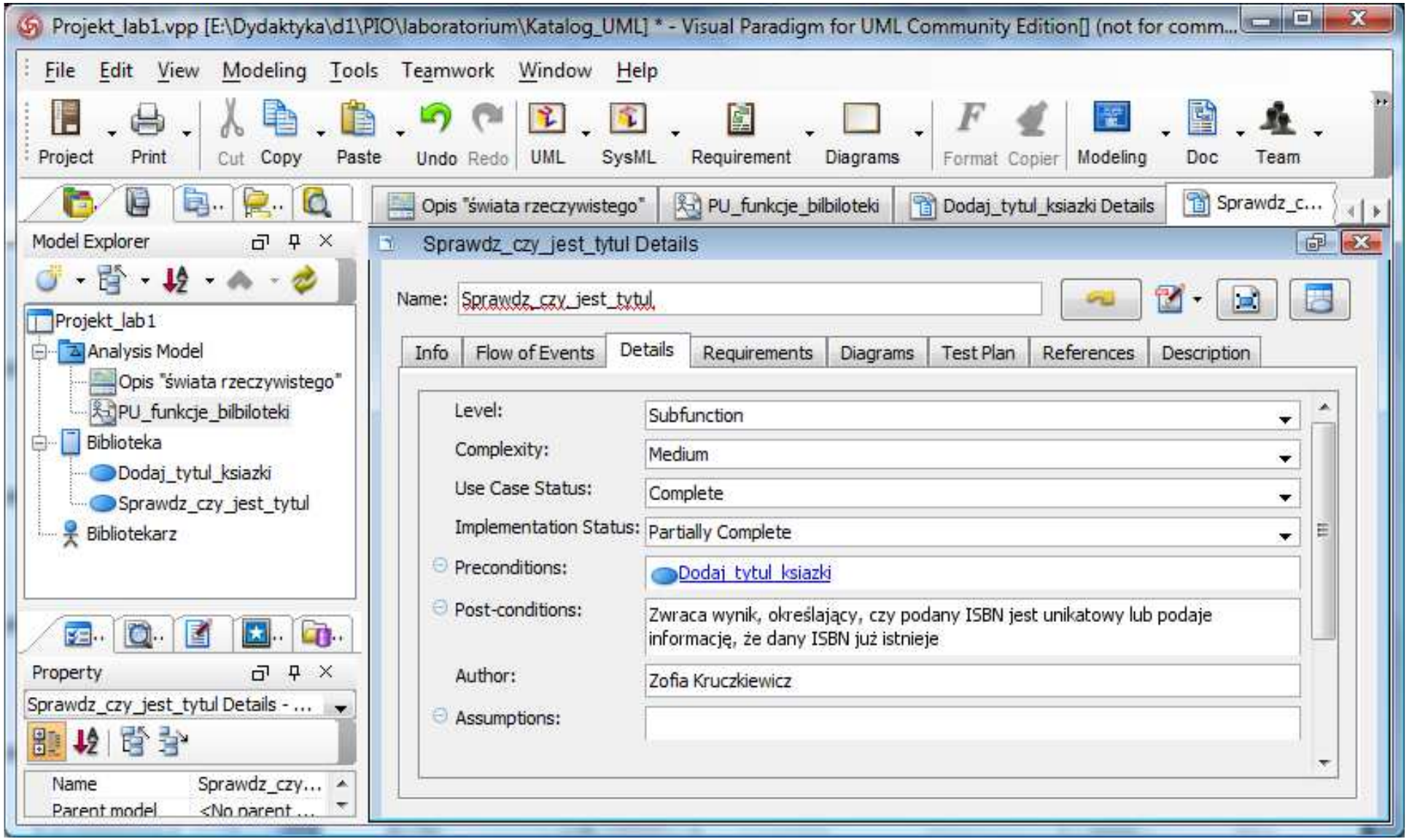
15.6. Po kliknięciu prawym klawiszem myszy na PU *Dodaj_tytul_ksiazki* należy należy wybrać z listy opcję *Open Specification* – powinien ukazać tekst wprowadzony wcześniej w podformularzu *Info* opcji *Open Use Case Details*



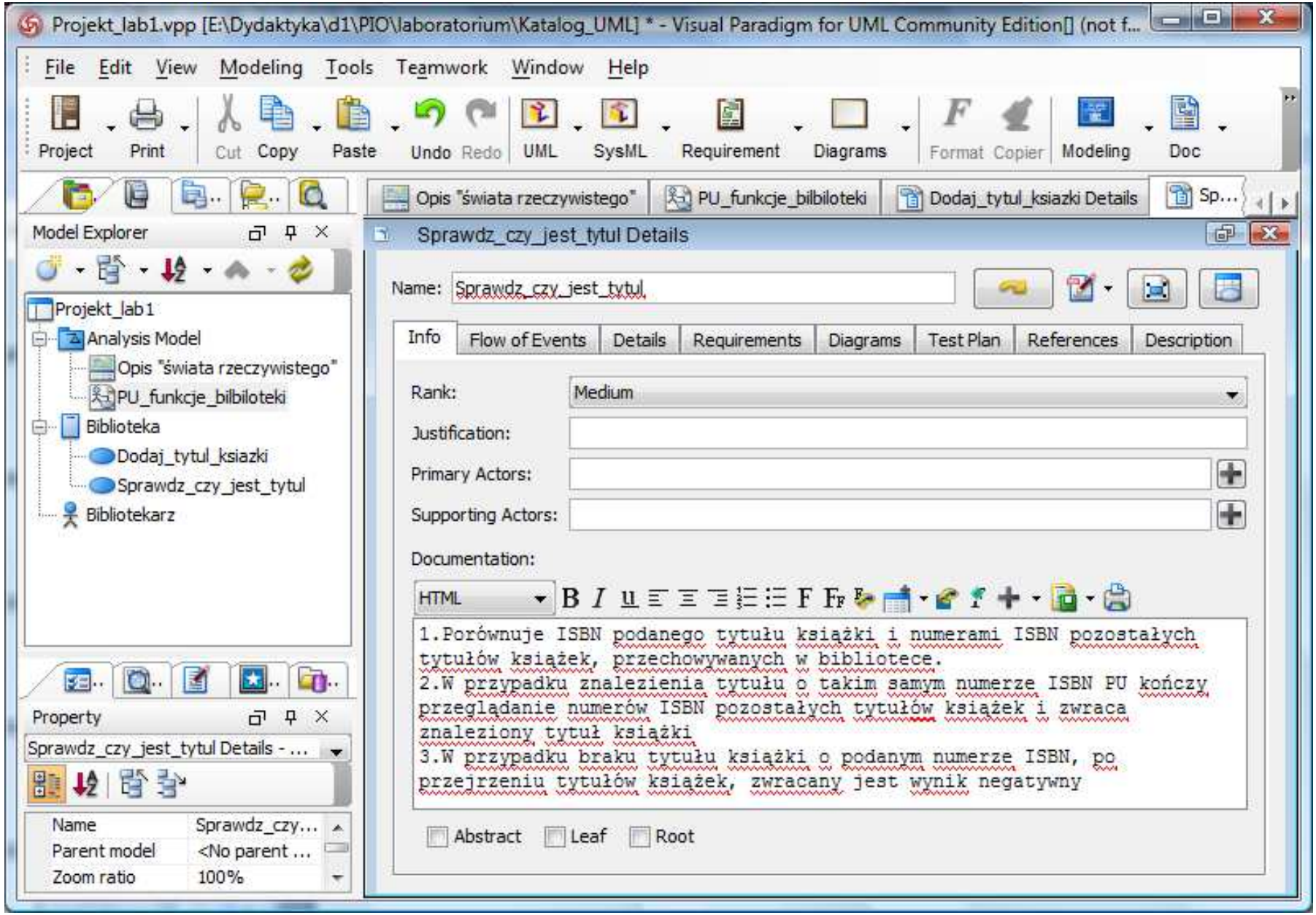
15.7. Definiowanie elementów typu Use Case – po kliknięciu prawym klawiszem myszy na PU *Sprawdz_czy_jest_tytul* należy dokonać wyboru z listy opcji *Open Use Case Details*



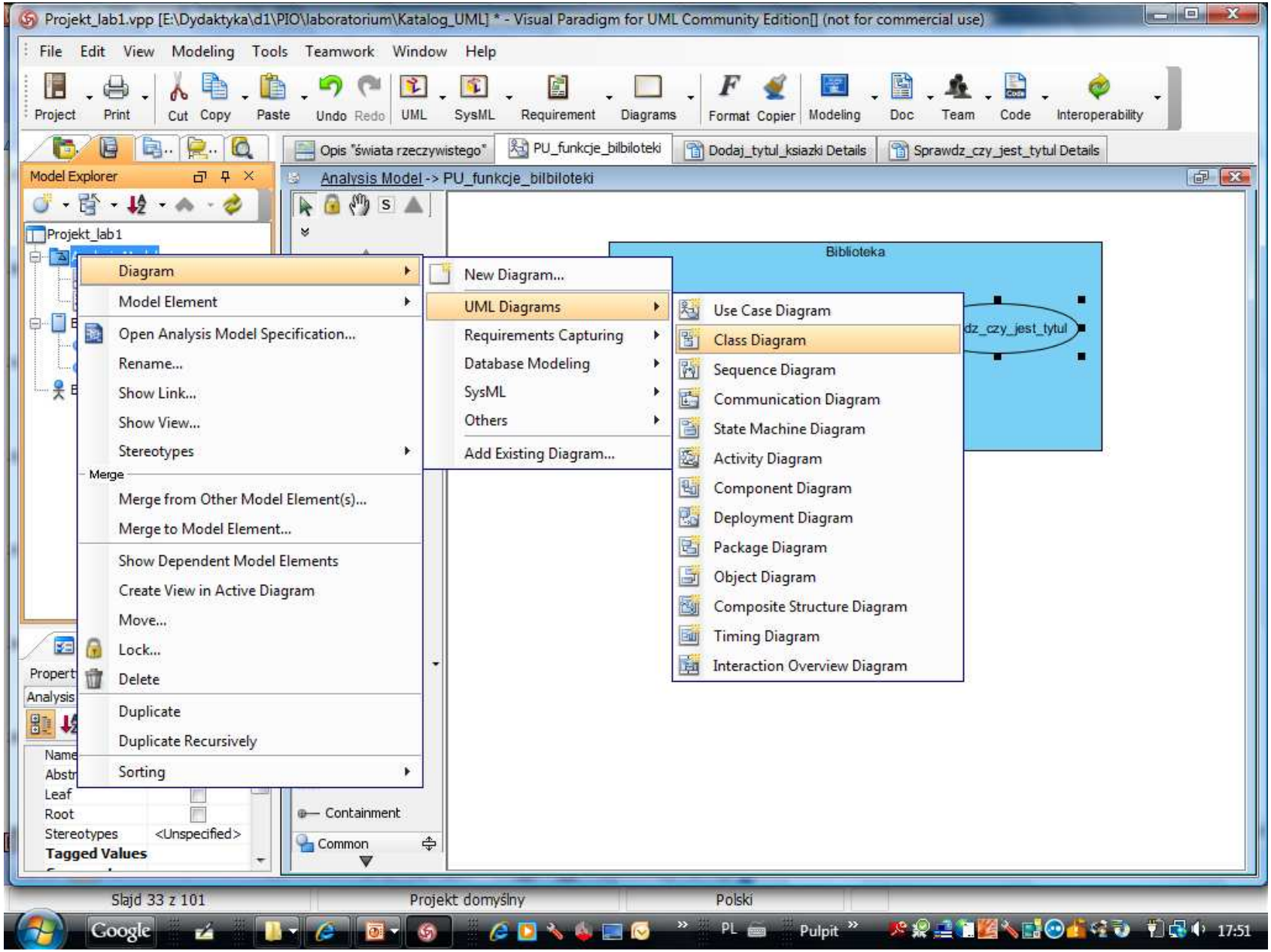
15.8. Wybór podformularza *Details* związanego z wybranym wcześniej PU – nadanie wartości poszczególnym polom formularza przez wybór z listy lub wprowadzenie tekstu



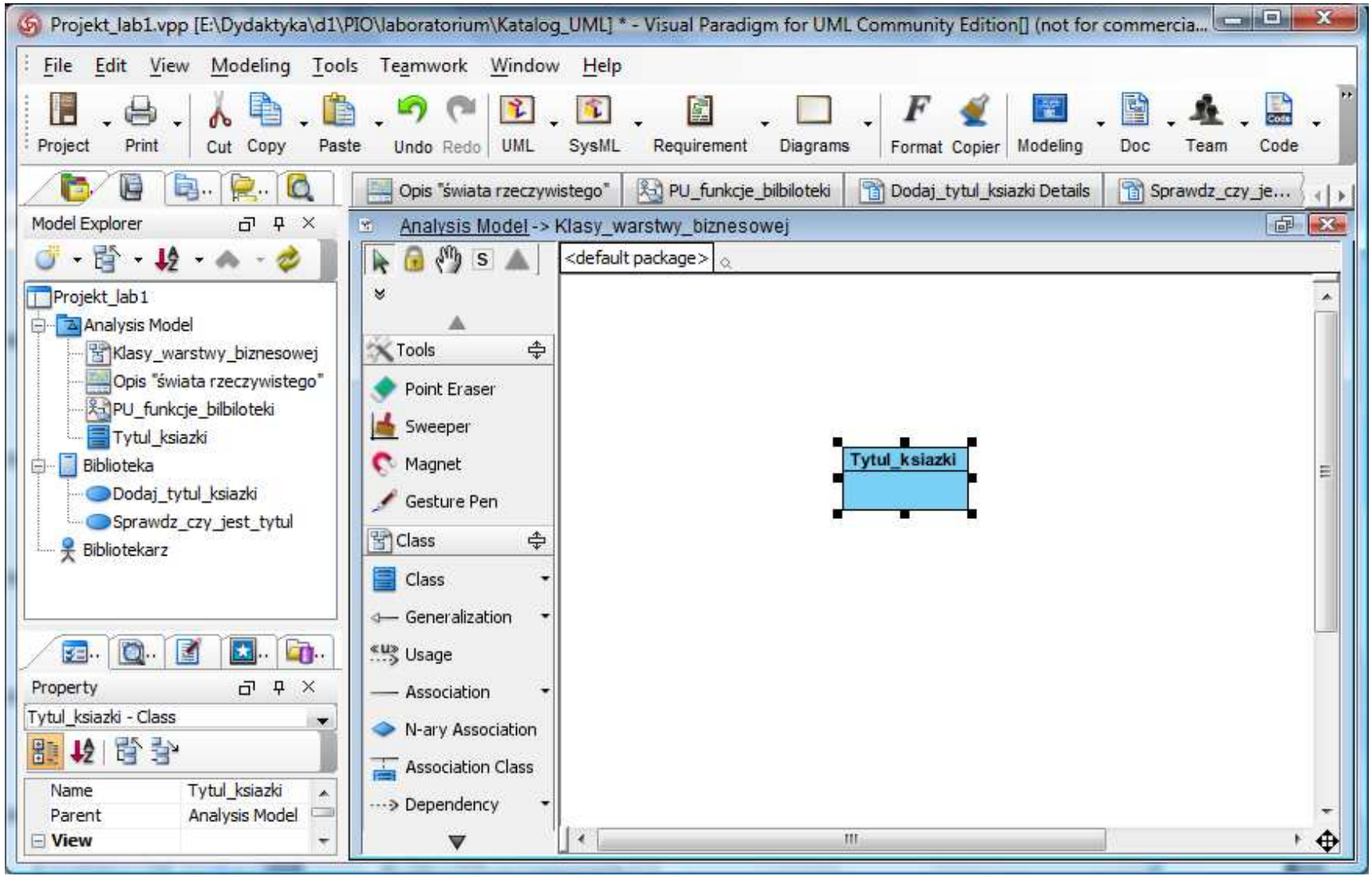
15.9. Wpisanie do podformularza *Info* w części *Documentation* scenariusza przypadku użycia PU *Sprawdz_czy_jest_tytul*



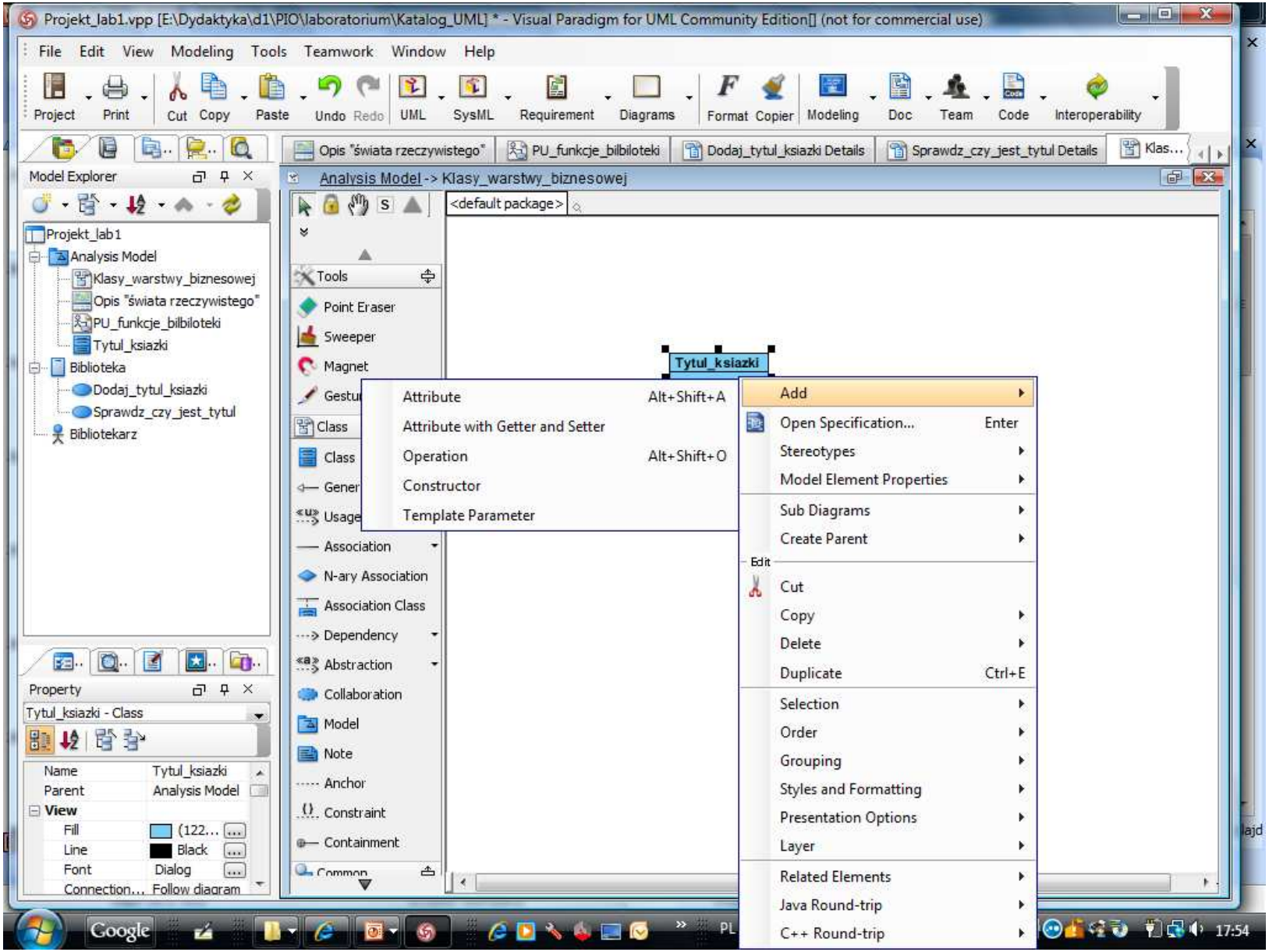
16. Dodanie diagramu klas do projektu – należy kliknąć prawym klawiszem na nazwe modelu w okienku *Model Explorer* i wybrać z list *Diagram/UML Diagram/Class Diagarm*



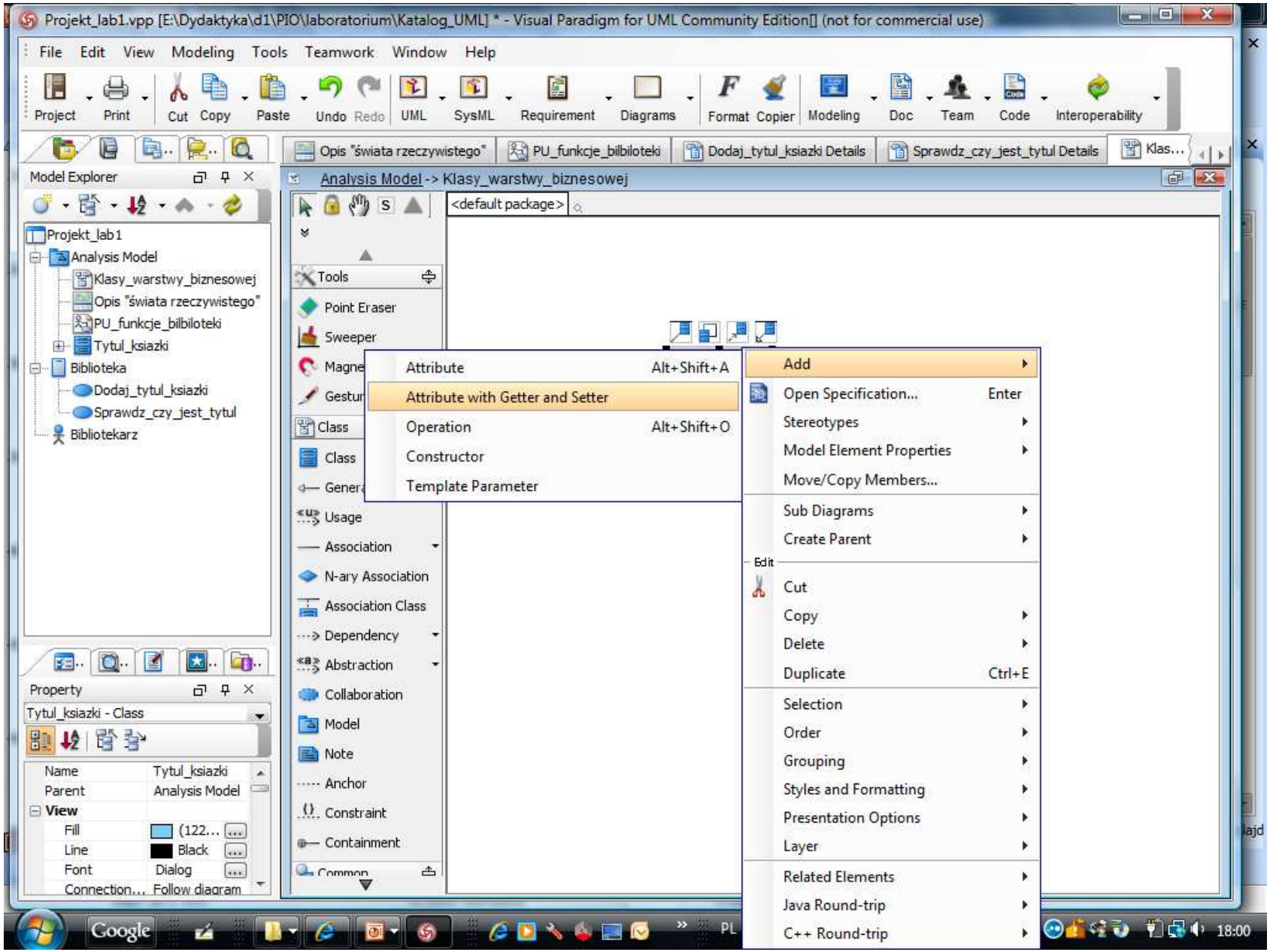
17.1. Po nadaniu nazwy diagramowi klas warstwy biznesowej jako **Klasy_warstwy_biznesowej** należy zdefiniować klasy zidentyfikowane na podstawie scenariuszy przypadków użycia. Pierwsza definiowana klasa zawiera dane tytułu książki – należy przeciągnąć ikonę klasy z palety lewym klawiszem myszy i położyć na diagramie i nadać jej nazwę **Tytul_ksiazki**



