

Wykład 3 – część druga

Iteracyjno-rozwojowy cykl oprogramowania 3

Wstawienie nowego przypadku użycia `Dodaj_książke` w relacji `<<usage>>` i wstawienie opisu w polu `Documentation`

The screenshot shows the NetBeans IDE interface with a UML Use Case diagram. The diagram features an actor named `Bibliotekarz` connected to two use cases: `Dodaj_tytul` and `Dodaj_książke`. Both relationships are labeled with `<<usage>>`. The `Dodaj_książke` use case is currently selected, indicated by a dashed blue border.

In the foreground, a dialog box titled `Dodaj_książke - Documentation` is open, containing the following text:

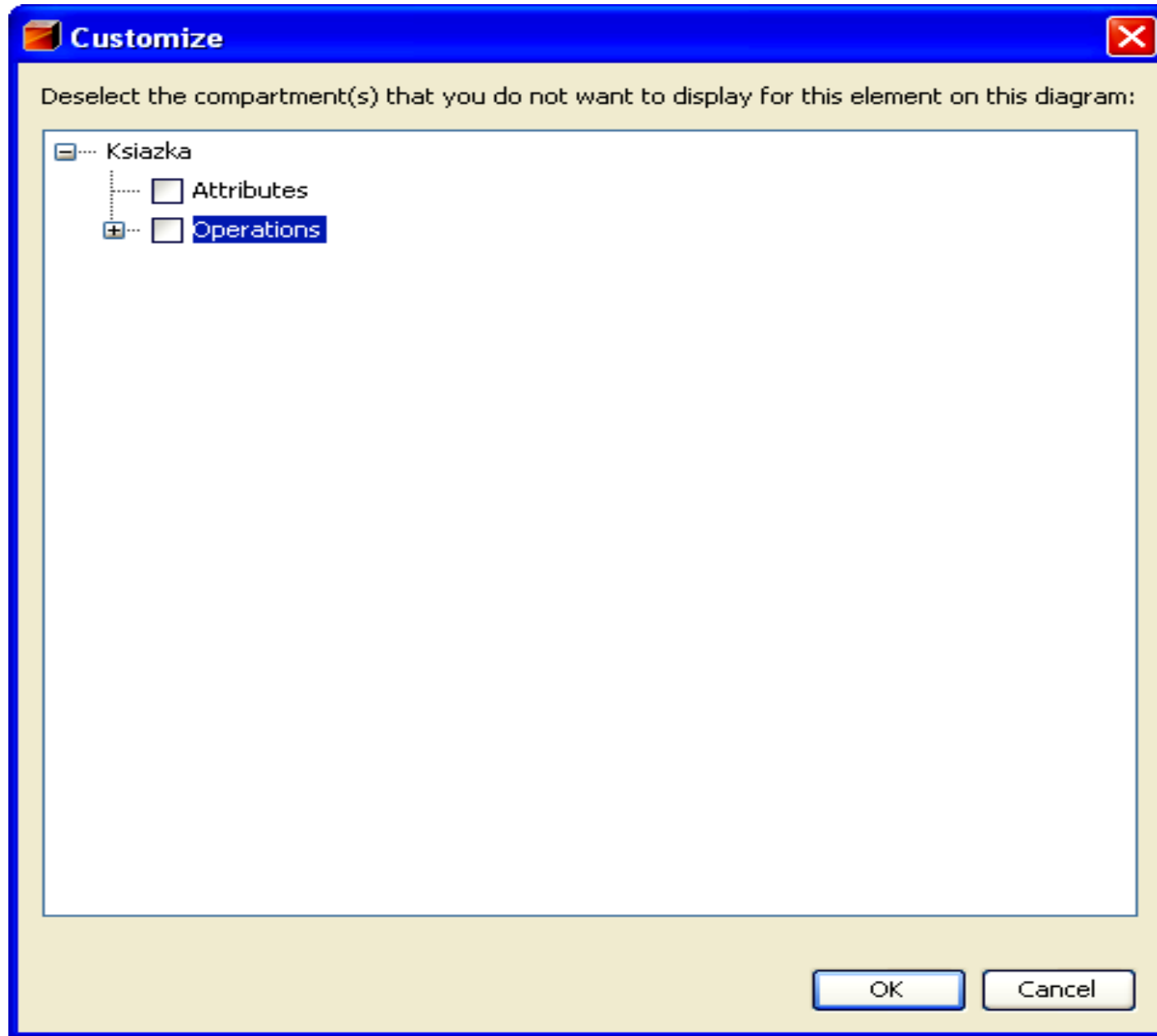
```
<ol>1.Należy sprawdzić, czy istnieje dany tytuł książki  
2. Jeśli istnieje, można wstawić kolejny egzemplarz tej książki z unikatowym numerem  
3. Jeśli nie ma takiego tytułu, nie można wstawić nowej książki  
</ol>
```

