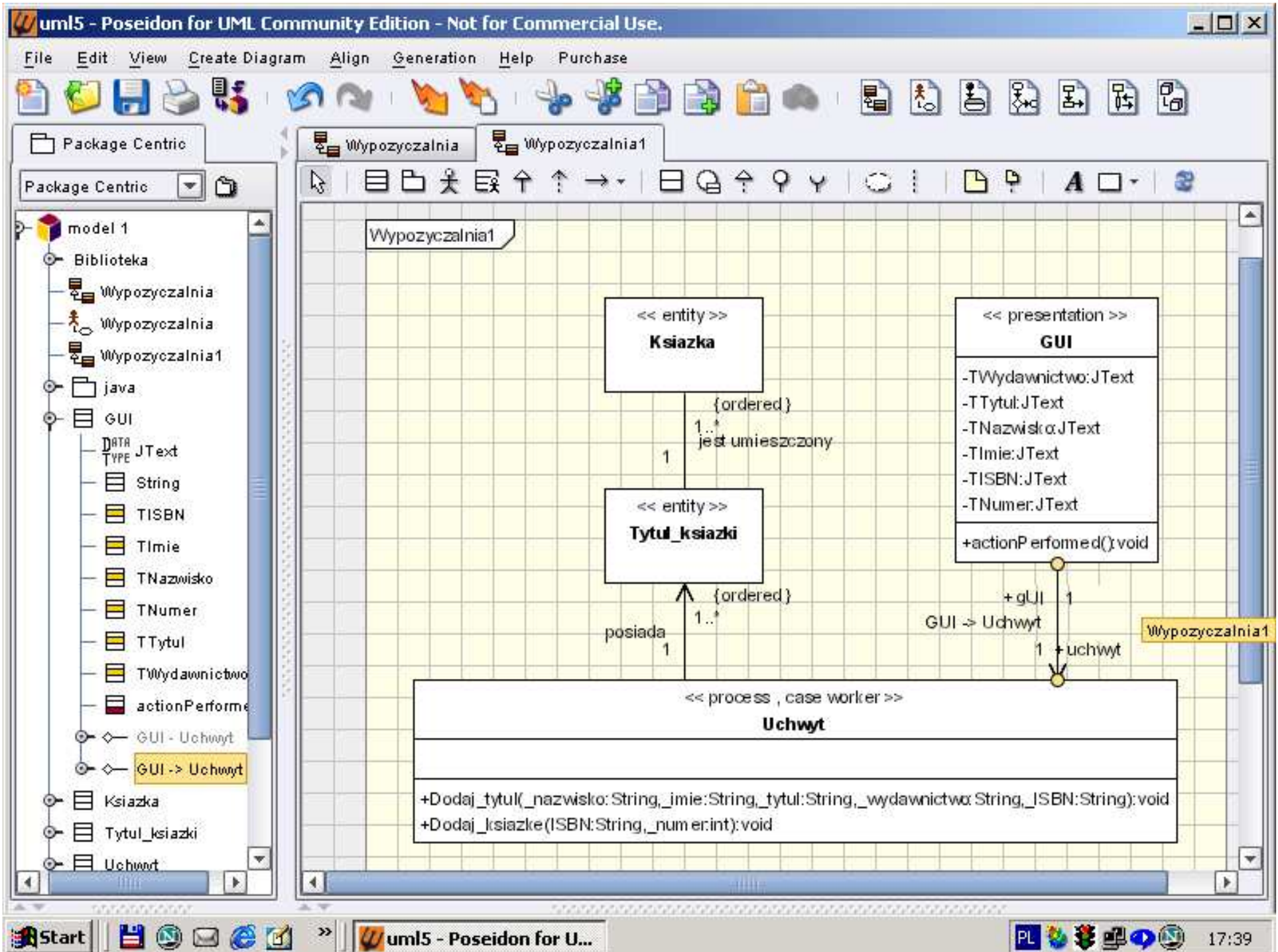


Wykład 4_2 – część druga

Iteracyjno-rozwojowy cykl oprogramowania 4



uml5 - Poseidon for UML Community Edition - Not for Commercial Use.