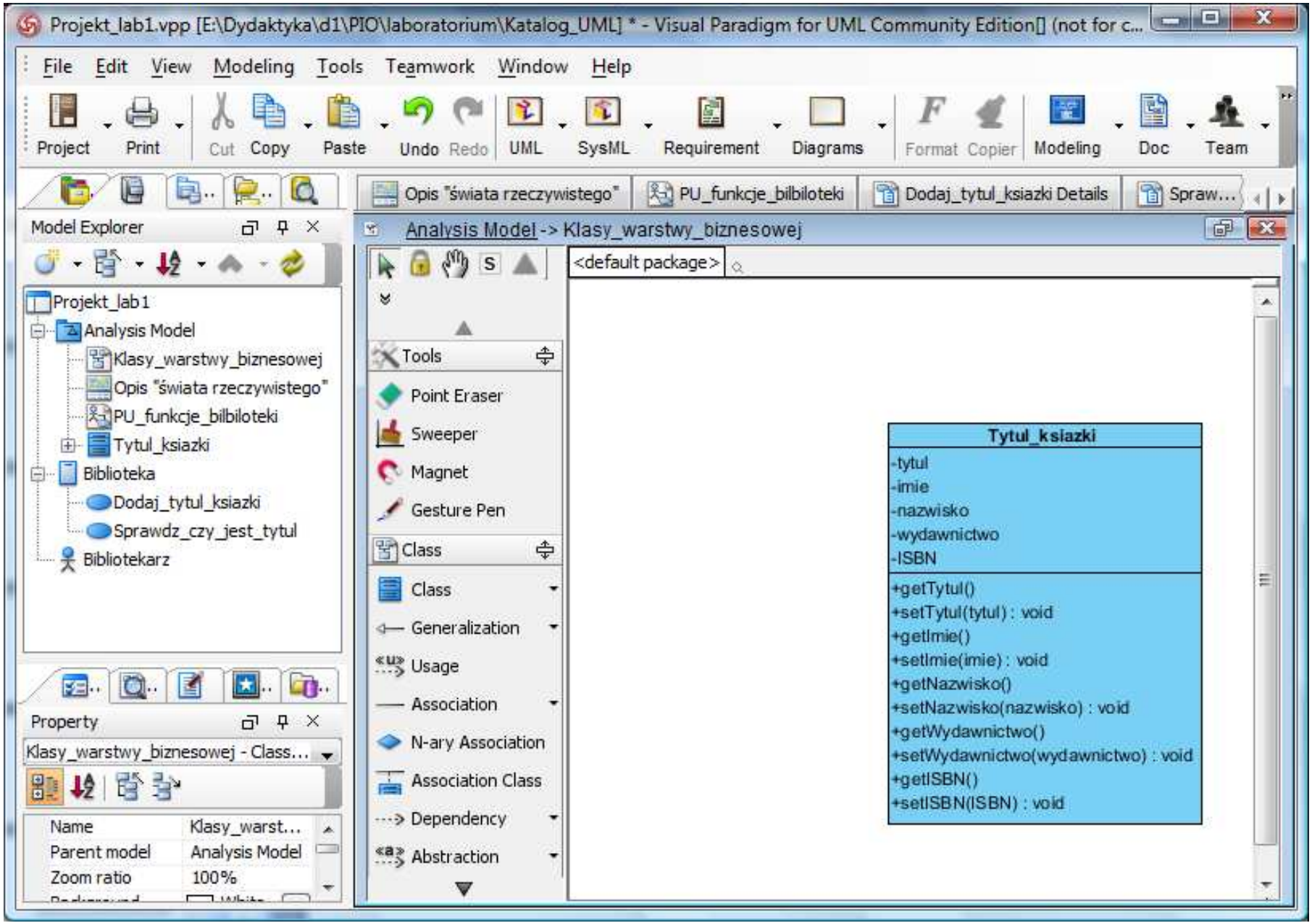
17.2. Zdefiniowanie atrybutów i metod – po kliknięciu prawym klawiszem na klasę należy wybrać z listy pozycje *Attribute* do definiowania nowych atrybutów lub *Operation* do definiowania metod



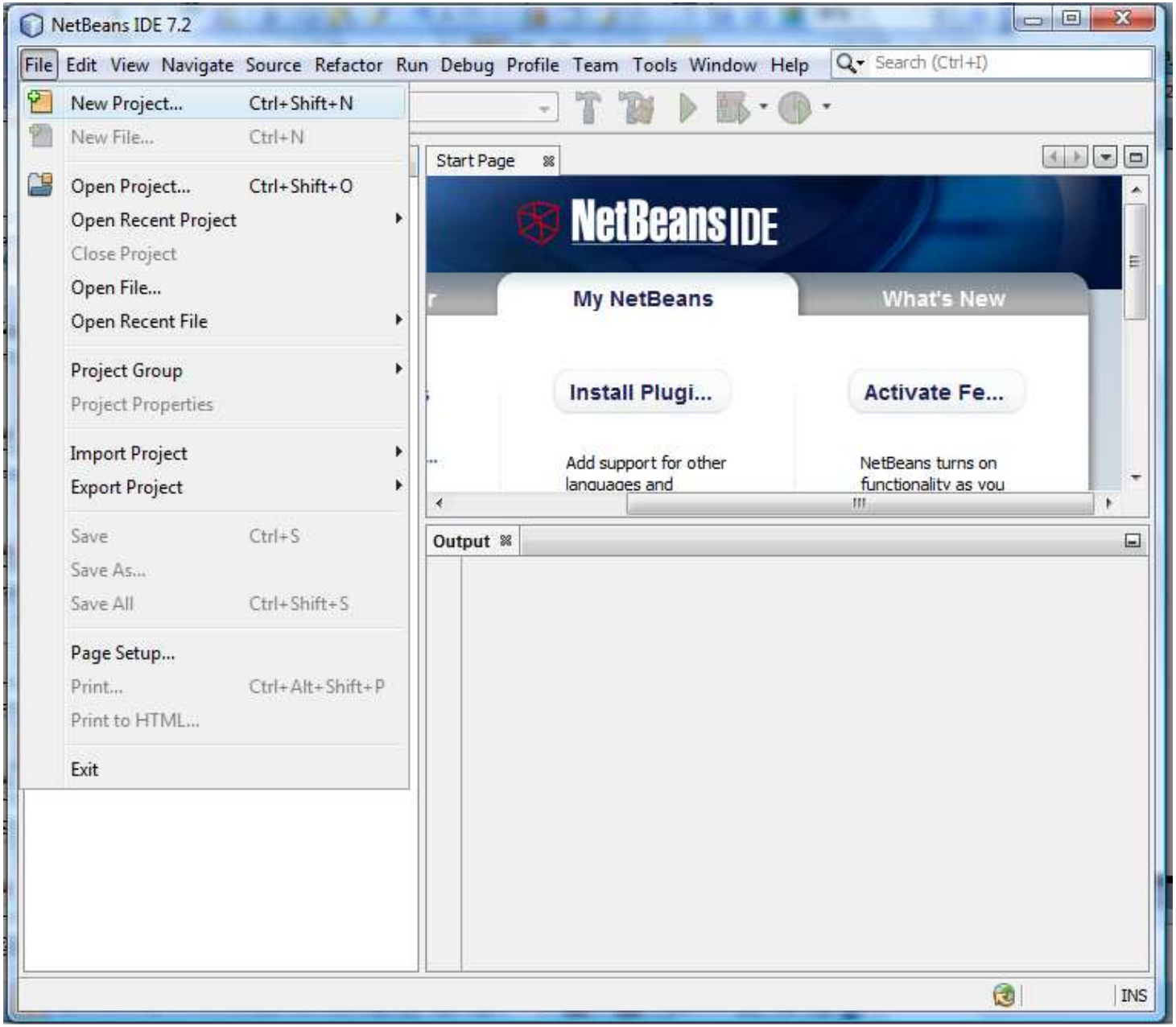
17.3. Zdefiniowanie atrybutów i metod dostępu do atrybutów – po kliknięciu prawym klawiszem na klasę należy wybrać z listy pozycję *Attribute with Getter and Setter*



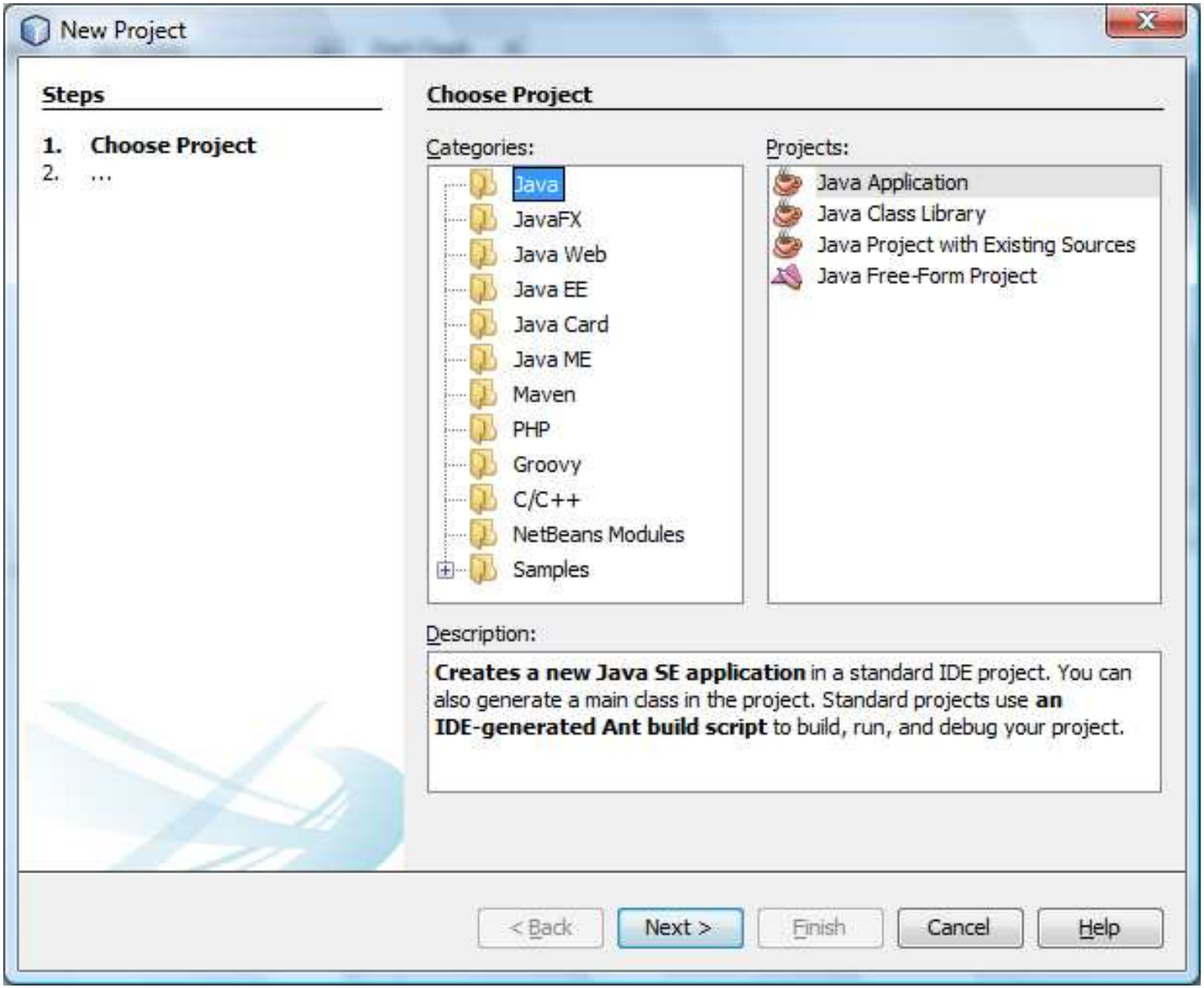
17.4. Dodano prywatne atrybuty i publiczne metody dostępu do atrybutów typu get i set klasie **Tytul_ksiazk**i



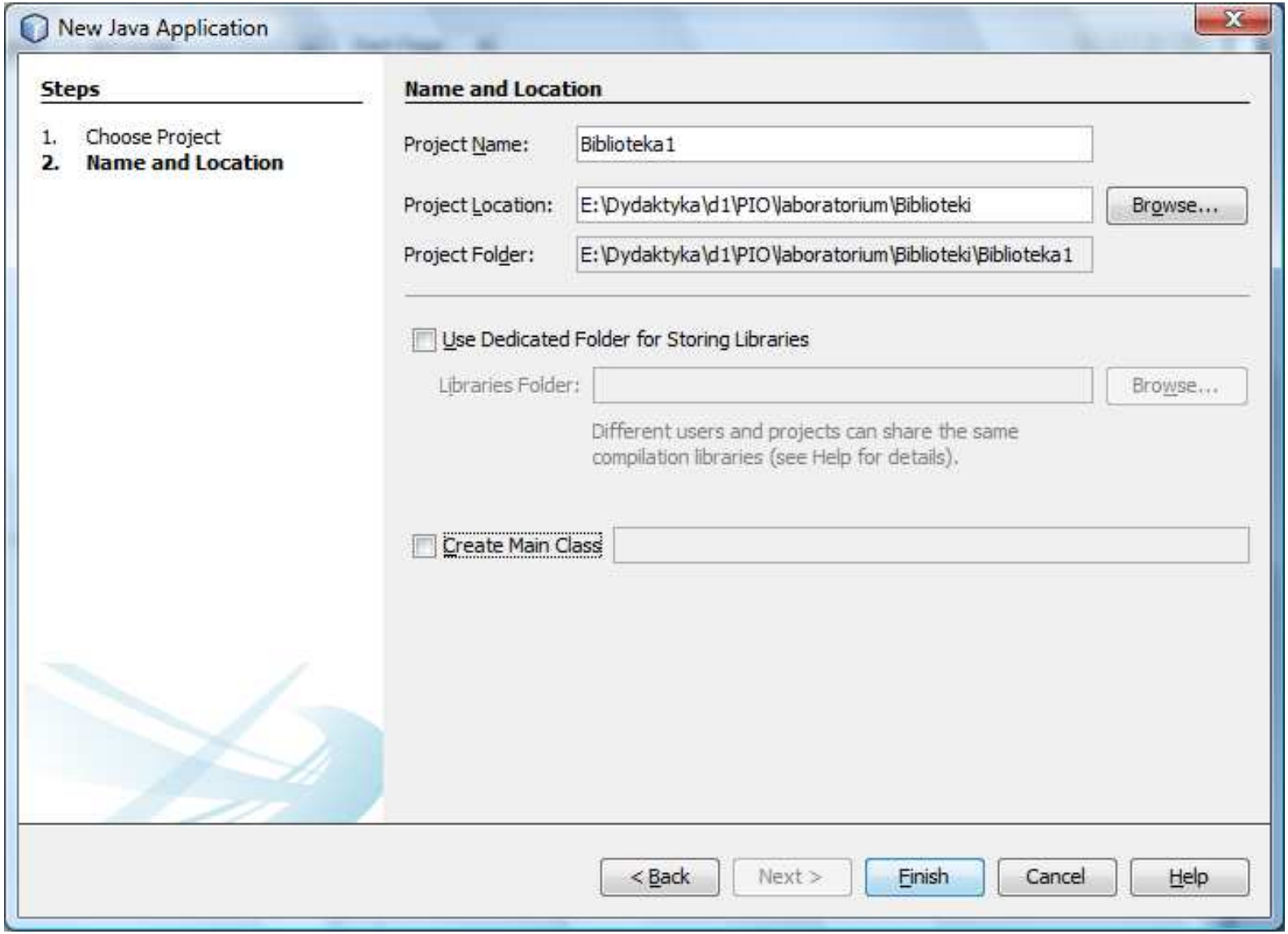
18.1. Wykonanie projektu typu aplikacja Javy w środowisku typu NetBeans – *File/New Project*



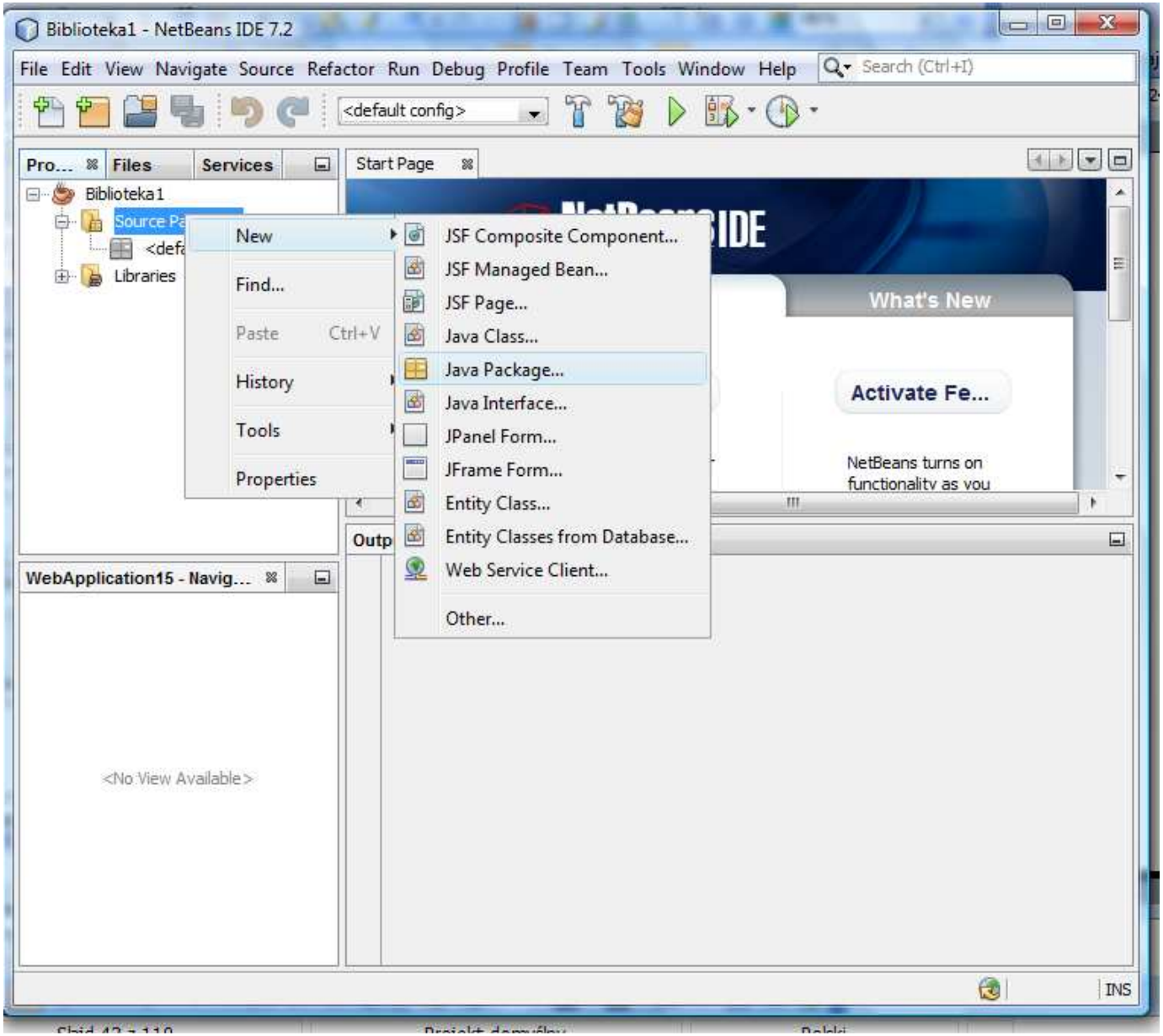
18.2. Wykonanie projektu typu *Java/ Java Application* – wybór w kolumnie *Categories* pozycji Java oraz pozycji *Java Application* w kolumnie *Projects*