The background interface includes a `Palette` on the right with categories like `Basic`, `Dependencies`, and `Comments`. The `Properties` window for `Dodaj_książke` shows the `Documentation` field with the value `1.Należy spraw...`. The `Navigator` and `Outline` panes on the left show the project structure, including a `Model` package with various classes and use cases.

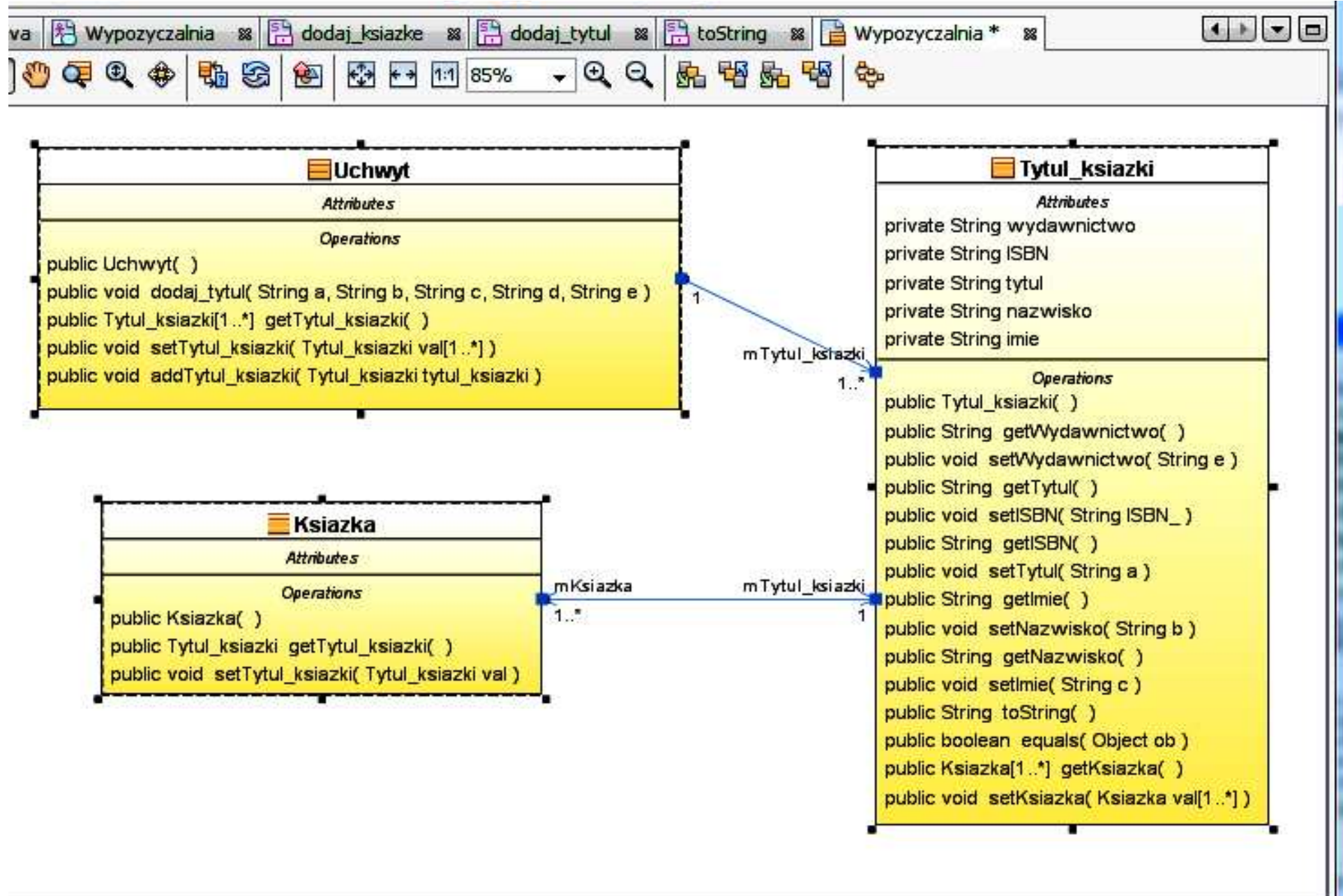
Wstawienie nowej klasy **Ksiazka**, która jest w relacji 1 do wiele z klasą **Tytul_ksiazki** i ukrycie atrybutów i operacji w klasach za pomocą opcji **Compartment**