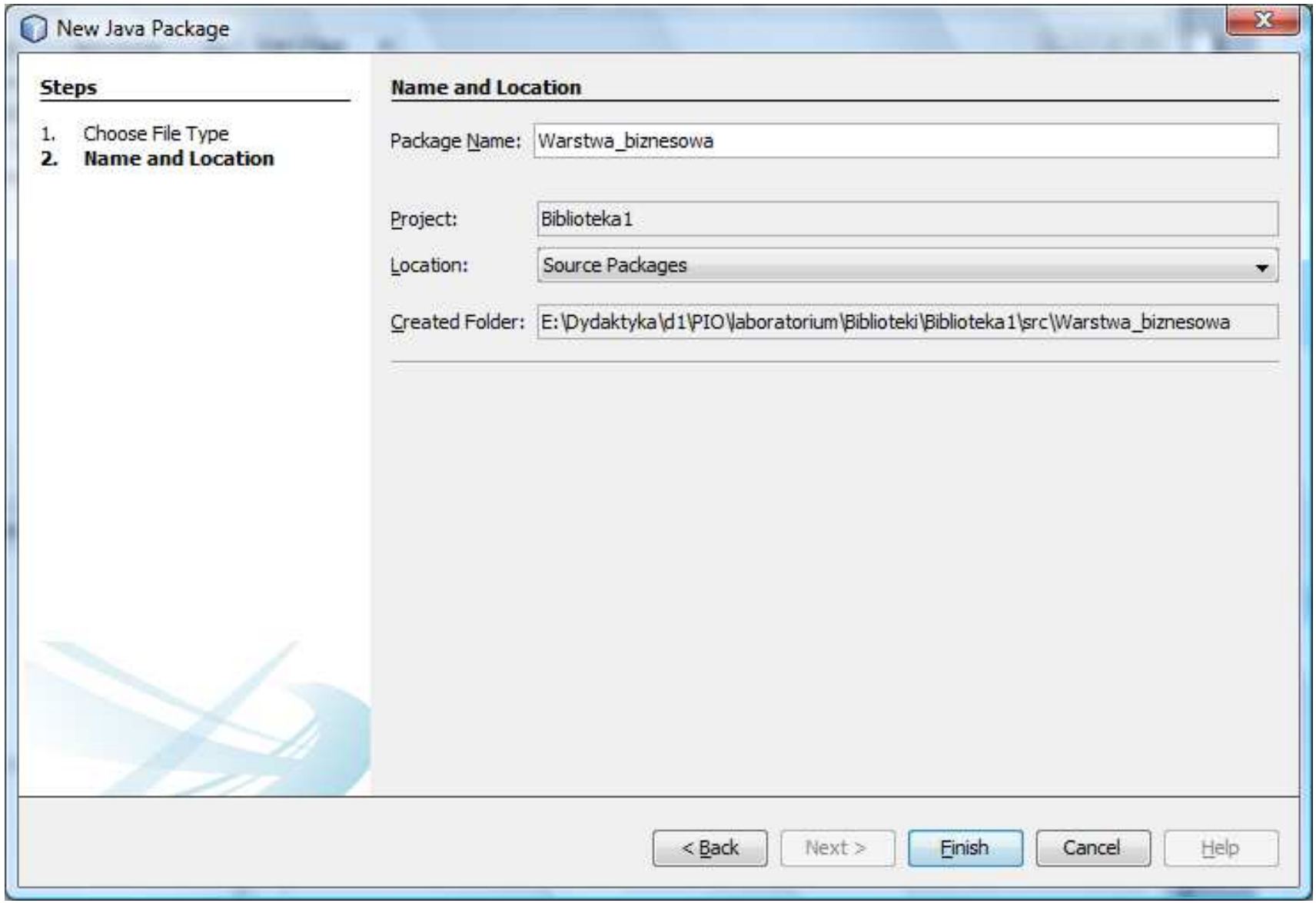
18.3. Wykonanie projektu typu *Java/ Java Application* – nadanie nwazy projektowi w polu *Project Name* oraz lokalizacji za pomocą klawiasza *Browse…* w polu *Project Location* bez ustawienia opcji *Create Main Class*



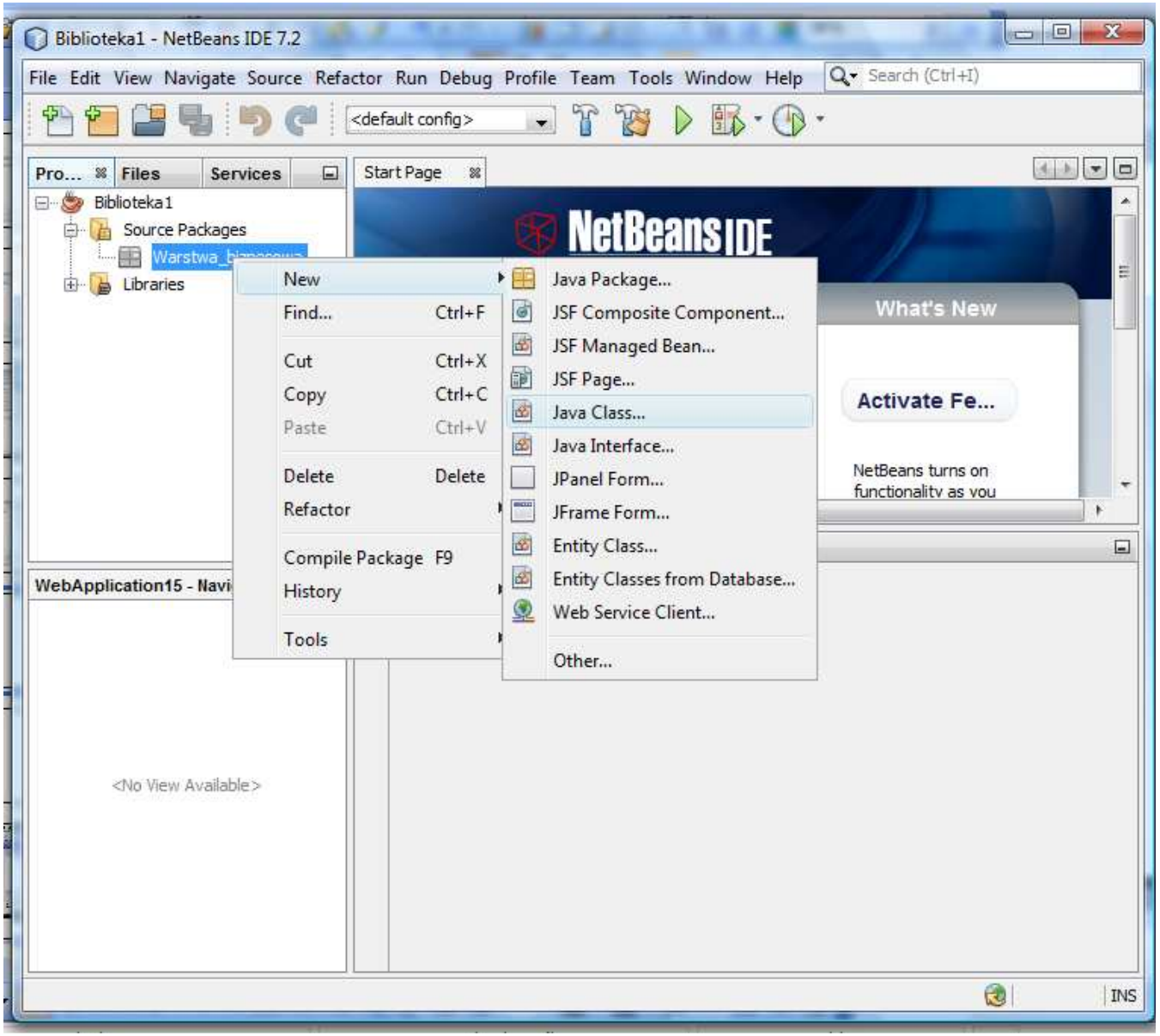
19.1. Wstawienie nowego pakietu do projektu – prawym klawiszem należy kliknąć na pozycję *Source Package* w okienku *Projects i wybrać z listy pozycję Java Package* (lub *Other*, jeśli nie ma takiej pozycji na liście)



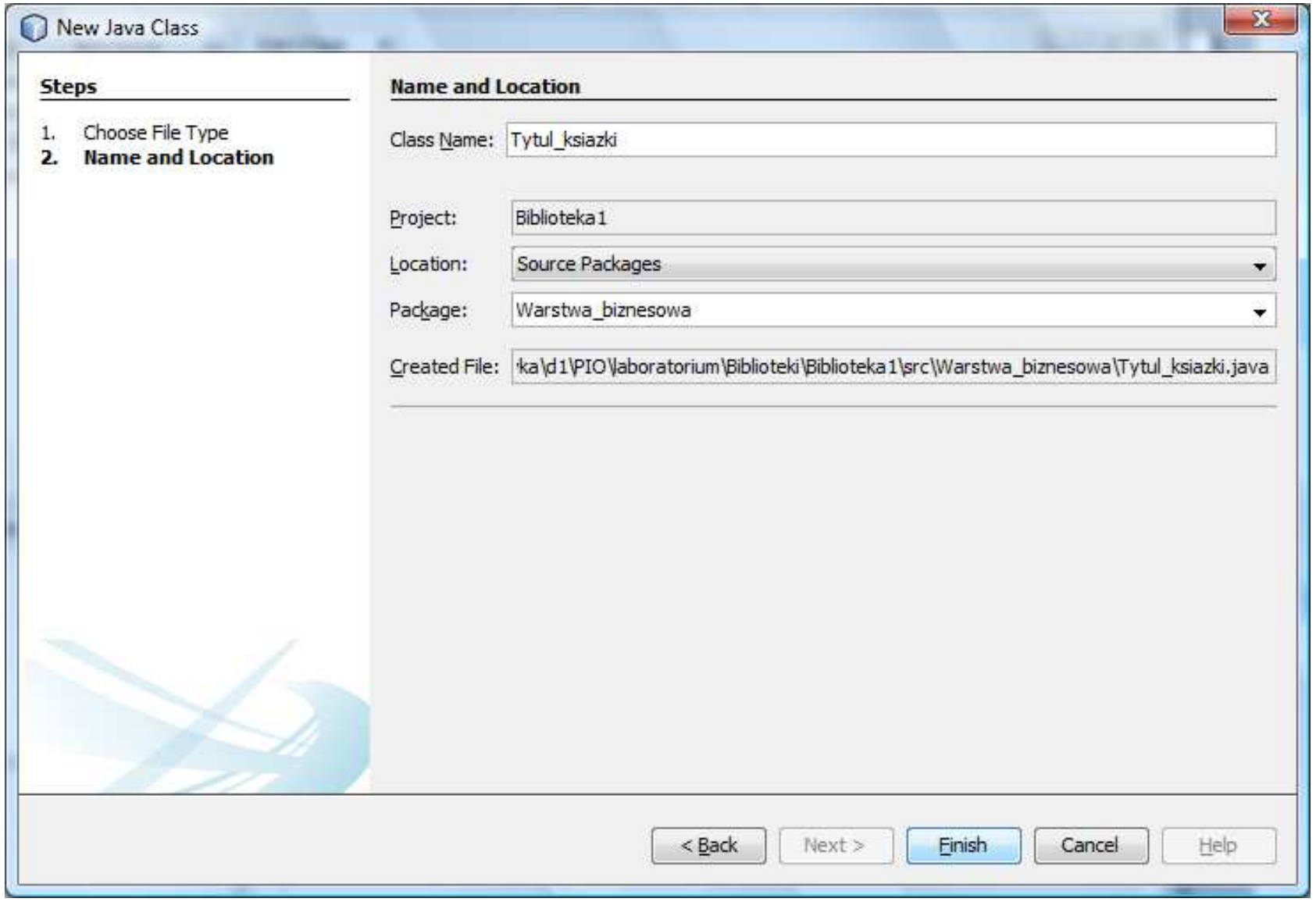
19.2. Wstawienie nowego pakietu do projektu – nadanie nazwy pakietowi **Warstwa_biznesowa** w polu *Package Name*



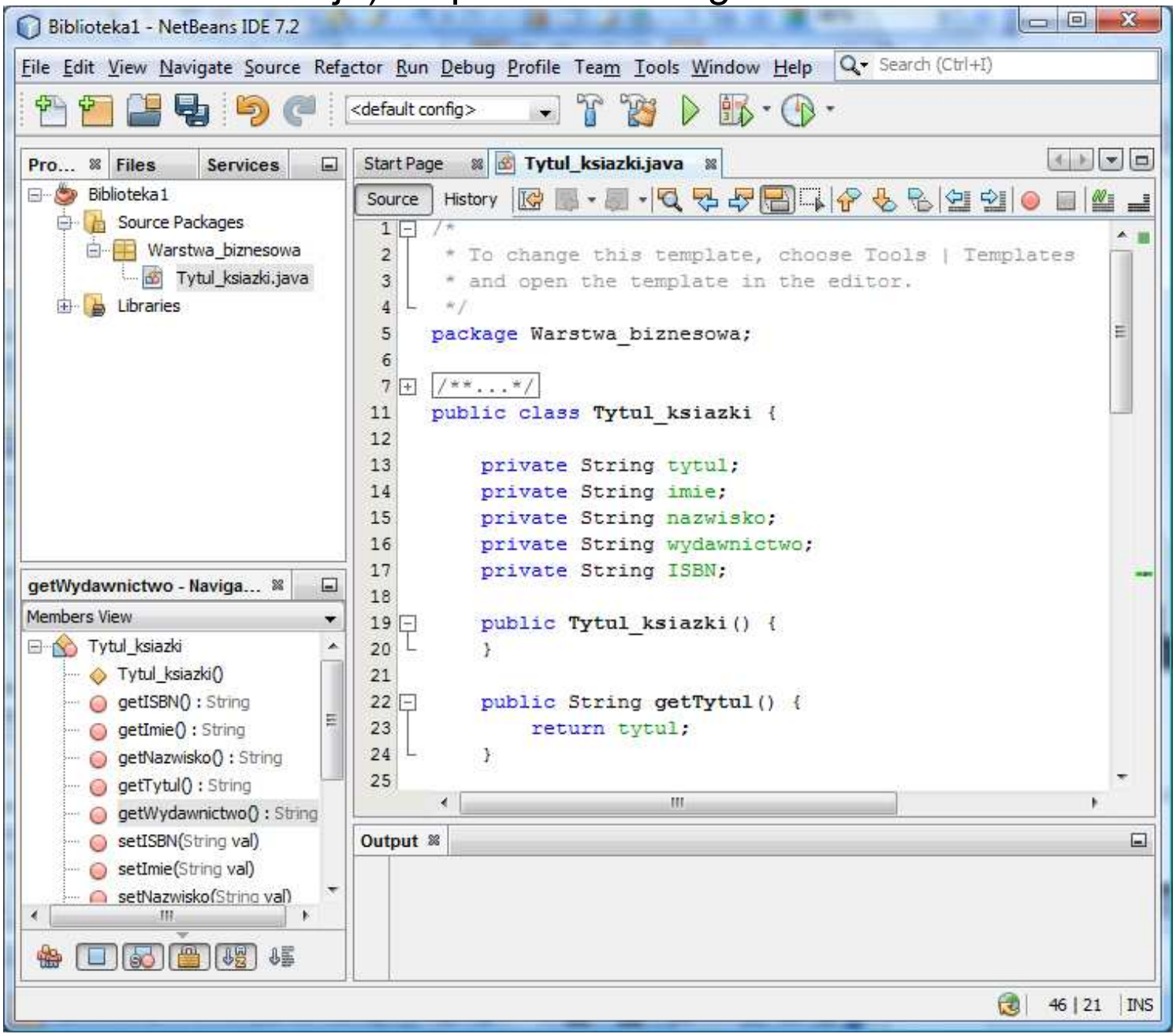
20.1. Wstawienie do pakietu **Warstwa_biznesowa** nowej klasy – należy kliknąć prawym klawiszem myszy na nazwę pakietu i wybrać z listy pozycję *Java Class* (lub *Other*, jeśli nie ma takiej pozycji na liście)



20.2. Nadanie nazwy nowej klasie **Tytul_ksiazki** w polu *Class Name*



20.3. Zdefiniowanie kodu klasy **Tytul_ksiazki** (kod klasy zawiera następny slajd) na podstawie diagramu klas

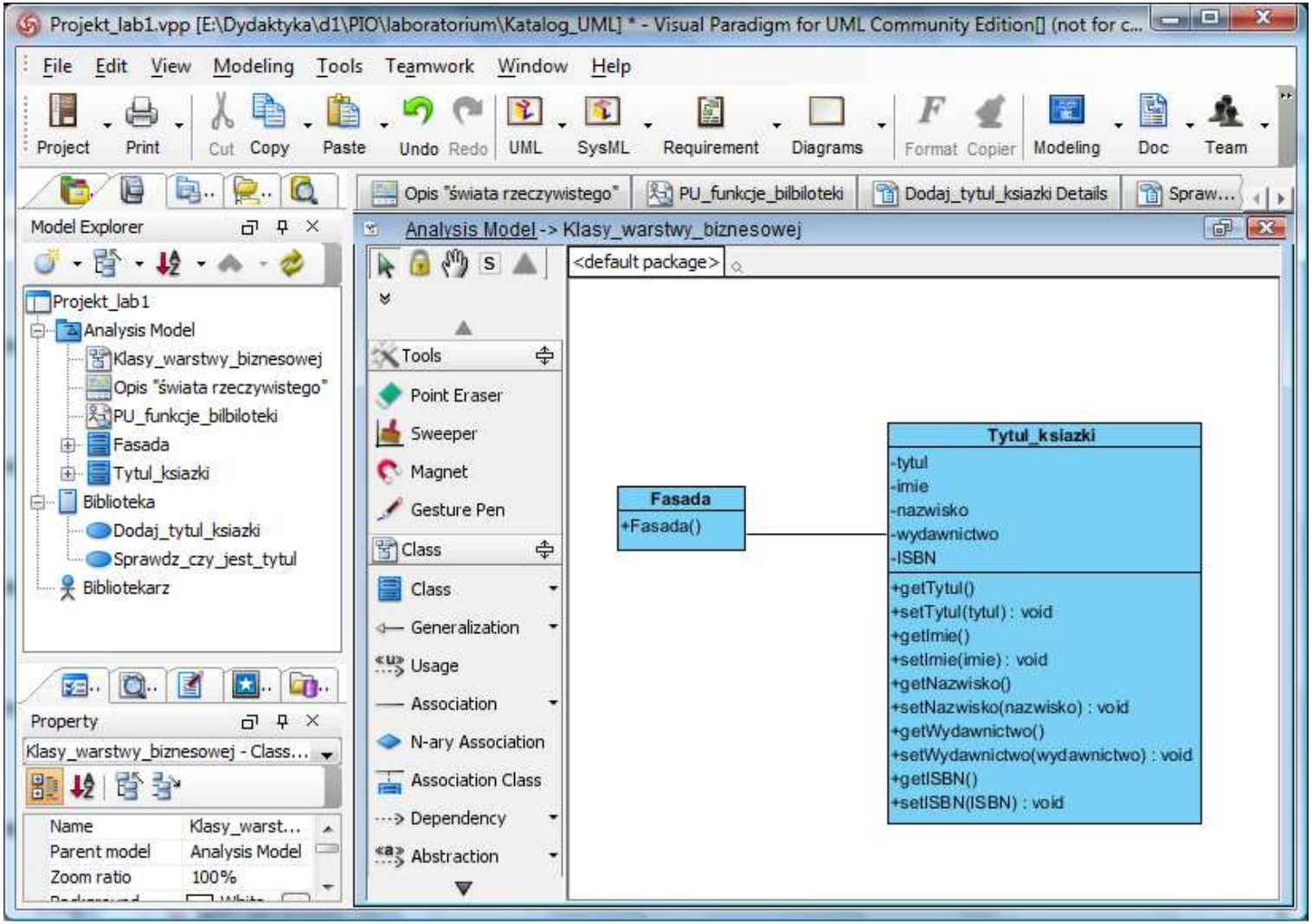


20.4, Kod klasy Tytul_ksiazki

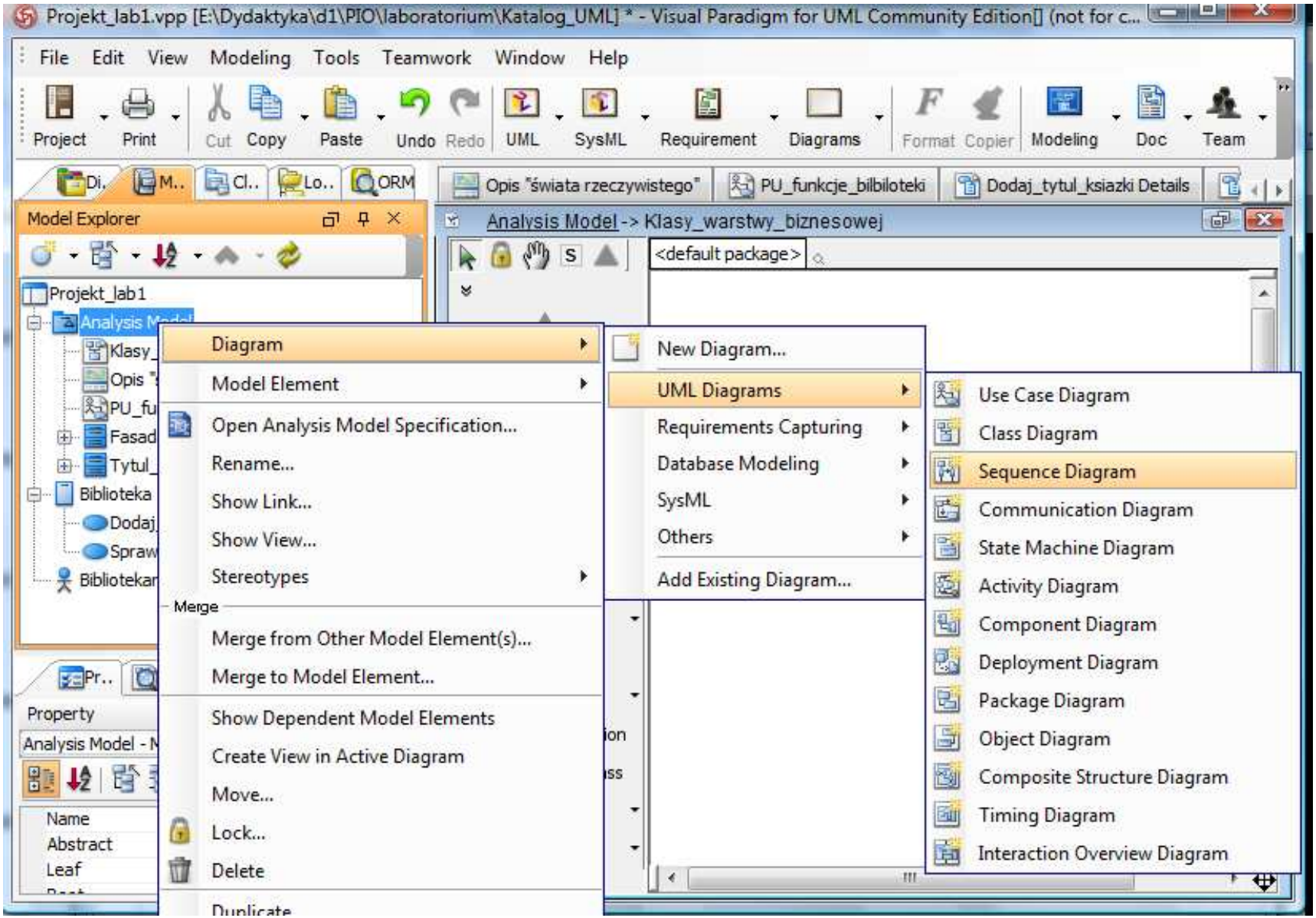
```
public class Tytul_ksiazki {
    private String tytul;
    private String imie;
    private String nazwisko;
    private String wydawnictwo;
    private String ISBN;

    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; }

}
```

Relacja jeden do jeden

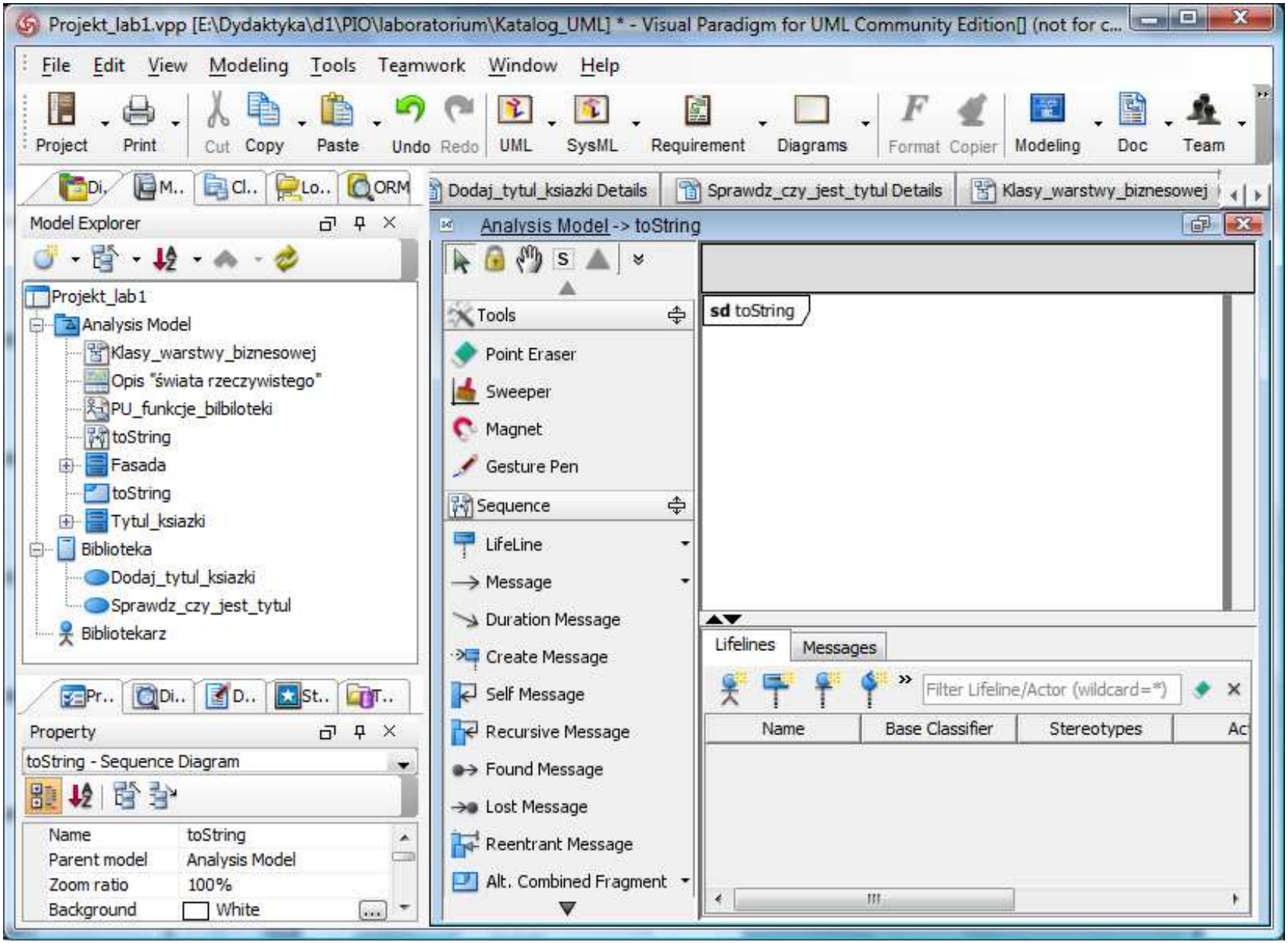
21. Wstawienie nowej klasy **Fasada** (podobnie jak klasę Tytul_ksiazki) powiązanej za pomocą relacji *Association* 1..0 z klasą typu **Tytul_ksiazki** – relację należy wybrać z palety z lewej strony lewym klawisxzem myszy oraz położyć ją na klasie **Fasada** i przeciągnąć na klasę **Tytul_ksiazki.** Klasa ta reprzentuje wzorzec projektowy Fasada - będzie zastosowana do obsługi wywołań przypadków użycia przez warstwę interfejsu graficznego użytkownika.



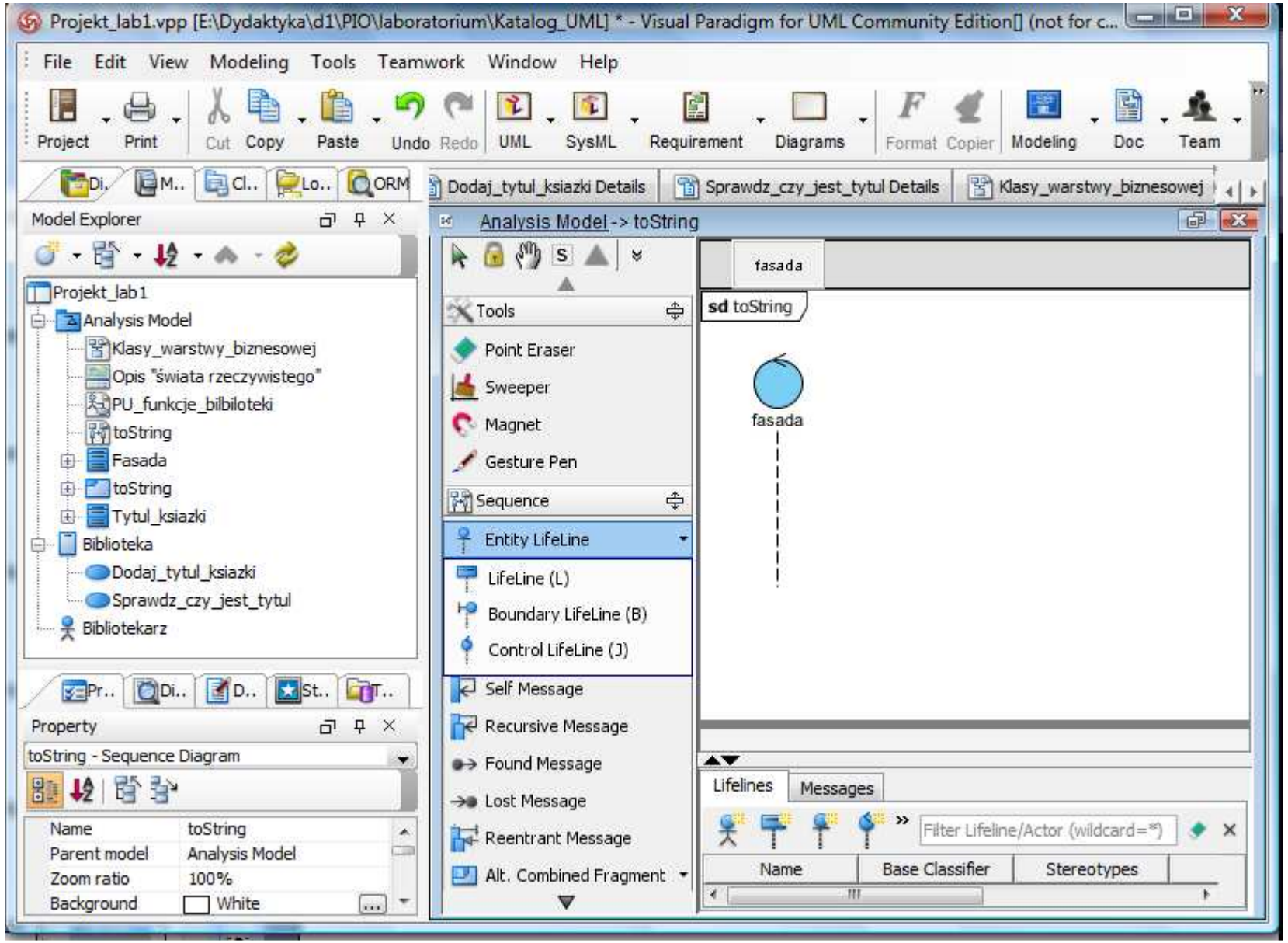
22.1. Wstawianie diagramu sekwencji – należy kliknąć prawym klawiszem myszy na nazwę modelu w okienku Model Explorer i wybrać z listy opcję *Diagram/UML Diagram/Sequence Diagram*



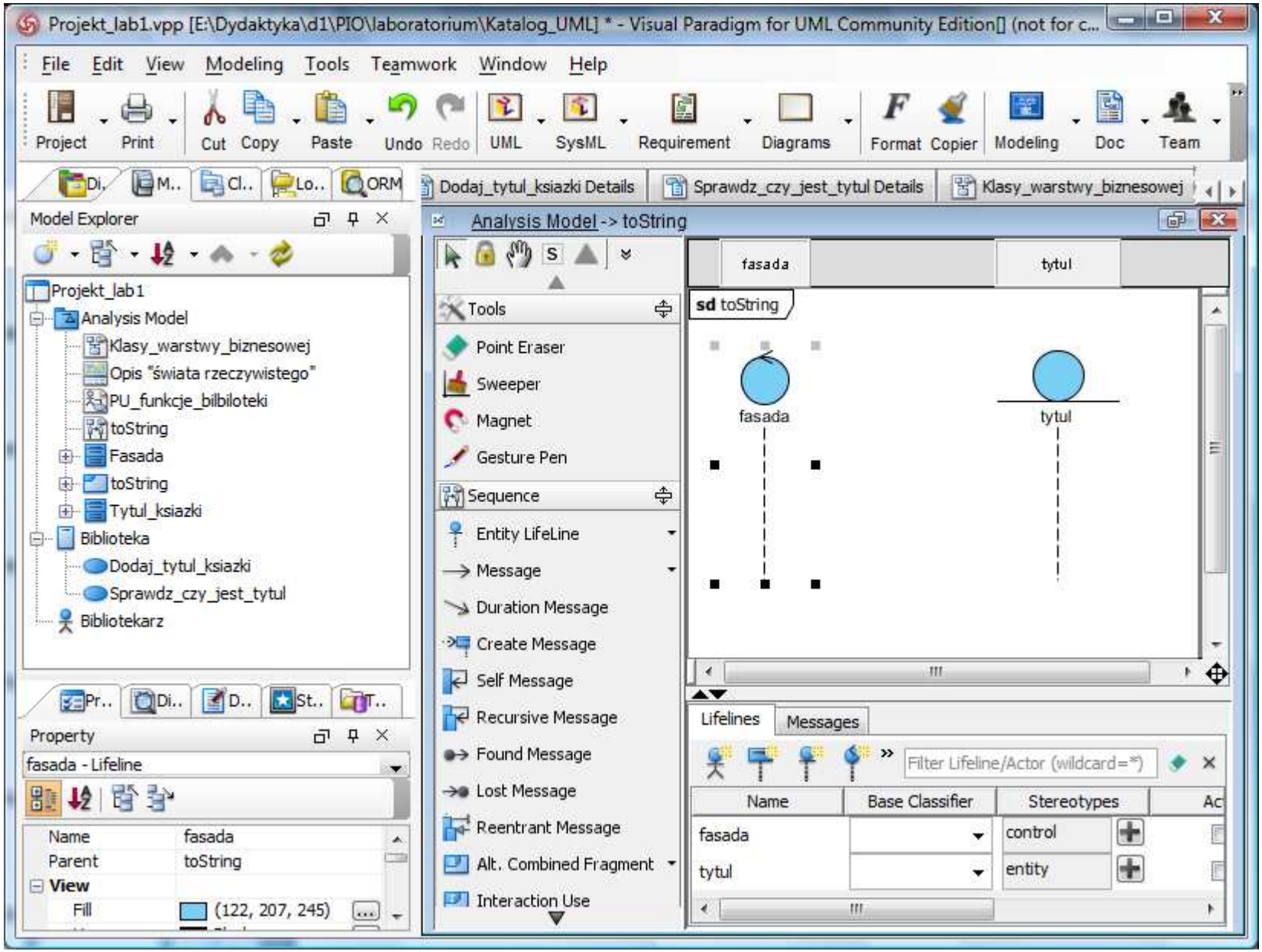
22.2. Należy nadać nazwę t**oString** diagramowi sekwencji – diagram będzie zawierał definicję metody **toString** w klasie **Tytul_książki**



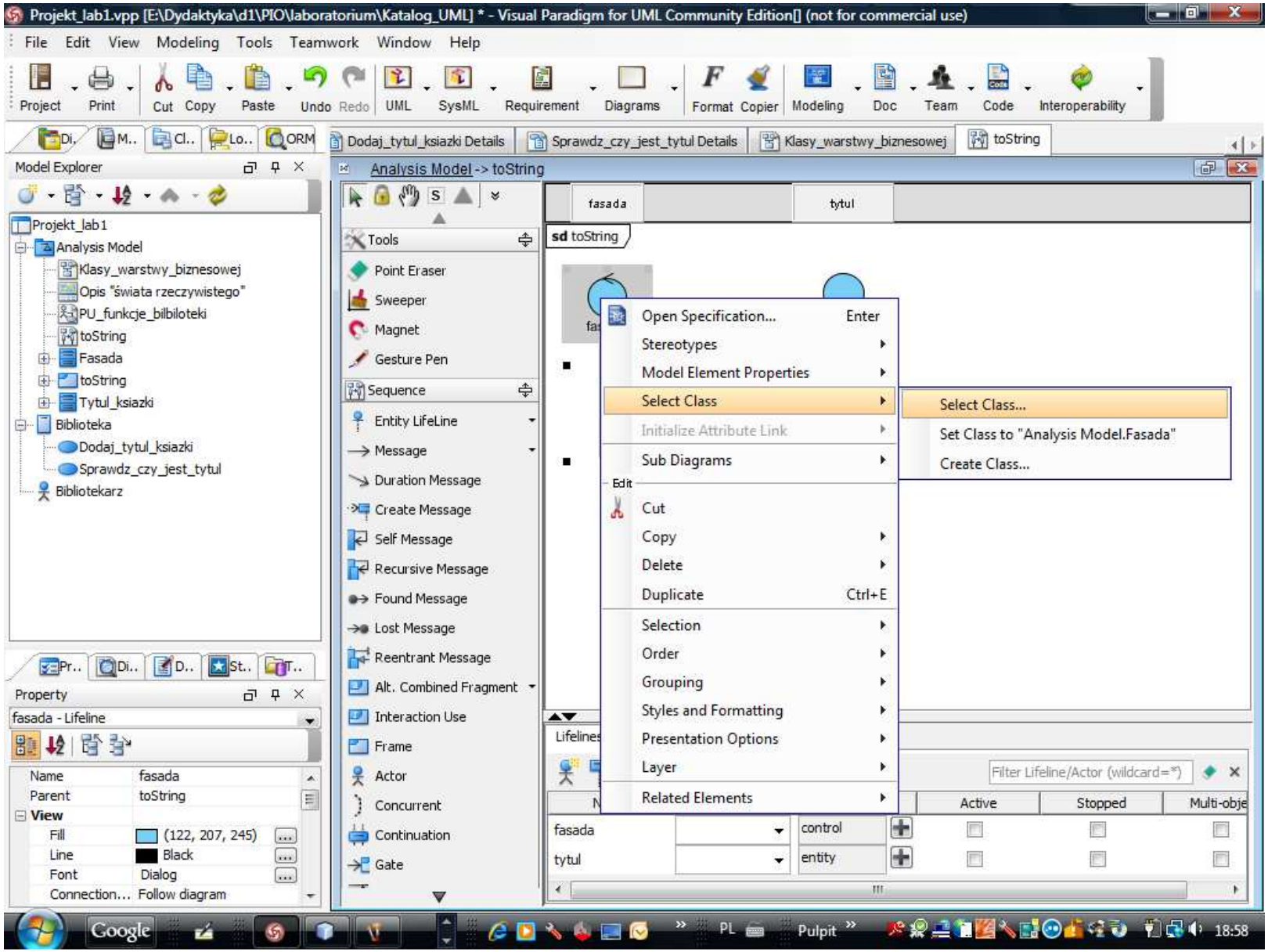
22.3. Należy z palety z lewej strony wybrać z listy Lifeline linię życia typu Control Lifeline



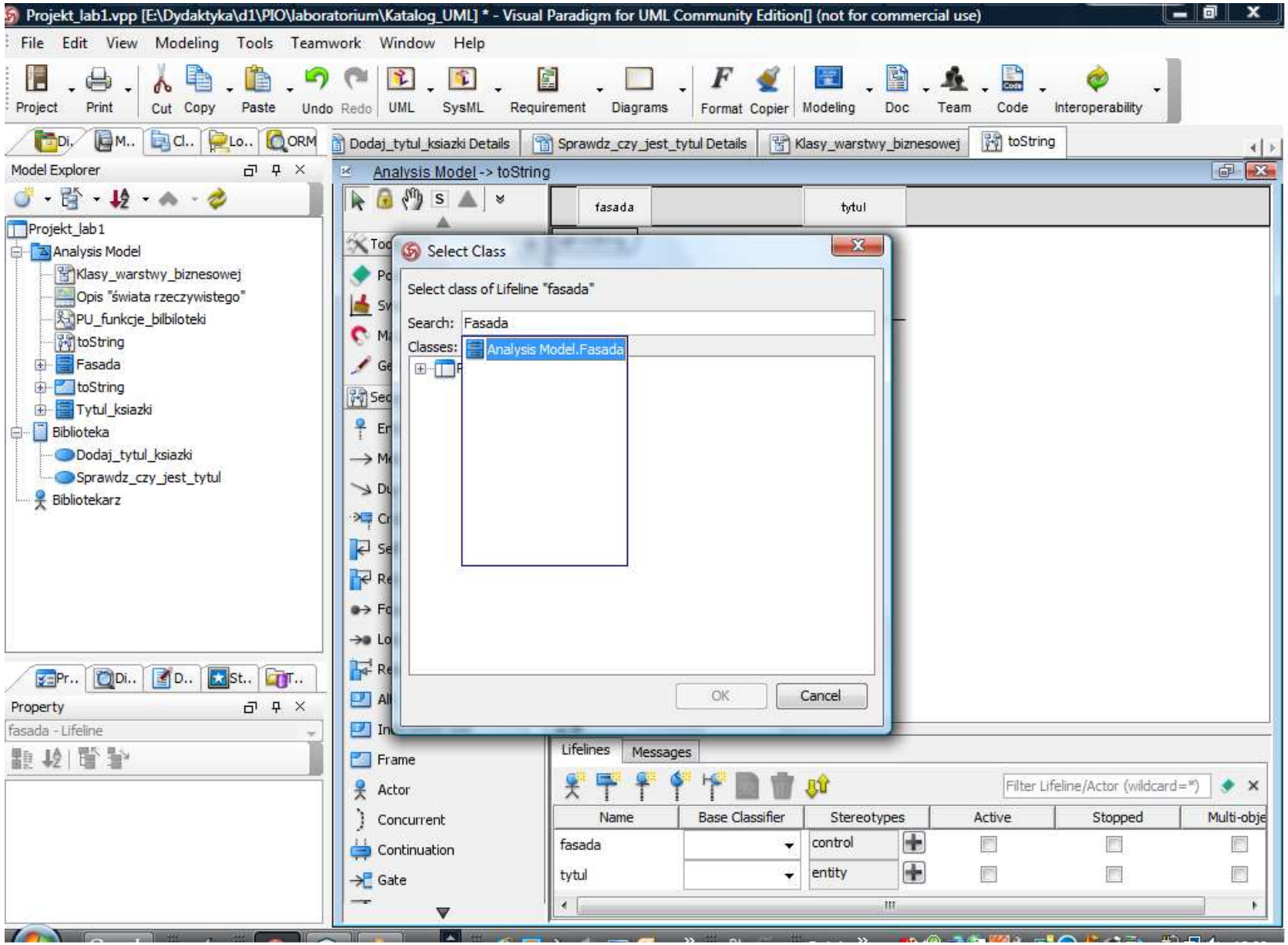
22.4. Należy z palety z lewej strony wybrać z listy Lifeline linię życia typu *Entity Lifeline* i nadać nazwę **tytul**



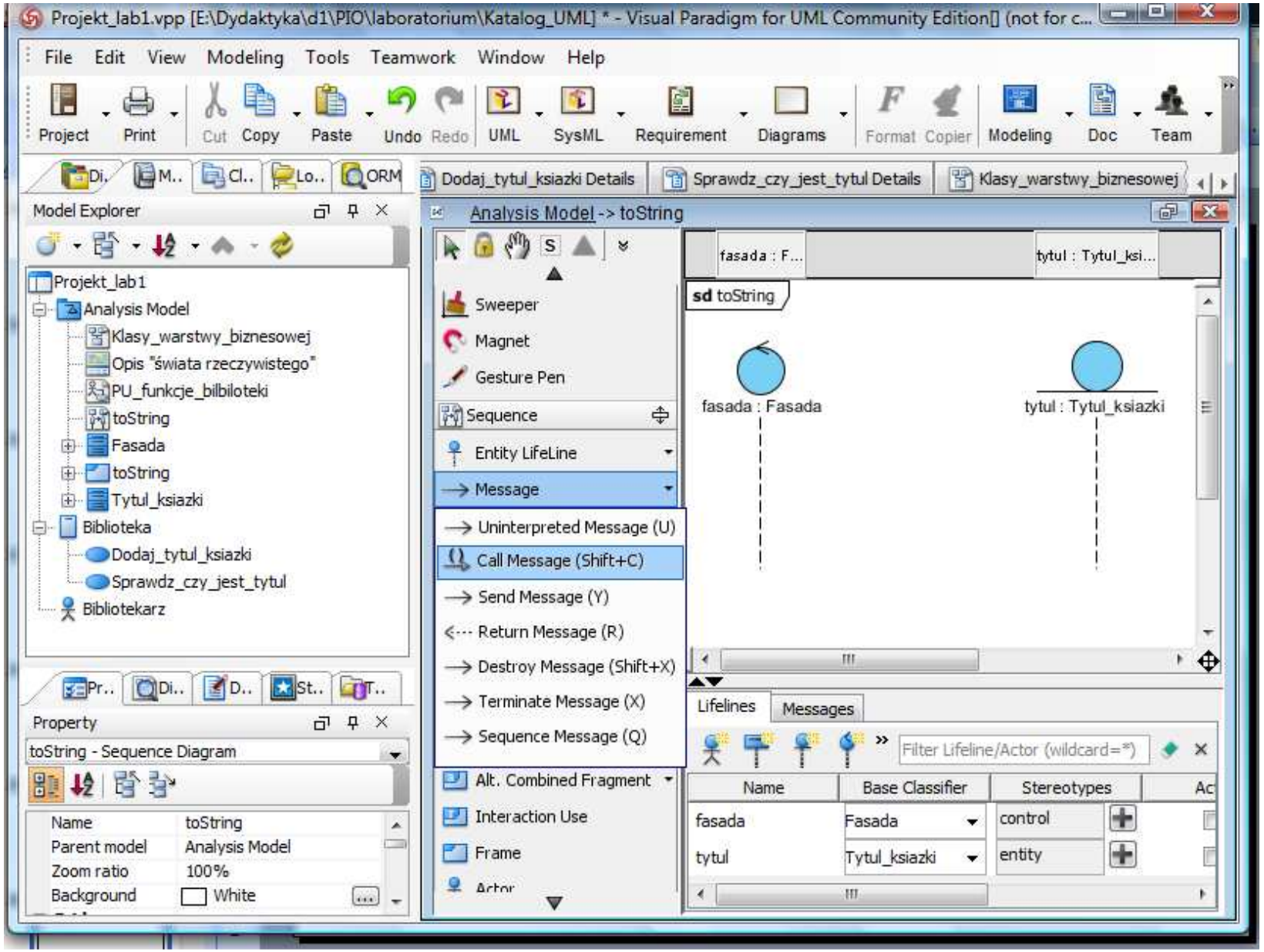
22.5. Należy linie życia obiektów powiązać z klasami z diagramu klas – po wybraniu linii życia **fasada** należy kliknąć prawym klawiszem myszy i wybrać z listy opcję *Select Class/Select Class*