The screenshot shows the NetBeans IDE interface with a UML class diagram. The diagram includes three classes: **Uchwyt**, **Tytul_ksiazki**, and **Ksiazka**. **Uchwyt** is associated with **Tytul_ksiazki** (multiplicity 1 to 1..*). **Tytul_ksiazki** is associated with **Ksiazka** (multiplicity 1 to 0..*). A context menu is open over the **Ksiazka** class, with the **Compartment** option selected. The **Ksiazka - Properties** window is also visible, showing the class name and its properties. The bottom status bar shows the system tray with the time 22:31.

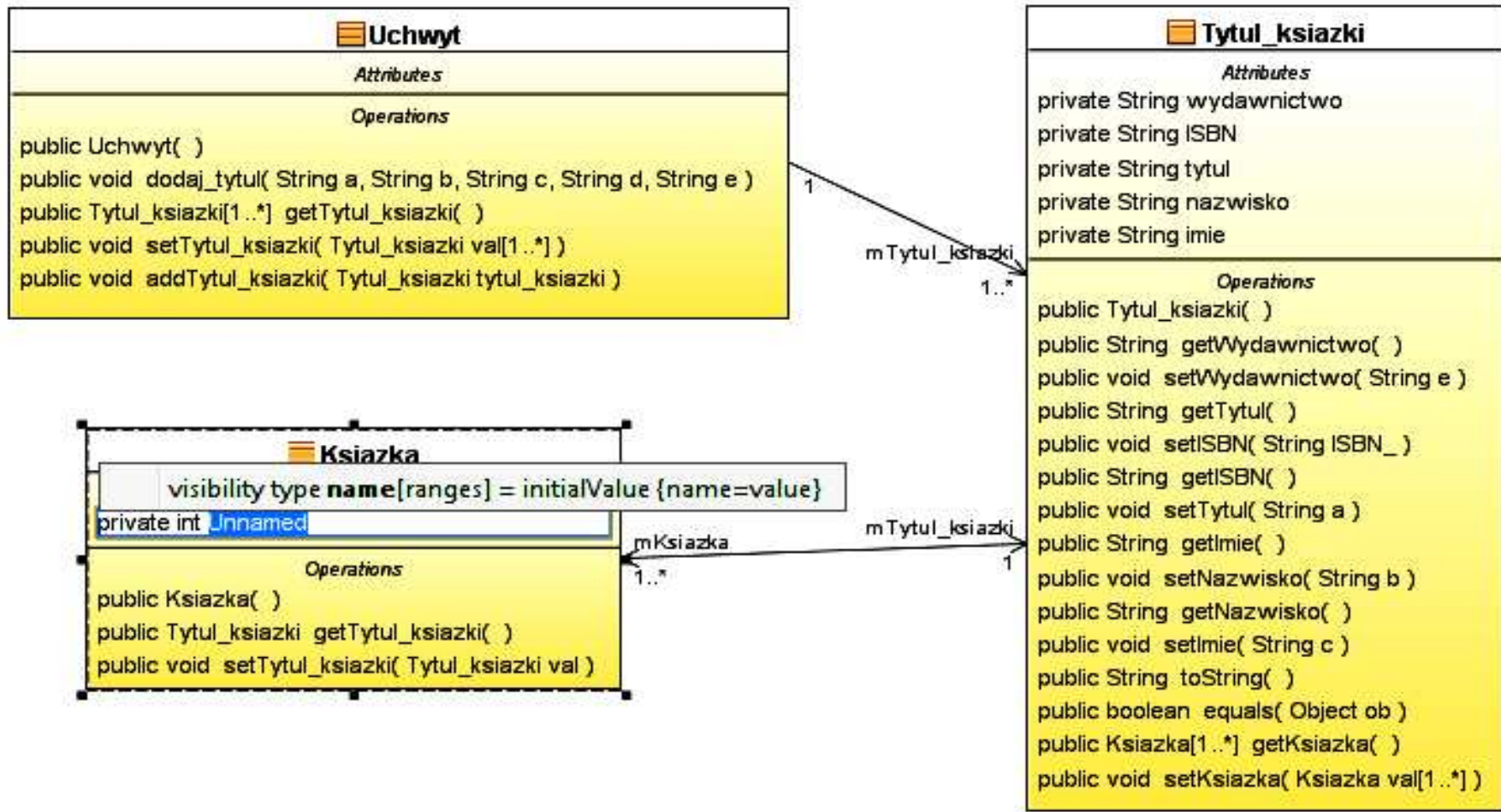
Ukrycie atrybutów i operacji w wybranej klasie – edytor opcji **Compartment**