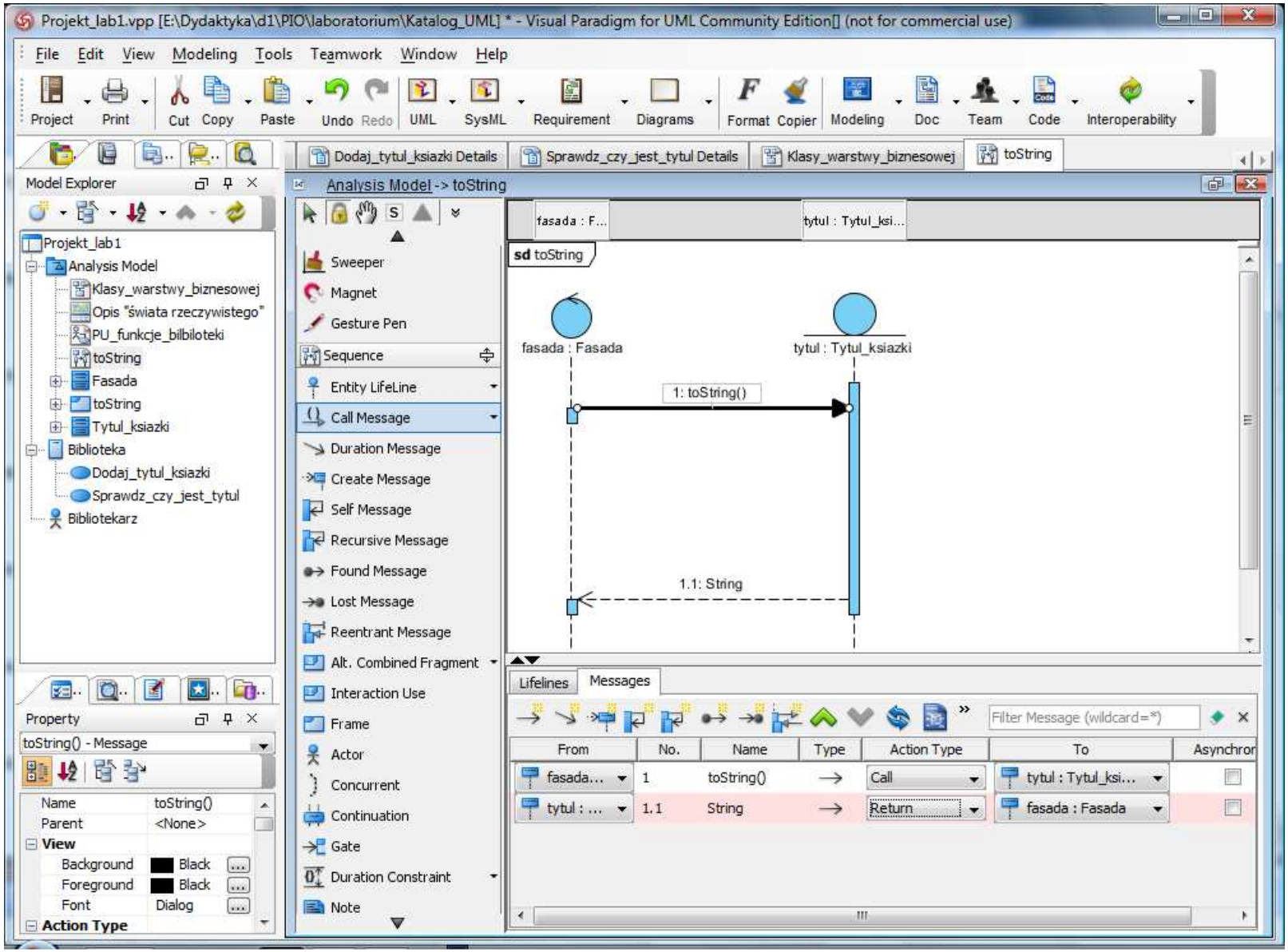
22.6. W formularzu *Select Class* należy w polu *Search* wpisać fragment nazwy klasy **Fasada**. W ukazanym okienku wybrać właściwą klasę i nacisnąć klawisz *OK.* Podobnie należy powiązać linię życia **tytul.**



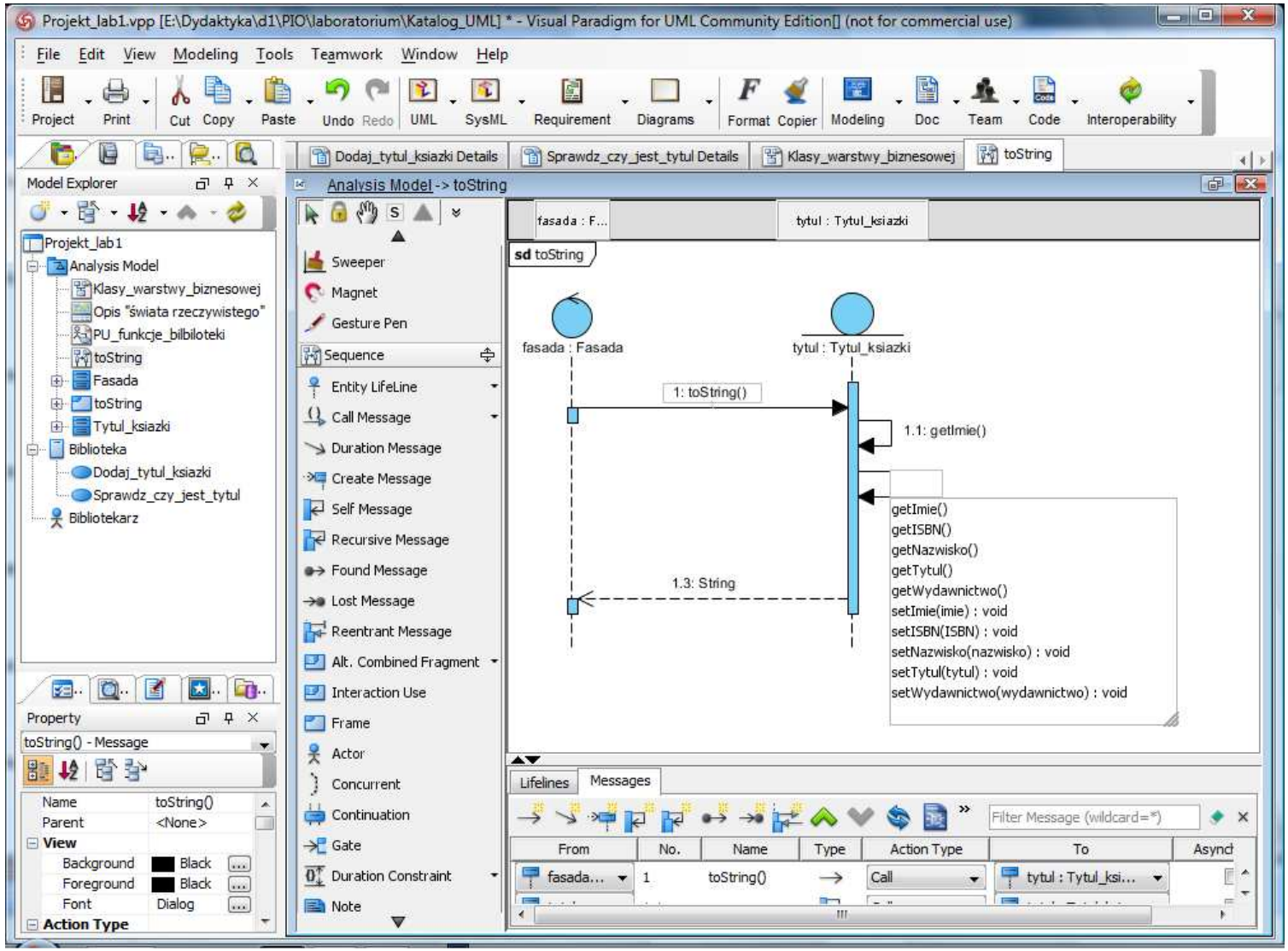
23.1. Należy wybrać z listy *Message* typ metody *Call Message* i przeciągnąć ją kładąc na linii życia **fasada** i przeciągając położyć na linii życia **tytul**. Podobnie należy zrobić z wiadomością typu *Return Message*.



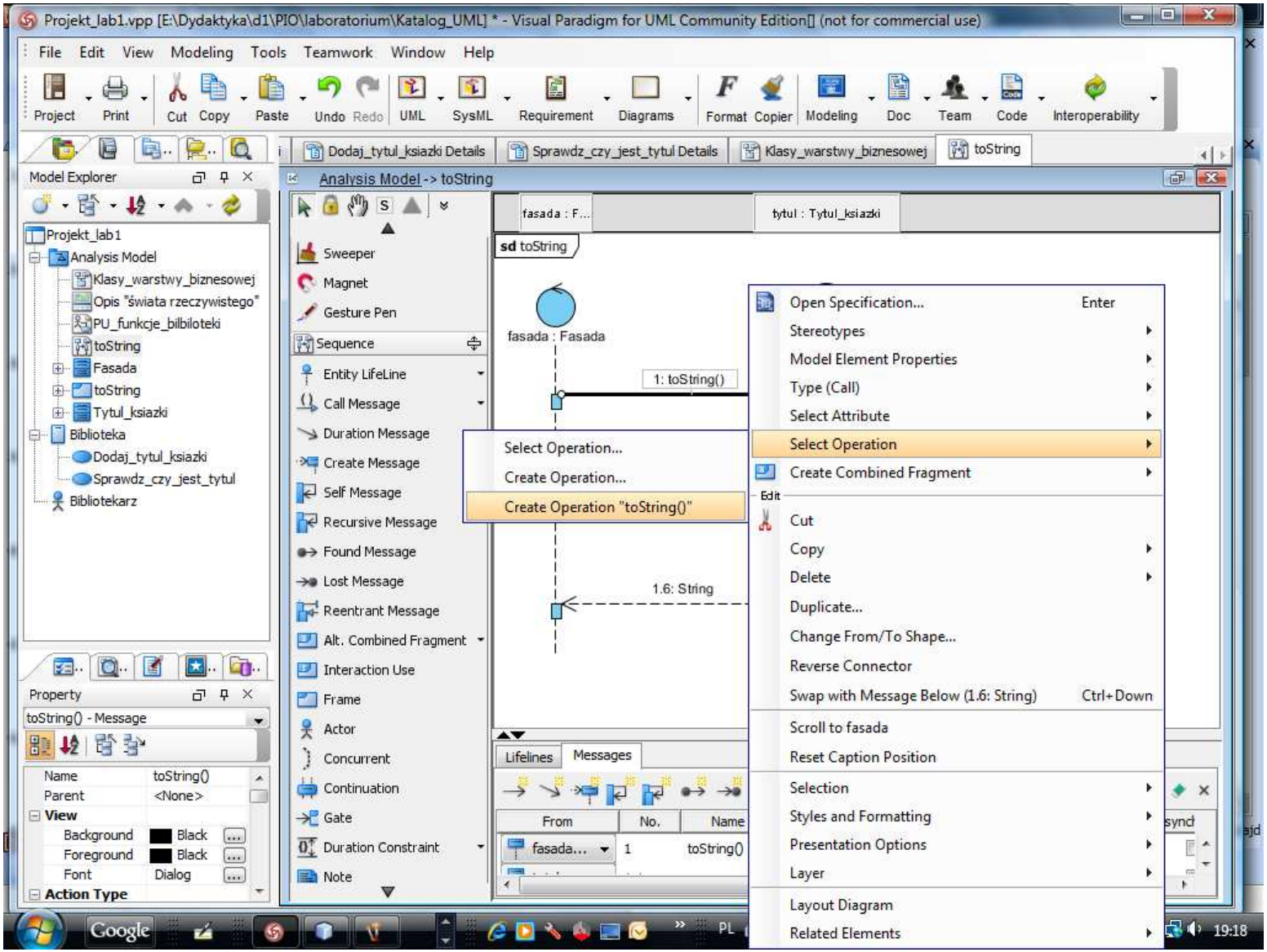
23.2. Należy wybrać z listy *Message* typ metody *Call Message* i przeciągnąć ją kładąc na linii życia **fasada** i przeciągając położyć na linii życia **tytul**. Podobnie należy zrobić z wiadomością typu *Return Message,* która powinna wychodzic z linii życia **tytul** i wchodzić do linii życia **fasada**.



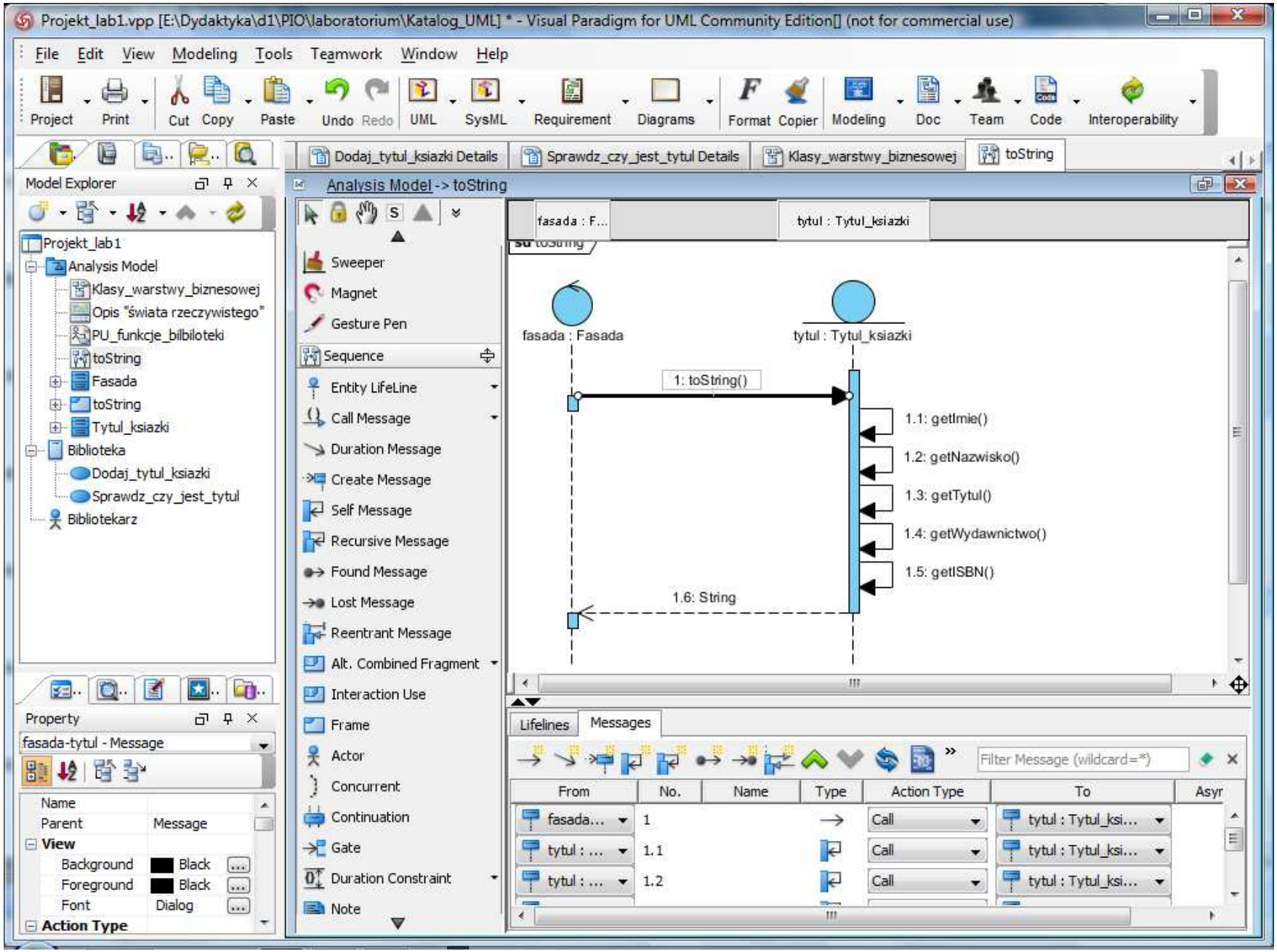
23.3. Należy zdefiniować ciało metody **toString** w klasie **Tytul_ksiazki** za pomocą wiadomości *Self Message*, przeciąganych z palety z lewej strony. Podczas wstawiania wiadomości pokazuje się lista metod klasy **Tytul_ksiazki** zdefiniowanych podczas tworzenia diagramu klas. Należy dokonać wyboru właściwej metody typu get z listy metod klasy **Tytul_ksiazki**.



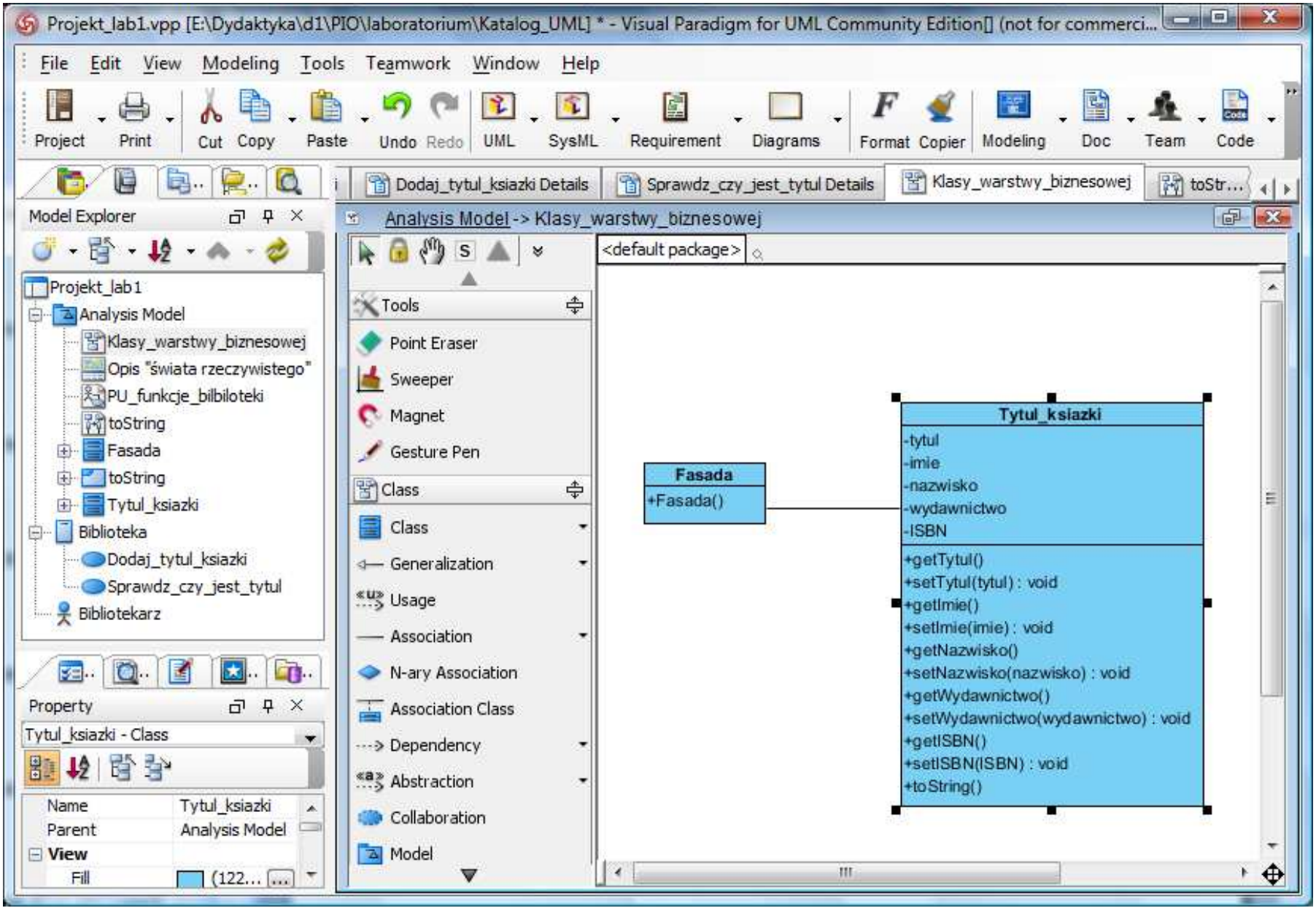
23.4. Włączenie zdefiniowanej metody do klasy **Tytul_ksiazki** – po wybraniu wiadomości o nazwie **toString** klikając prawym klawiszem myszy należy wybrać z list pozycje *Select Operation/Select Operation „toString()”*



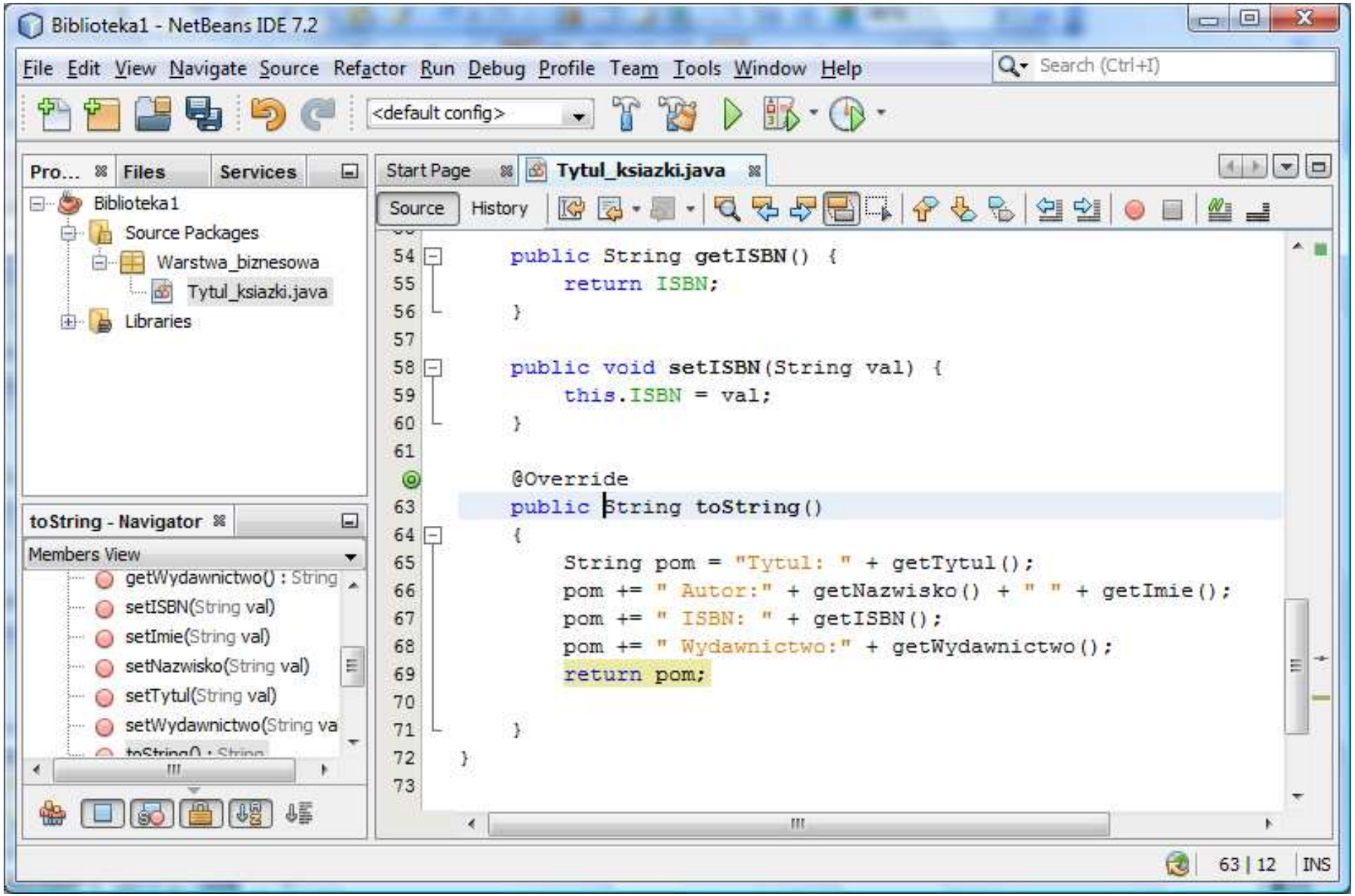
23.5. Rezultat wykonania scenariusza metody *toString* w klasie **Tytul_ksiazki.**



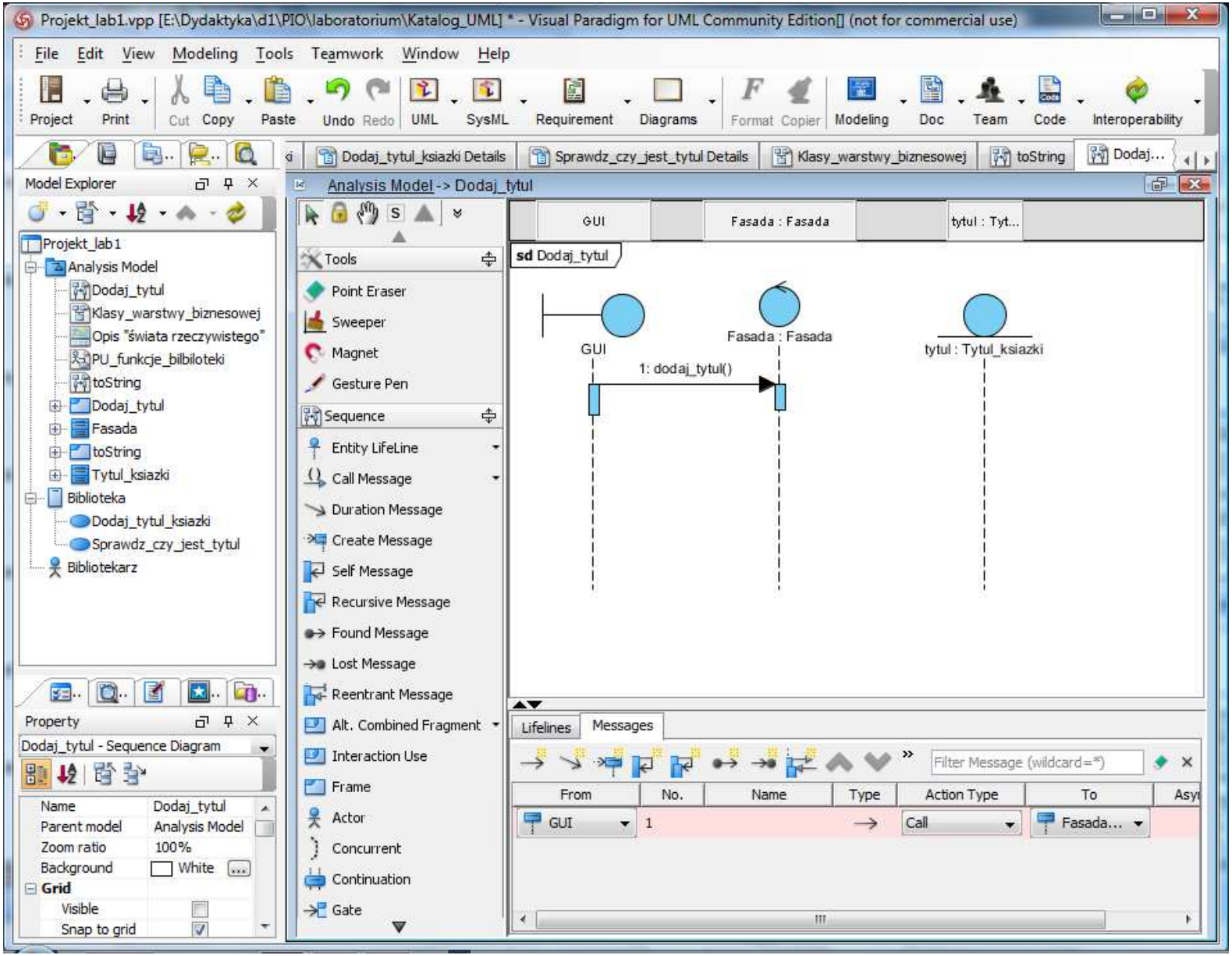
23.6. Metoda **toString** zdefiniowana na diagramie sekwencji pojawiła się w klasie **Tytul_ksiazki**



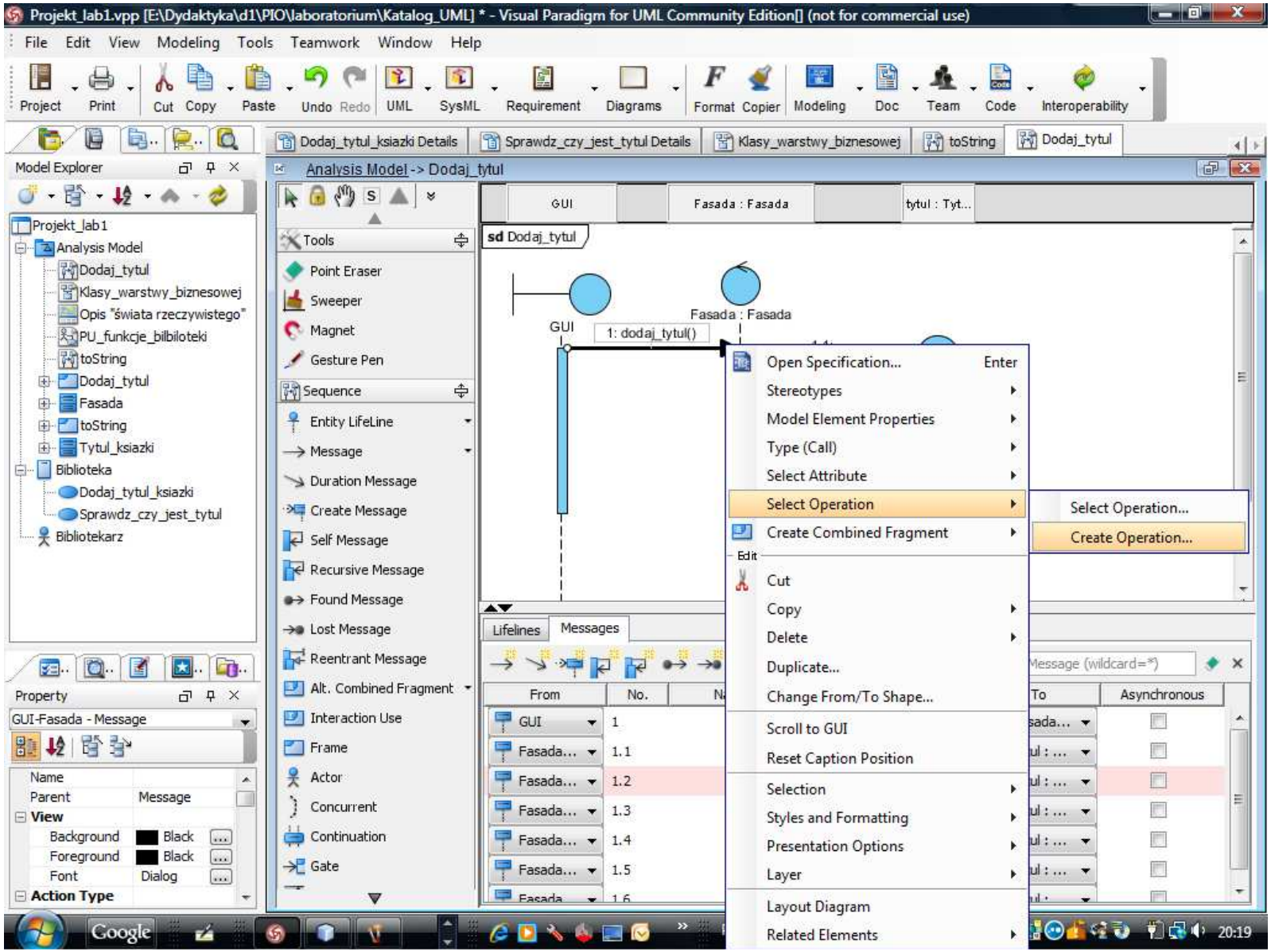
23.7. Definicja kodu metody **toString** w klasie **Tytul_ksiazki** zdefiniowana na podstawie diagramu sekwencji (p. 23.5)



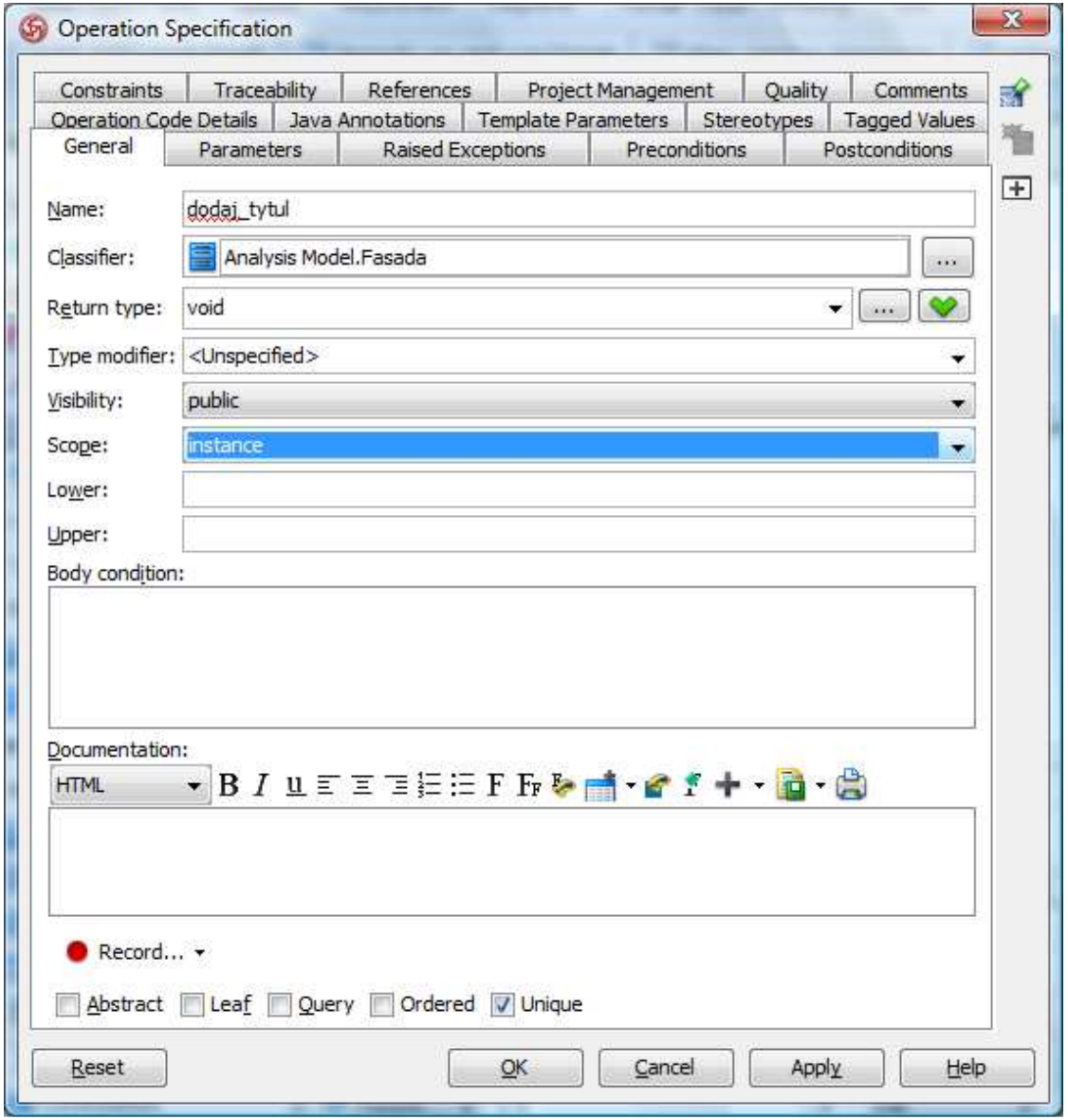
24.1. Dodanie nowego diagramu sekwencji o nazwie **Dodaj_tytul** reprezentującego definicję przypadku użycia **Dodaj_tytul**.



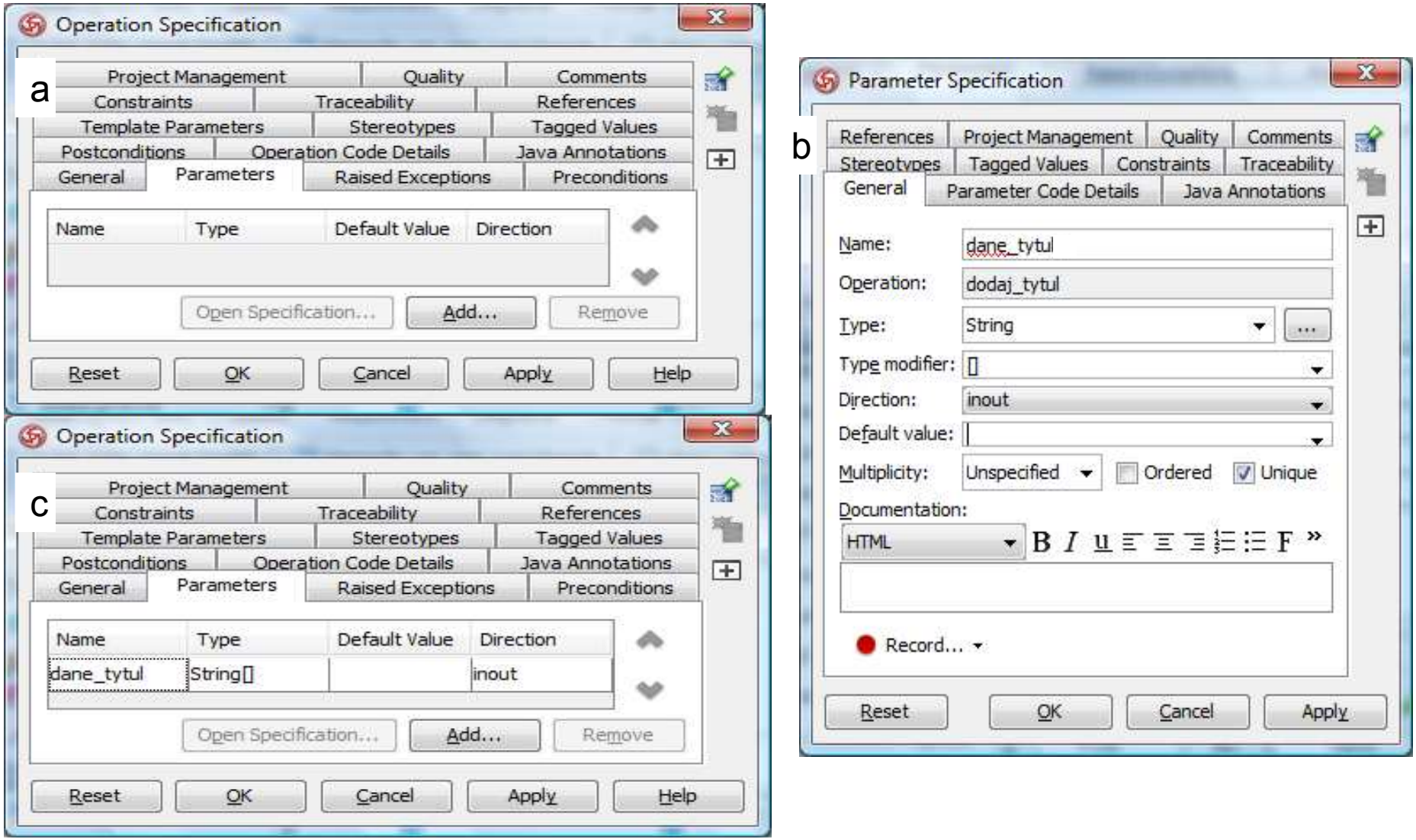
24.2. Należy wiadomość **Dodaj_tytul** z klasy przypisać do klasy **Fasada** – należy kliknąć prawym klawiszem myszy na strzałkę reprezentującą metodę i wybrać z list opcje *SelectOperation/Create Operation*



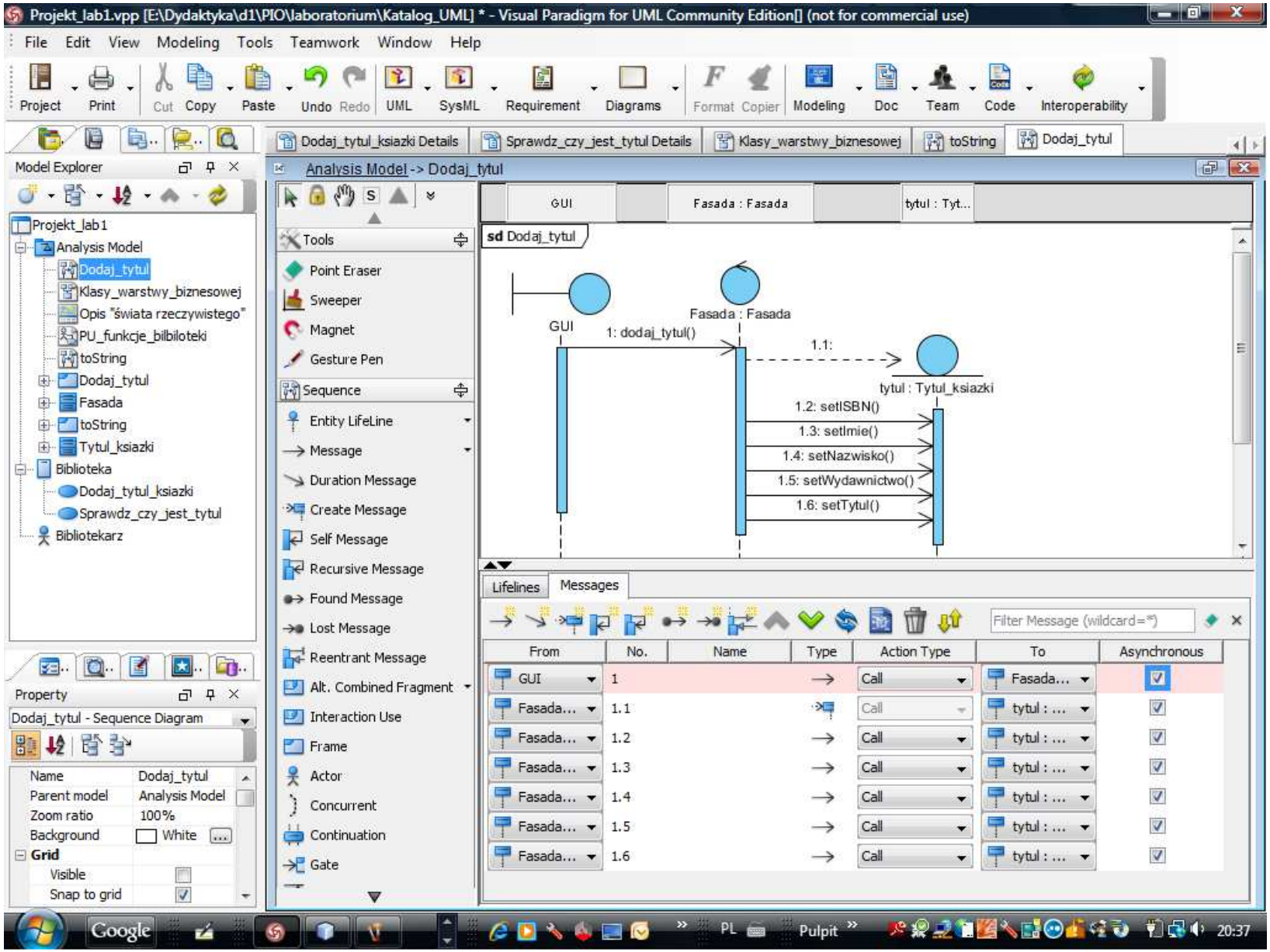
24.3. Na formularzu *Operation Specification* należy wypełnić nastepujące pola: *Name* (nazwa metody), *Classifier* (Nazwę klasy), *Return type* (wynik zwracany przez metodę), *Visibility* (określanie zasięgu metody). Następnie należy zdefiniować listę parametrów metody wybierając kolejny formularz w zakładce *Parameters*