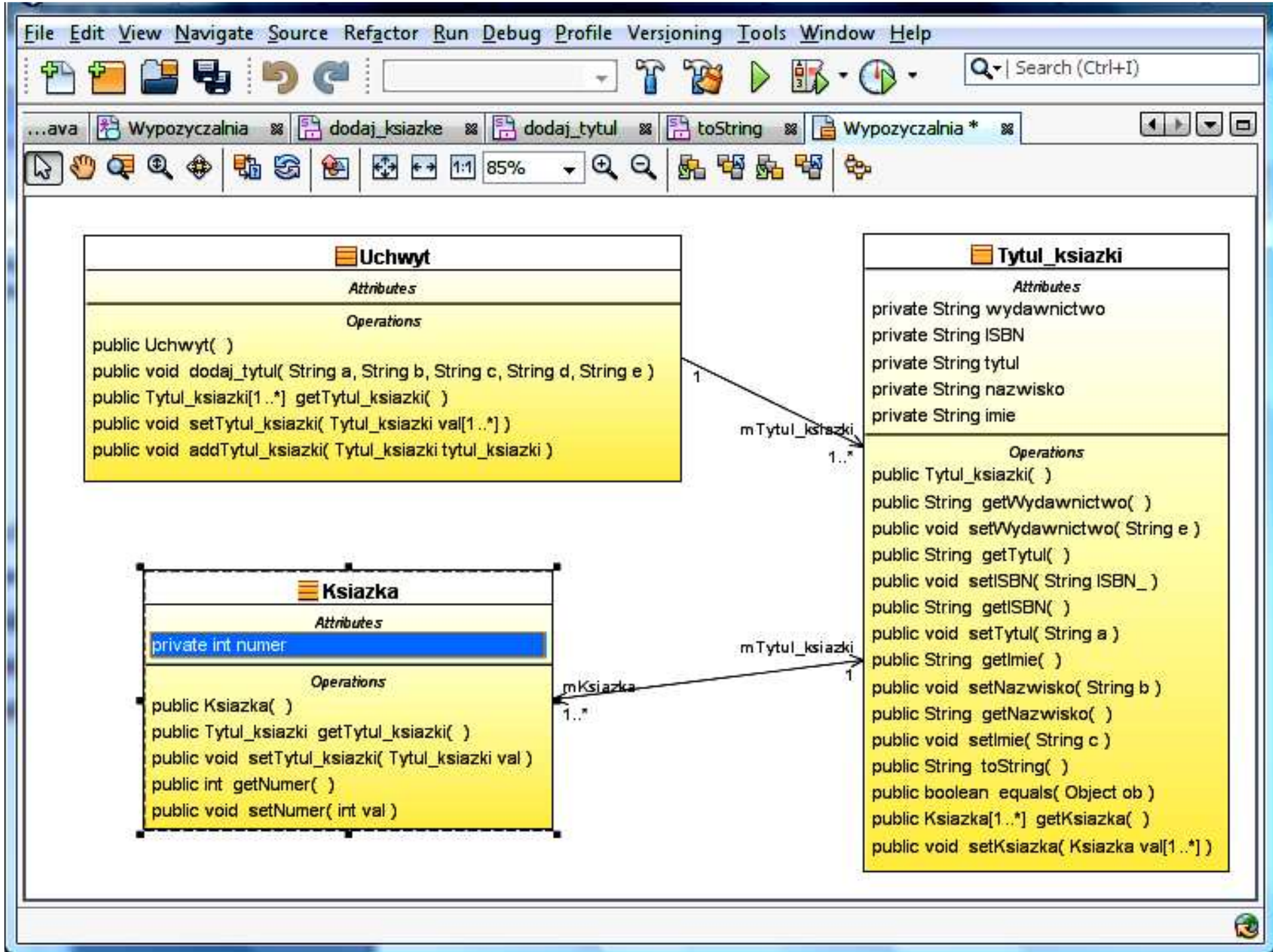


Połączenie dwukierunkową (oba końce relacji typu `Navigable`) relacją 1 do wiele klasy `Książka` z klasą `Tytul_książki`