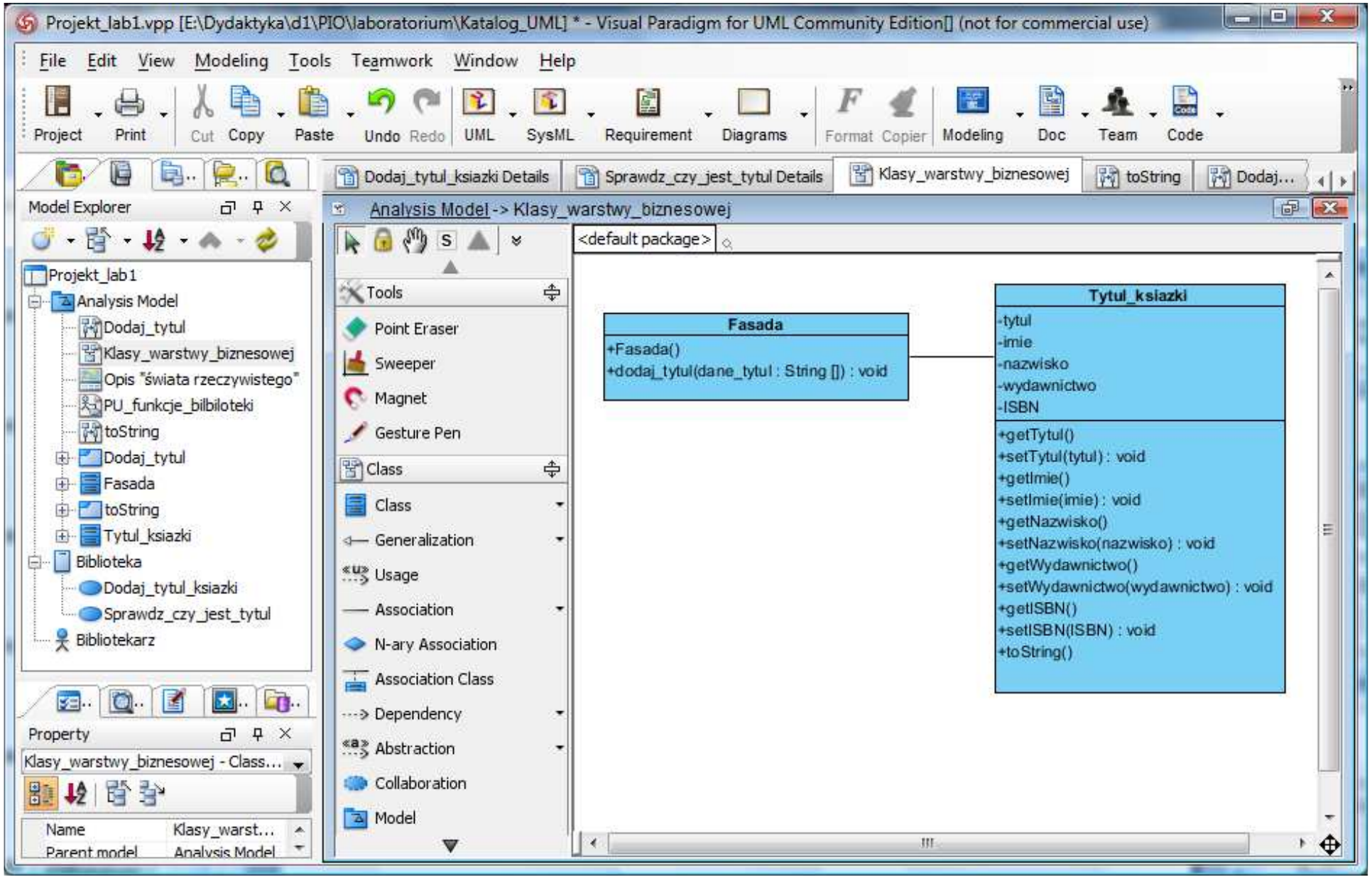
24.4. Zdefiniowanie parametru metody **Dodaj_tytul** w oknie *Parameters*: w okienku *a* należy kliknąć klawisz *Add*. W formularzu *b* w polu *Name* należy wpisać nazwę parametru **dane_tytul**, w polu *Operation* nazwę metody, w polu *Type* typ parametru, w polu *Type modifier* uzupełnienie definicji typu parametru. Po kliknięciu na klawisz *OK* następuje powrót do okna *Parameters*, gdzie uwidoczny jest nowy parametr



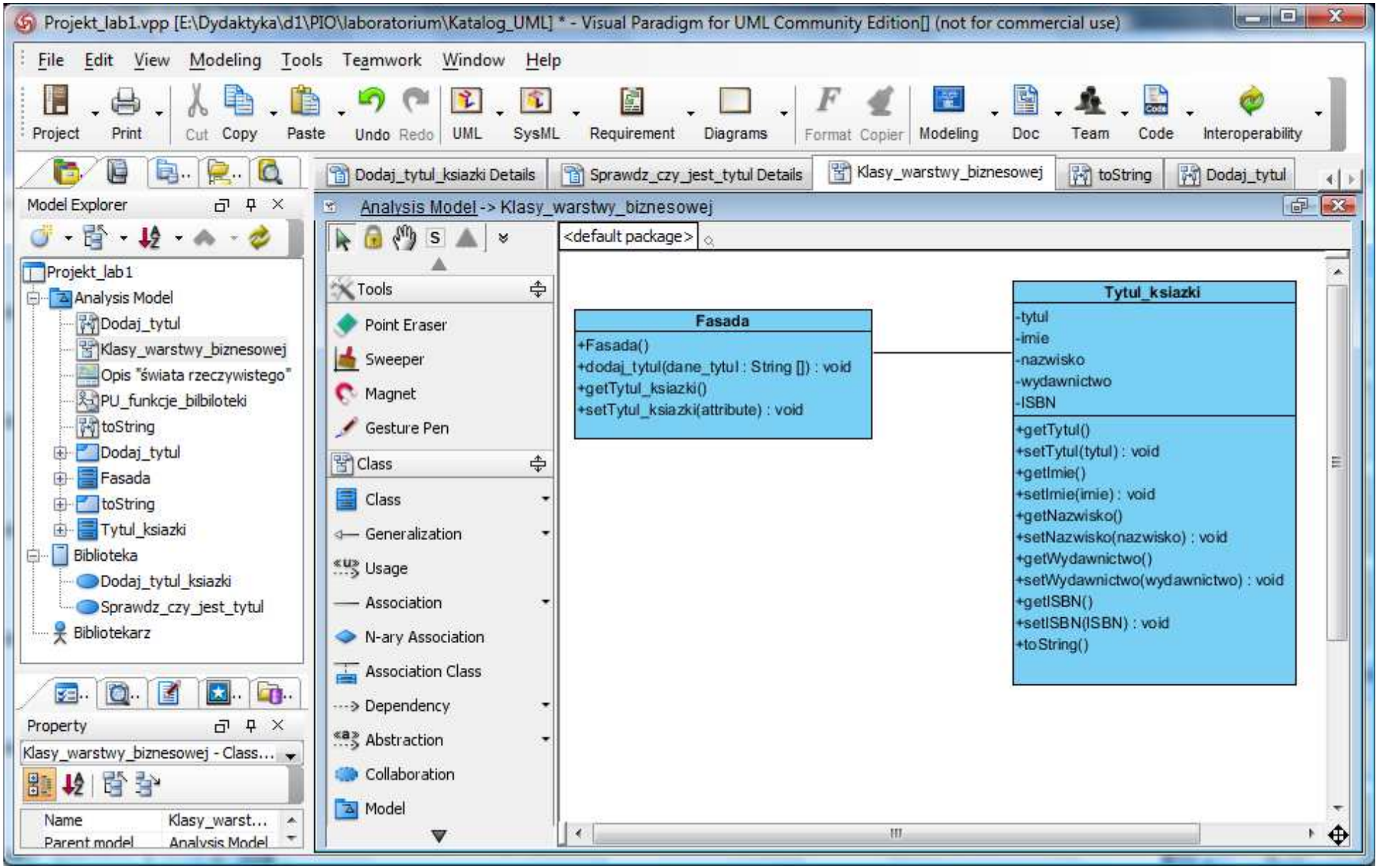
24.5.Pełna definicja metody **dodaj_tytul** w klasie **Fasada** - linia życia **tytul** typu **Tytul_ksiazki** została zdefiniowana jako *Create Message* przez przeciągnięcie na linię życia typu *Entity Lifeline* ikony *Create Message* z palety z lewej strony.



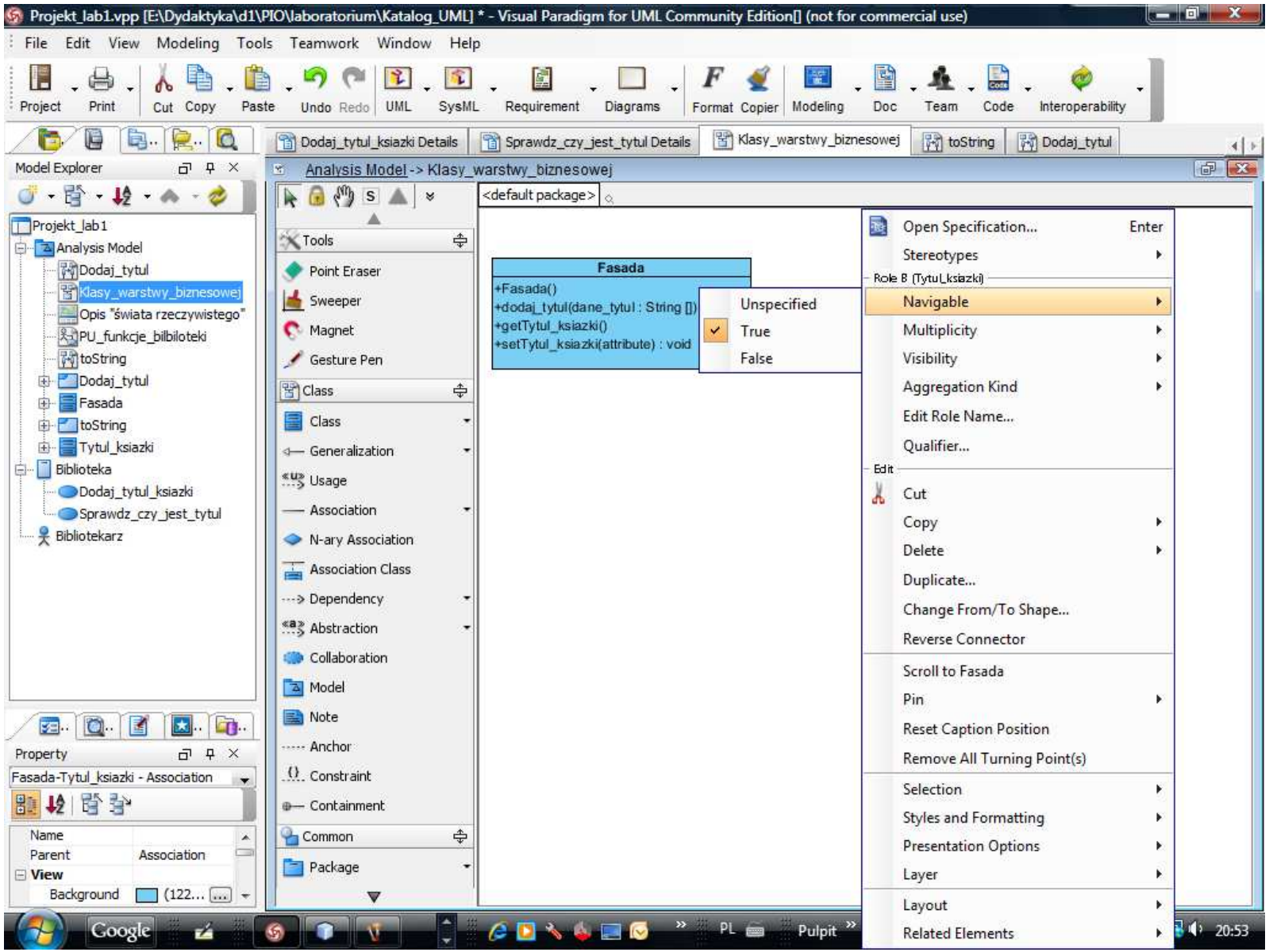
24.6. Uwidocznienie nowej metody **dodaj_tytul** z listą parametrów w klasie **Fasada**, zdefiniowanej na diagramie sekwencji z p.



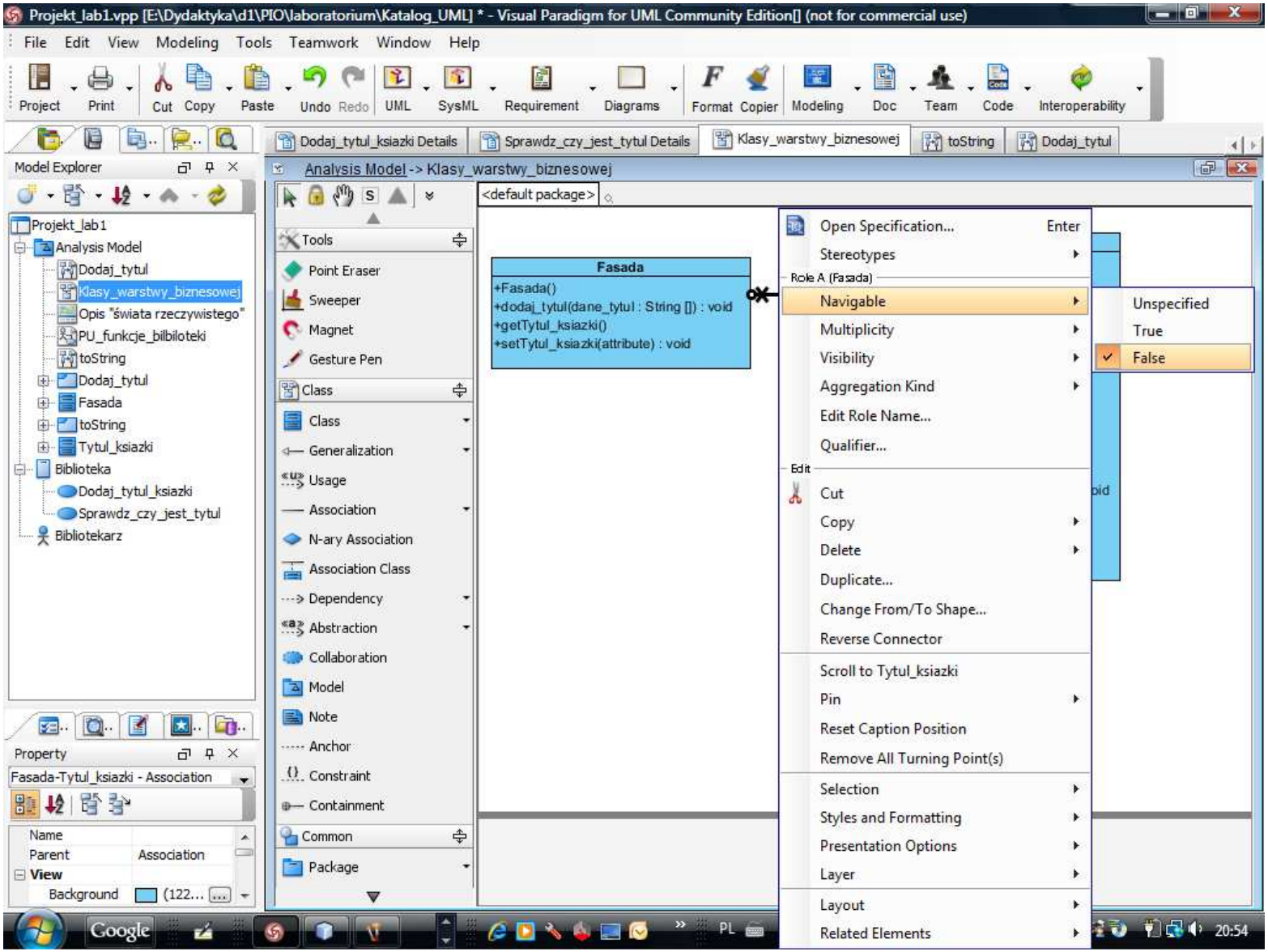
25.1. Dodanie dwóch operacji typu **get** i **set**, reprezentujących implemenetację relacji *Association* w klasie **Fasada** (obiekt typu **Fasada** będzie miał dostęp do jednego obiektu typu **Tytul_ksiazki**)



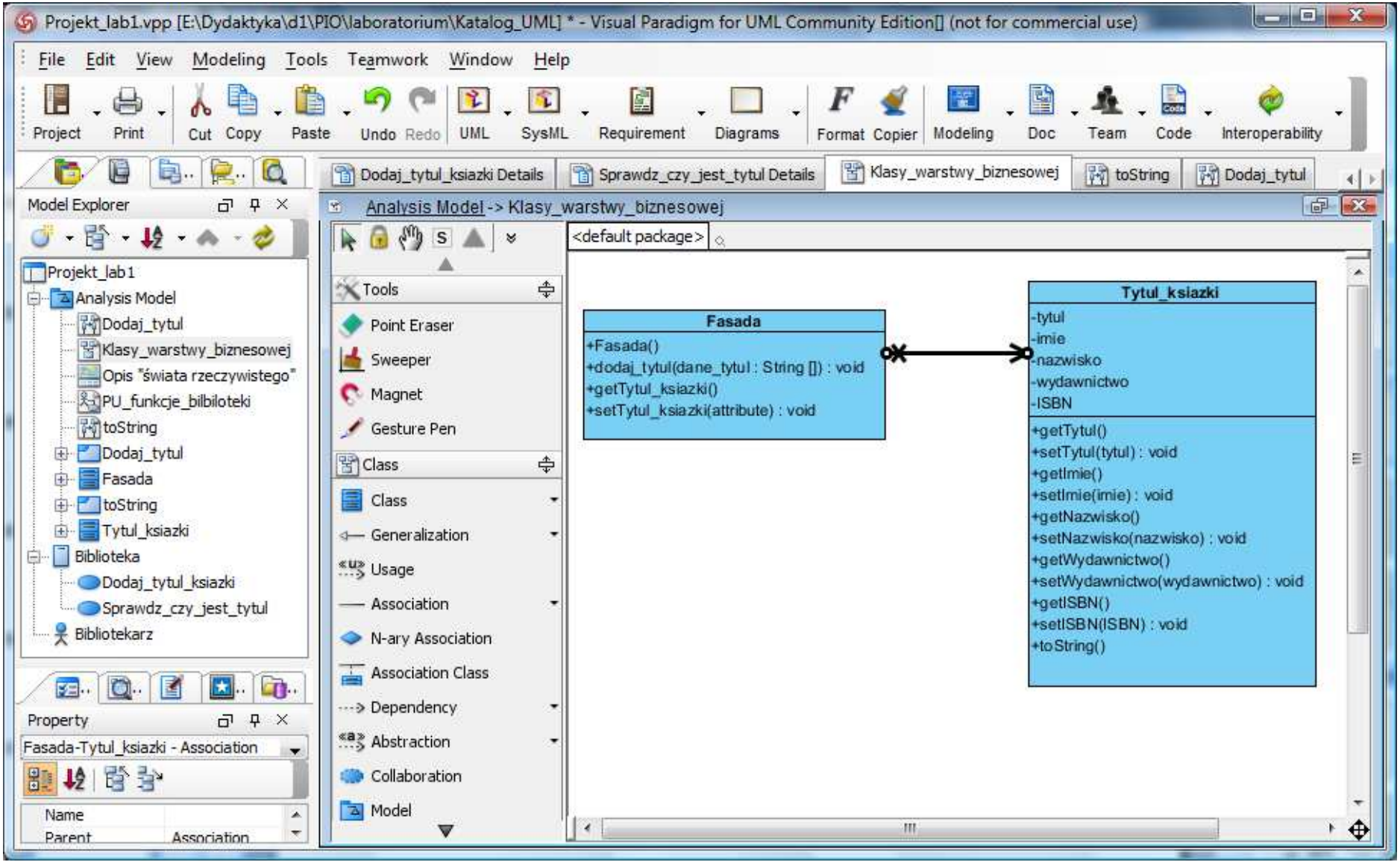
25.2. Definicja powiązania jednokierunkowego typu *Association* – należy kliknąć prawym klawiszem na koniec relacji powiązanej z klasą **Tytul_ksiazki** i wybrać z list opcje *Navigable/True*



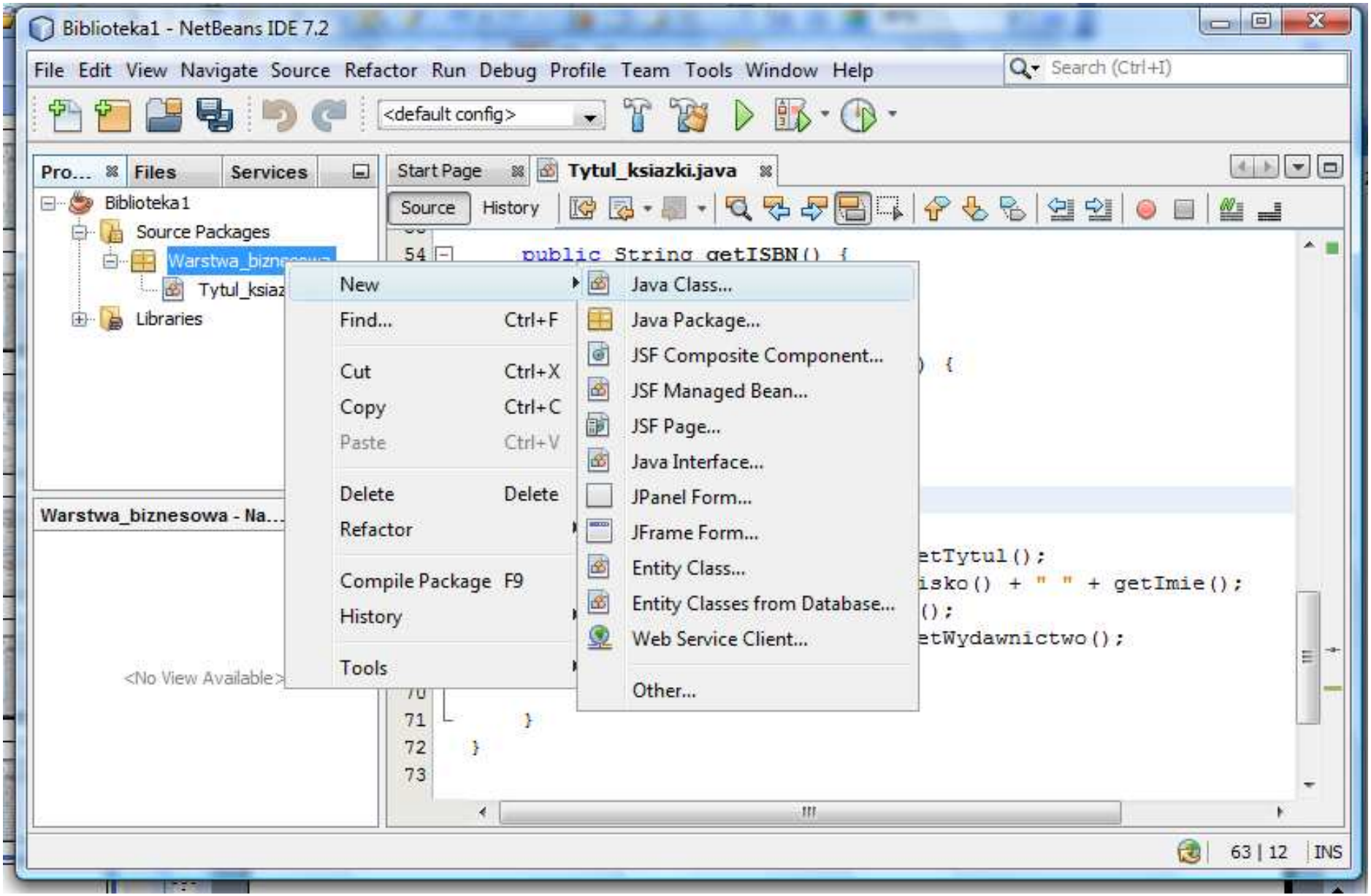
25.3. Definicja powiązania jednokierunkowego typu *Association* – należy kliknąć prawym klawiszem na koniec relacji powiązanej z klasą **Fasada** i wybrać z list opcje *Navigable/False*



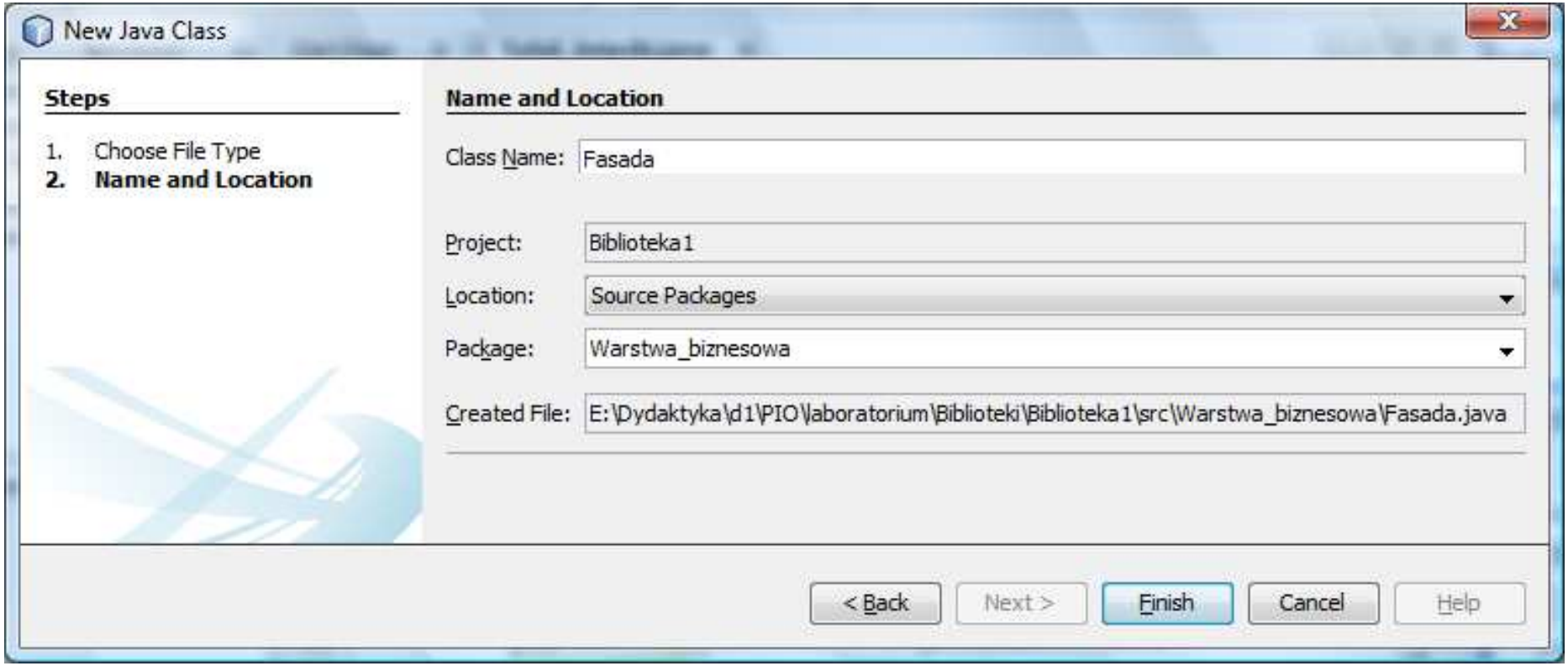
25.4. Uzyskano jednokierunkową relację typu *Association* – oznacza ona, że obiekt typu **Fasada** zawiera referencję do obiektu typu **Tytul_ksiazk**i, natomiast obiekt typu **Tytul_ksiazki** nie ma dostępu do obiektu klasy **Fasada** (definicja relacji jednokierunkowej typu 1..0..1)



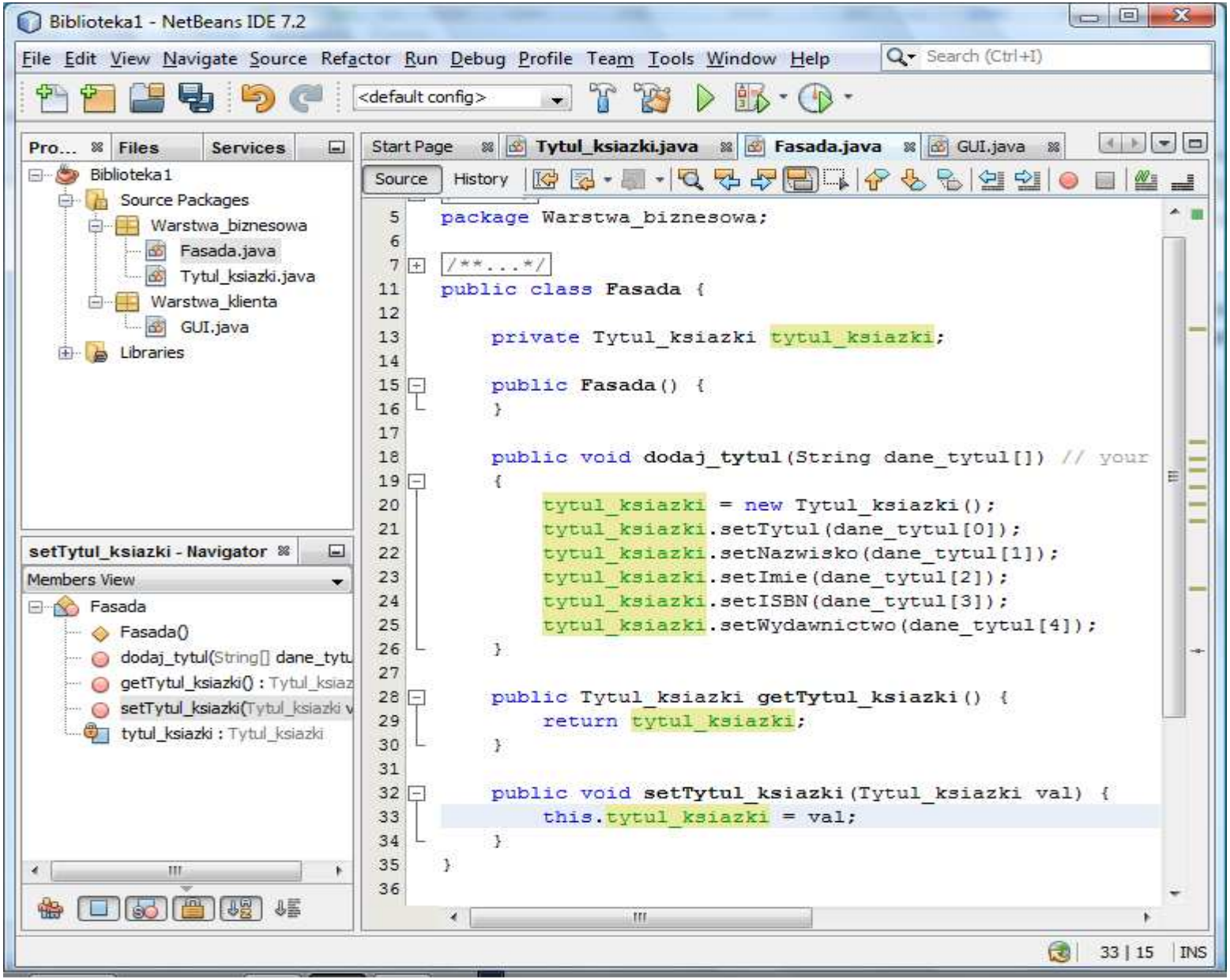
26.1. Uzupełnienie kodu projektu **Java Application** zawierającego implementację diagramów sekwencji **dodaj_tytul** – należy dodać nową klasę typu **Fasada** (podobnie jak **Tytul_ksiazki**)



26.2. Nadanie nazwy nowej klasie w polu *Class Name*



26.3. Dodanie kodu metody **dodaj_tytul** do klasy **Fasada** wg diagramu z p.24.5 oraz metody typu set i get dla rerefencji typu **Tytul_ksiazki**

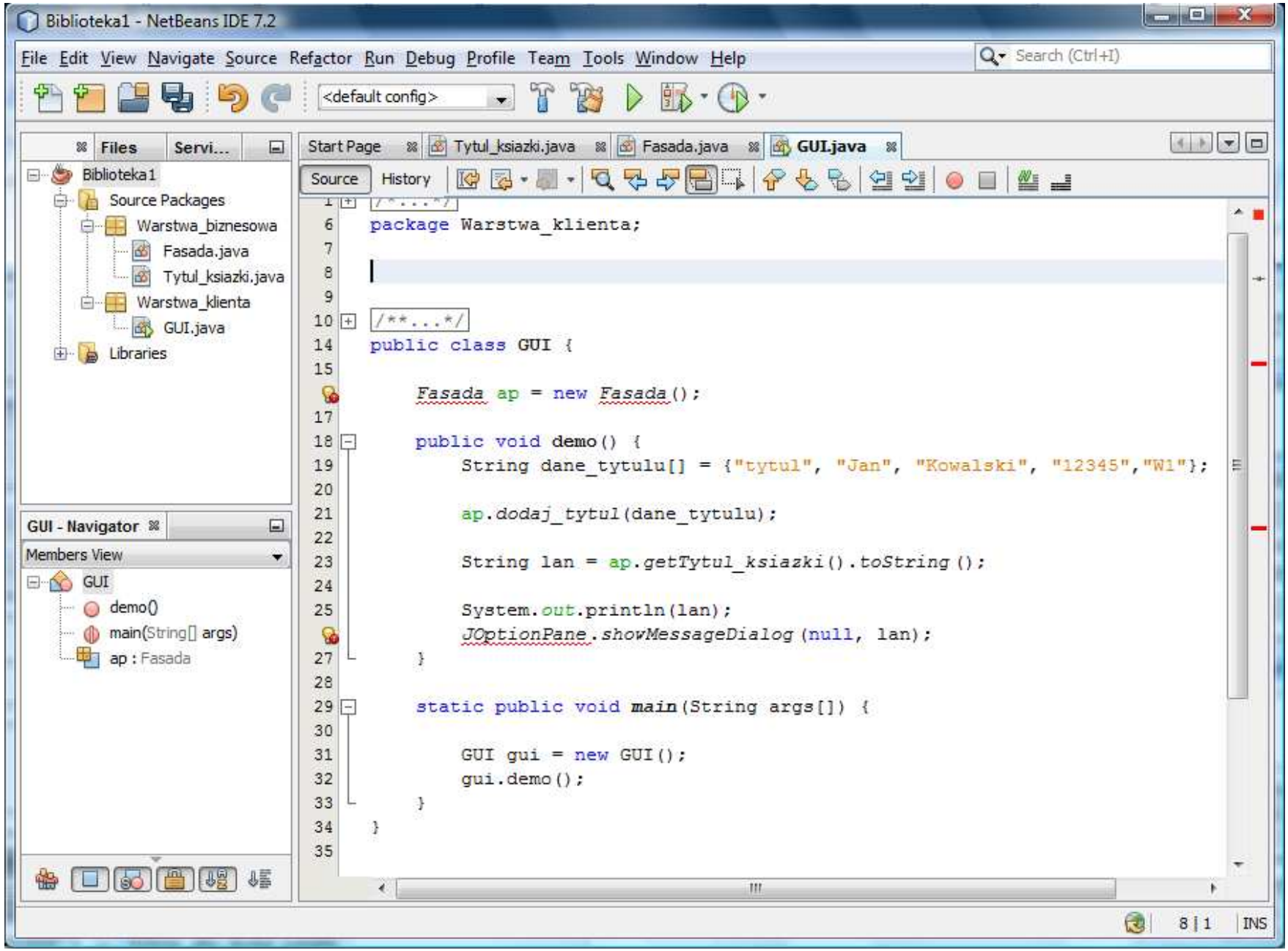


26.4. Kod klay **Fasada**

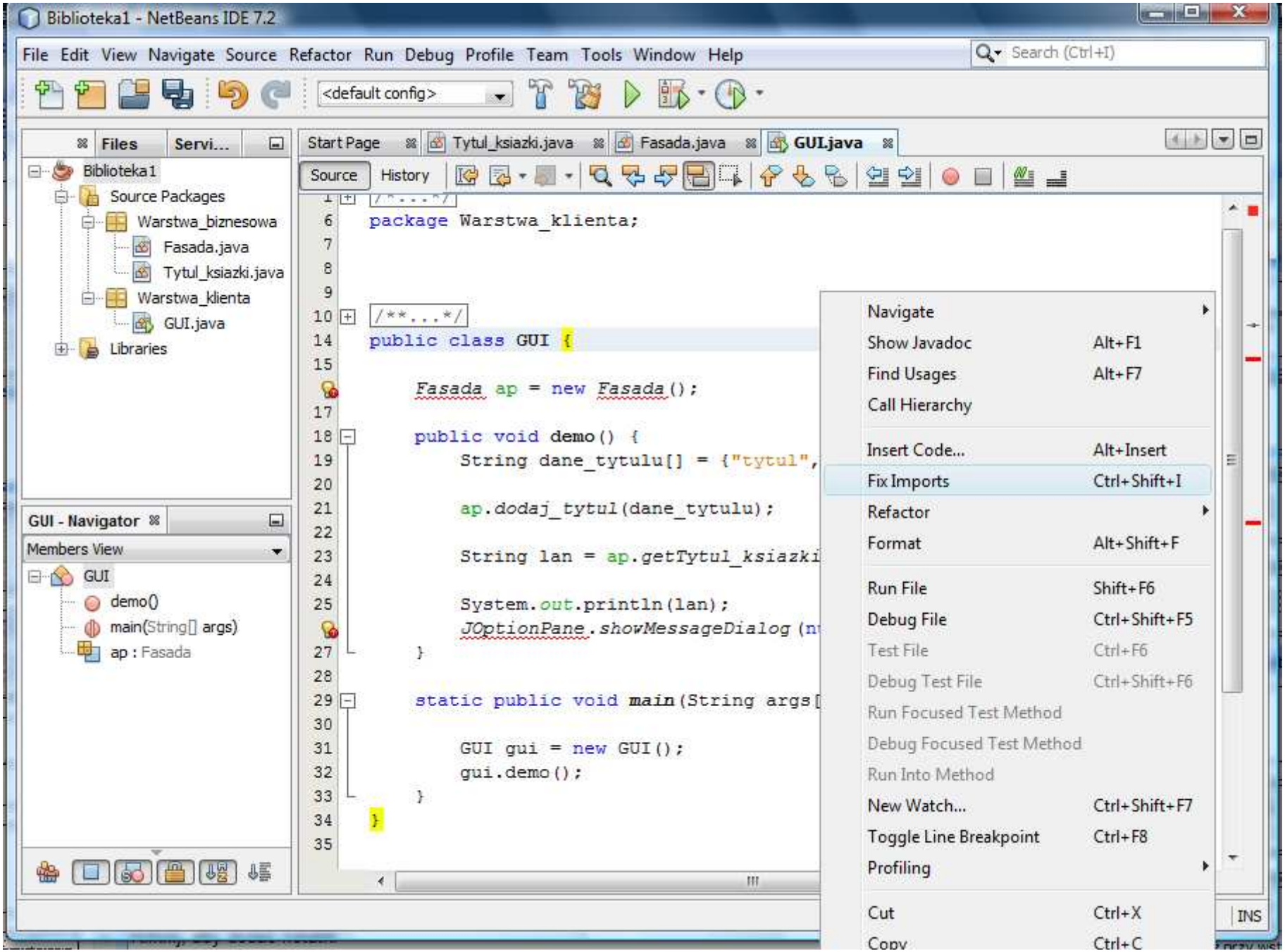
Referencja do obiektu klasy Tytul_ksiazki reprezentuje relację 1 do 1 po stronie klasy Fasada, która jest „właścicielem” relacji

```
package Warstwa_biznesowa;

public class Fasada {
    private Tytul_ksiazki tytul_ksiazki;
    public Fasada() {
    }
    public void dodaj_tytul(String dane_tytul[]) // your code here
    {
        tytul_ksiazki = new Tytul_ksiazki();   //implementacja Create Message
        tytul_ksiazki.setTytul(dane_tytul[0]);
        tytul_ksiazki.setNazwisko(dane_tytul[1]);
        tytul_ksiazki.setImie(dane_tytul[2]);
        tytul_ksiazki.setISBN(dane_tytul[3]);
        tytul_ksiazki.setWydawnictwo(dane_tytul[4]);
    }
    public Tytul_ksiazki getTytul_ksiazki() {
        return tytul_ksiazki;
    }
    public void setTytul_ksiazki(Tytul_ksiazki val) {
        this.tytul_ksiazki = val;
    }
}
```

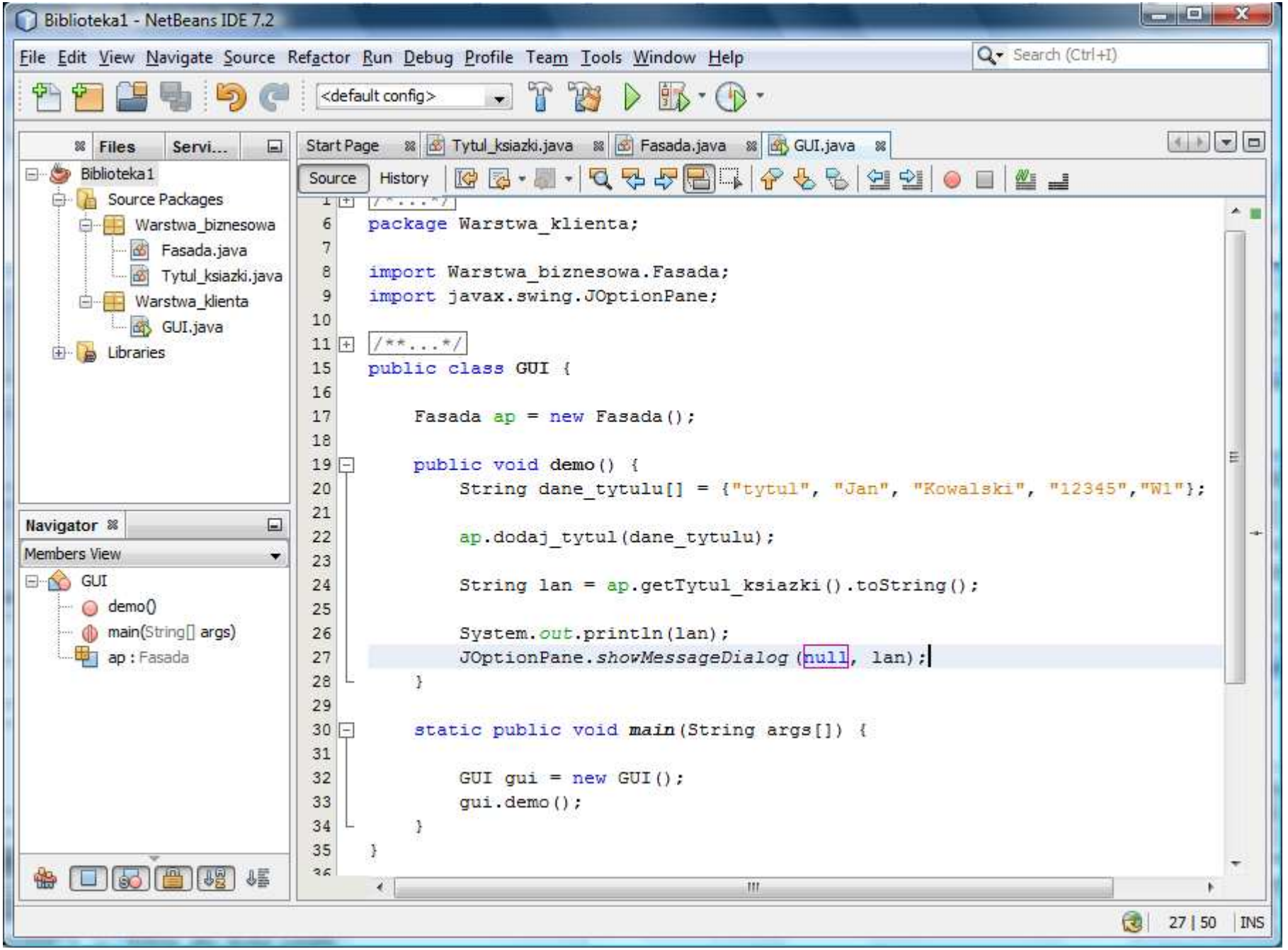
27.1. Kod klasy **GUI** reprezentujący warstwę klienta (interfejs graficzny użytkownika) z pakietu **Warstwa_klienta**



27.2. Definicja klasy **GUI** korzystającej z metod klasy **Fasada** z warstwy biznesowej



27.3. Kod klasy **GUI** po uzupełnieniu importów pakietu **Warstwa_biznesowa**



27.4. Kod klasy **GUI** korzystającej z metod klasy **Fasada**

```
package Warstwa_klienta;

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

public class GUI {
   Fasada ap = new Fasada();

   public void demo() {
      String dane_tytulu[] = {"tytul", "Jan", "Kowalski", "12345","W1"};

      ap.dodaj_tytul(dane_tytulu);

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

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

   static public void main(String args[]) {
      GUI gui = new GUI();
      gui.demo();
   }
}
```

28. Uruchomienie programu – metoda **demo** z klasy GUI pozwaka przetestować metody klasy **Fasada** i **Tytul_ksiazki**