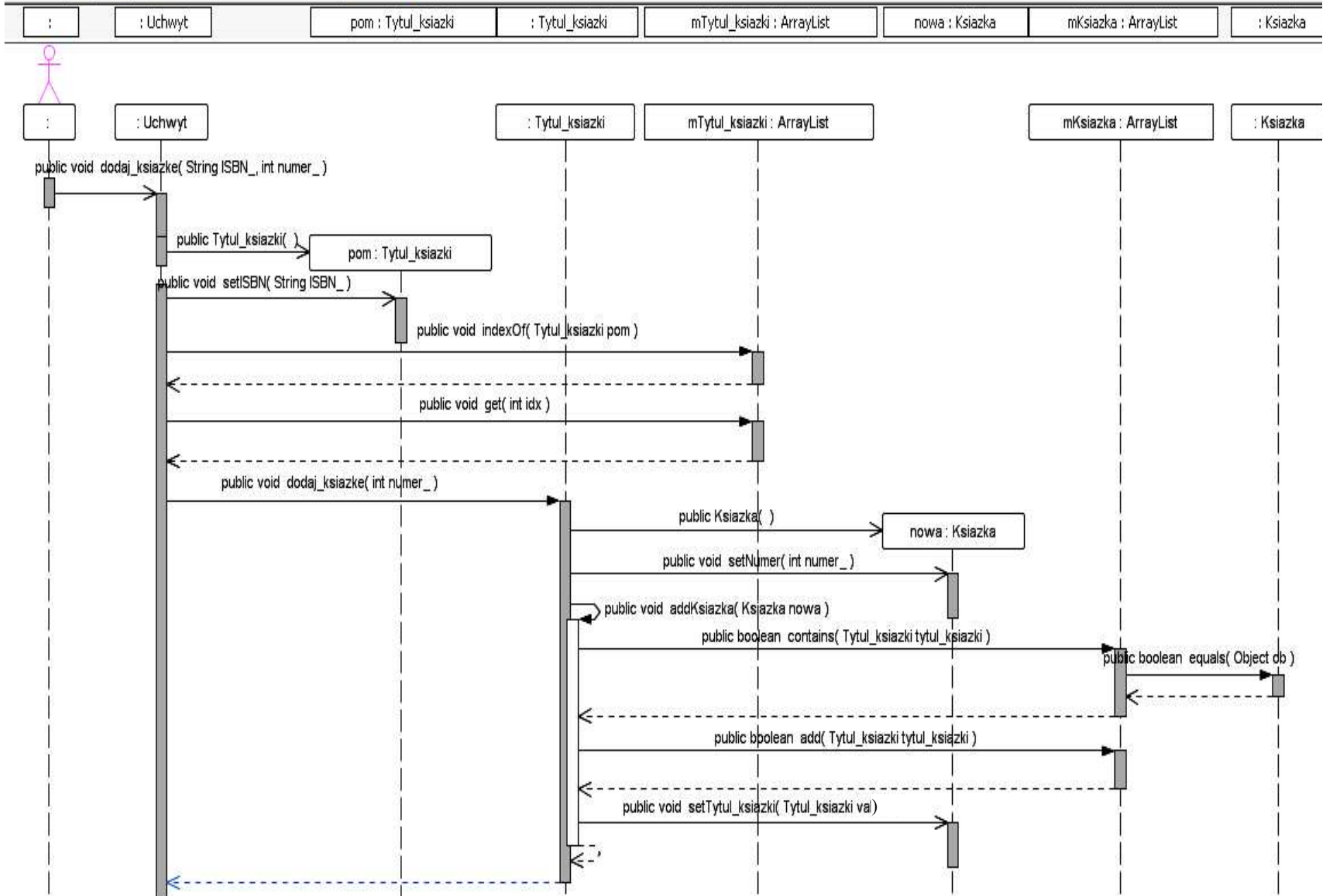


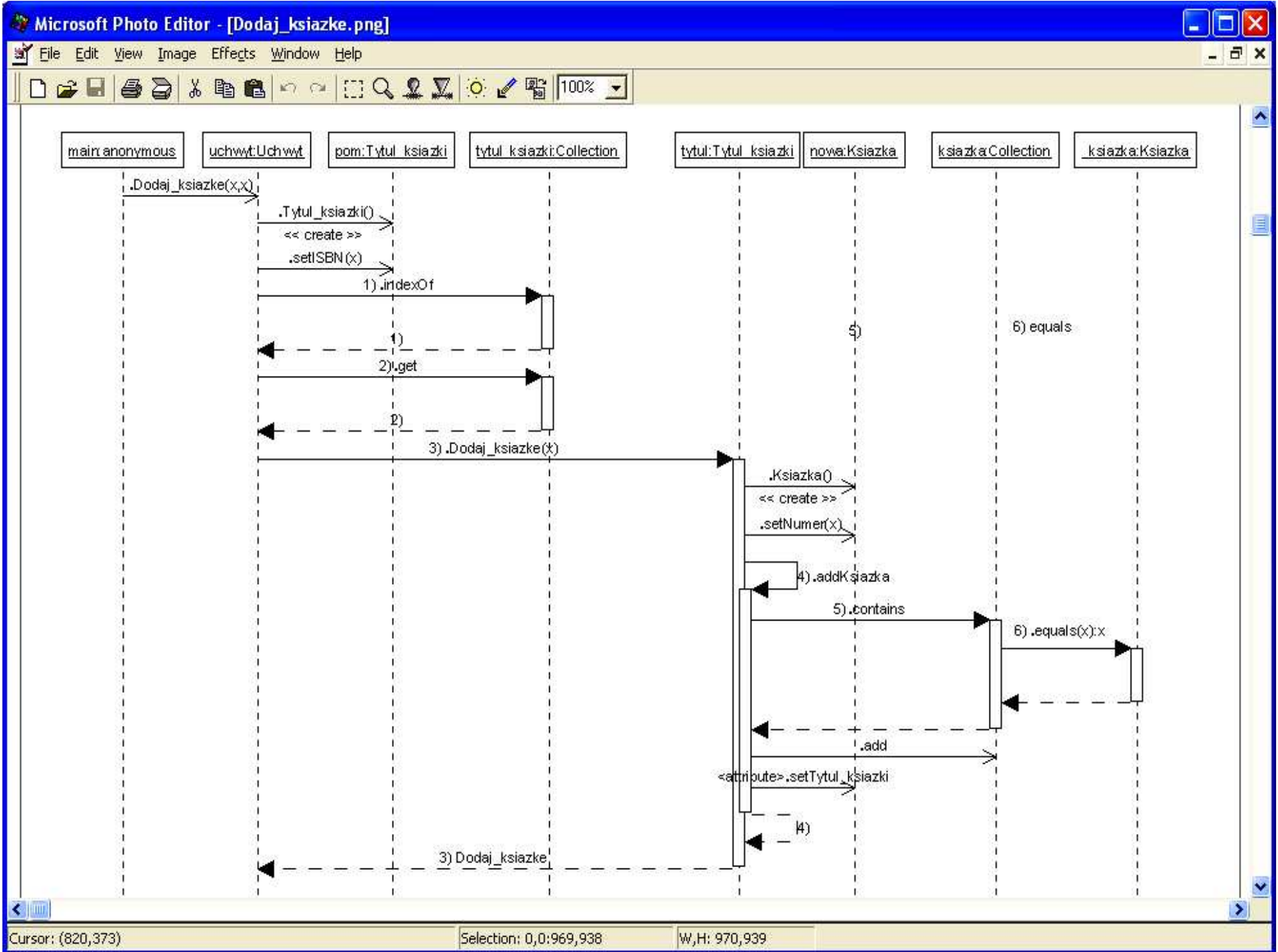
Wstawienie nowego atrybutu do klasy **Ksiazka**



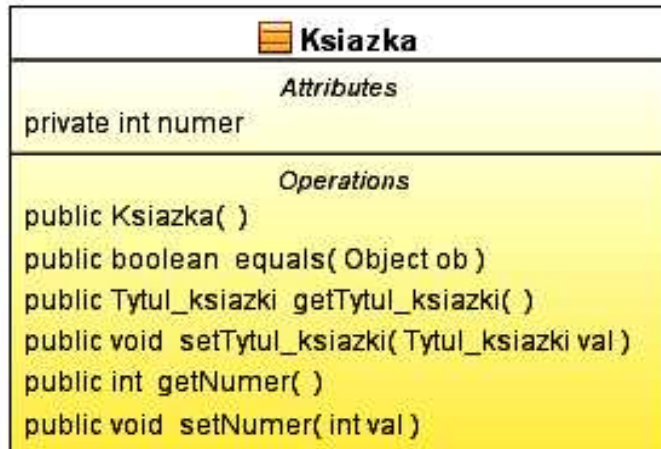
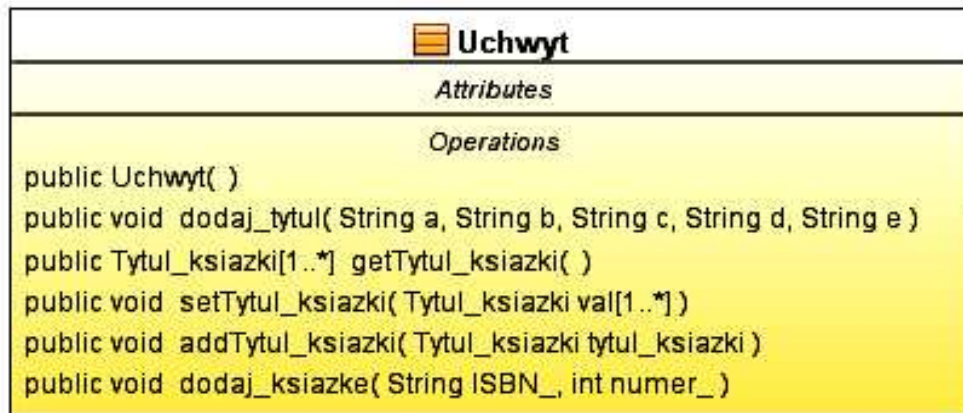


Definicja przypadku użycia dodaj_ksiazke – oparty na implementacji relacji 1 do n między klasą Uchwyt i klasą Tytul_ksiazki oraz klasą Tytul_ksiazki i klasą Ksiazka





Definicja klas Uchwyt, Tytul_książki, Książka po dodaniu przypadku użycia dodaj_książke



Definicja klasy `Ksiazka` po wygenerowaniu i uzupełnieniu kodu

```
public class Ksiazka
{ private int numer;

private Tytul_ksiazki mTytul_ksiazki;

public int getNumer()
    { return numer;}

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

public Ksiazka()
    { }

public boolean equals(Object ob)           // your code here
    { return numer==((Ksiazka)ob).getNumer(); }
```

```
public Tytul_książki getTytul_książki()  
    { return mTytul_książki;}
```

```
public void setTytul_książki(Tytul_książki val)  
    { mTytul_książki = val; }
```

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


Definicja klasy `Tytul_książki` po wygenerowaniu i uzupełnieniu kodu

```
public class Tytul_książki
{
    private String wydawnictwo;
    private String ISBN;
    private String tytul;
    private String nazwisko;
    private String imie;
    private ArrayList<Książka> mKsiążka=
        new java.util.ArrayList <Książka>();

    public Tytul_książki()
    {
    }

    public String getWydawnictwo()
    {
        return wydawnictwo;
    }

    public void setWydawnictwo(String e)
    {
        wydawnictwo = e;
    }
}
```

```
public String getISBN()  
    { return ISBN;    }  
public void setISBN(String ISBN_)  
    { ISBN = ISBN_;    }  
public String getTytul()  
    { return tytul;}  
public void setTytul(String a)  
    { tytul = a;    }  
public String getNazwisko()  
    { return nazwisko; }  
public void setNazwisko(String b)  
    { nazwisko = b; }  
public String getImie()  
    { return imie;    }  
public void setImie(String c)  
    { imie = c;    }
```

```
public boolean equals(Object ob) // your code here
{ boolean a;
  a=ISBN.equals(((Tytul_ksiazki)ob).getISBN());
  System.out.println(a); //linia tymczasowa
return a; }
```

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

```
public void Dodaj_ksiazke(int numer_) // your code here
{ Ksiazka nowa= new Ksiazka();
  if (nowa != null)
  { nowa.setNumer(numer_);
    addKsiazka(nowa); }
}
```

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

```
public ArrayList<Ksiazka> getKsiazka()
{ return mKsiazka; }
```

```
public void setKsiazka(ArrayList<Ksiazka> val)
{ this.mKsiazka = val; }
```

```
} //koniec Tytul_ksiazki
```


Definicja klasy `Uchwyt` po wygenerowaniu i uzupełnieniu kodu

```
public class Uchwyt
```

```
{ private ArrayList<Tytul_książki> mTytul_książki=  
    new ArrayList<Tytul_książki>();
```

```
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)
```

```
{ mTytul_książki = val; }
```

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

```
{ if (! mTytul_książki.contains(tytul_książki))  
    mTytul_książki.add(tytul_książki);}
```

```
public void Dodaj_książke(String ISBN_, int numer_) // your code here
```

```
{ Tytul_książki pom=new Tytul_książki();  
    pom.setISBN(ISBN_);  
    int idx=mTytul_książki.indexOf(pom);  
    if (idx!=-1)  
        { Tytul_książki pom1=mTytul_książki.get(idx);  
            pom1.dodaj_książke(numer_);  
            System.out.println(pom1.getKsiążka().toString()); } //linia tymczasowa  
}
```

```
public static void main(String t[])           //kod dodany
{
    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);

    ap.Dodaj_ksiazke("1",1);
    ap.Dodaj_ksiazke("1",2);
    ap.Dodaj_ksiazke("1",2);
    ap.Dodaj_ksiazke("2",1);
}
} //koniec klasy Uchwyt
```

Projects Files Runtime ...va Ksiazka.java x Uchwyt.j... Dynamic Help

Katalog4
Source Packages

Projects Window
See Also

Output

Generate Code Log x Katalog4 (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]
true
[Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 1]
true
[Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 1, Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 2]
true
[Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 1, Tytul: 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 2]
false
true
[Tytul: 2 Autor:2 2 ISBN: 2 Wydawnictwo:2 Numer: 1]
BUILD SUCCESSFUL (total time: 7 seconds)
```

Output HTTP Monitor