File Edit View Create Diagram Align Generation Help Purchase

Package Centric

Package Centric

model 1

- Biblioteka
 - Wypożyczalnia
 - Wypożyczalnia
 - Wypożyczalnia1
- java
- GUI
- Książka
- Tytuł_książki
- Uchwył
- Interface_1
- Bibliotekarz
- Collaboration_1
- Collaboration_1
- Interaction_1

Wypożyczalnia1 Dodaj_książkę Przypadki_użycia

Prac1:Bibliotekarz gui:GUI uchwyłUchwył

.actionPerformed()

.Dodaj_tytul(x,x,x,x)

.Dodaj_książkę(x,x)

Class

Properties Style Documentation SourceCode Constraints Tagged Values

Name Interaction_1

Namespace

Visibility public protected package private

Modifiers abstract static final root active

<< >> Stereotypes ...

Operations

Extends

Start uml5 - Poseidon for U... PL 16:54

The screenshot displays the uml5 - Poseidon for UML Community Edition interface. The main workspace shows an interaction diagram with three lifelines: Prac1:Bibliotekarz (actor), gui:GUI, and uchwyłUchwył. The sequence of messages is: Prac1:Bibliotekarz sends .actionPerformed() to gui:GUI, which then sends .Dodaj_tytul(x,x,x,x) to uchwyłUchwył, which finally sends .Dodaj_książkę(x,x) to uchwyłUchwył. On the left, a Package Centric view shows a project structure with 'model 1' containing 'Biblioteka' and 'Interaction_1'. At the bottom, a Properties window for 'Interaction_1' is open, showing fields for Name, Namespace, Visibility (set to public), Modifiers, Stereotypes, and Operations.

I. Przykład pierwszy programu z warstwą klienta (interfejs graficzny użytkownika) i warstwą biznesową dostępną za pomocą metod klasy Uchwyt

Aplikacja UML

Tytuł książki 1

Nazwisko autora książki 1

Imie autora książki 1

ISBN tytułu 1

Wydawnictwo 1

Numer książki 3

Zapisz tytuł Zapisz książkę Wyswietl tytuły Wyswietl książki

Tytuł; 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 ▼ Tytuły książek

Tytuł; 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 1 ▼ Książki

Tytuł; 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 1

Tytuł; 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 2

Tytuł; 1 Autor:1 1 ISBN: 1 Wydawnictwo:1 Numer: 3

```

/** * <p></p> * */
public class GUI
{ /** * <p>Represents ...</p> * */
private JText TWydawnictwo; /** *<p>Represents ...</p>* */
private JText TTytul; /** *<p>Represents ...</p>* */
private JText TNazwisko; /** * <p>Represents ...</p>* */
private JText TImie; /** * <p>Represents ...</p>* */
private JText TISBN; /** * <p>Represents ...</p>* */
private JText TNumer; /** * <p>Does ...</p> * */
public void actionPerformed() /** * <p></p> * */
    { // your code here }
public Uchwyt uchwyty_1; /** * <p></p> * */
public class String { /** * @poseidon-generated */
public Uchwyt getUchwyt_1() /** * @poseidon-generated */
    { return uchwyty_1; }
public void setUchwyt_1(Uchwyt uchwyty)
    { this.uchwyty_1 = uchwyty; }
}

```

// kod napisany przez programistę

```
public class Baza
{public Uchwyt uchwyty=new Uchwyt();
public GUI gui;
static public void main(String arg[])
{ Baza baza = new Baza();
  try
  { baza.gui = new GUI(baza.uchwyty);
    baza.gui.setVisible(true);
  } catch(Exception e)
  { System.out.println("Blad bazy "+e);}
}
}
```

Modyfikacja definicji klasy Tytul_ksiazki – dodanie pomocniczej metody **ksiazki()** stosowanej do prezentacji książek.

```
public class Tytul_ksiazki {  
  
    private String wydawnictwo;  
    private String ISBN;  
    private String tytul;  
    private String nazwisko;  
    private String imie;  
    private ArrayList<Ksiazka> mKsiazka = new java.util.ArrayList<Ksiazka>();  
  
    public Tytul_ksiazki() { }  
    public String getWydawnictwo()           { return wydawnictwo; }  
    public void setWydawnictwo(String e)     { this.wydawnictwo = e; }  
    public String getTytul()                 { return tytul; }  
    public void setISBN(String ISBN_)       { this.ISBN = ISBN_; }  
    public String getISBN()                  { return ISBN; }  
    public void setTytul(String a)          { this.tytul = a; }  
    public String getImie()                  { return imie; }  
    public void setNazwisko(String b)       { this.nazwisko = b; }  
    public String getNazwisko()              { return nazwisko; }  
    public void setImie(String c)           { this.imie = c; }  
  
    public ArrayList<Ksiazka> getKsiazka()  { return mKsiazka; }  
    public void setKsiazka(ArrayList<Ksiazka> val) { this.mKsiazka = val; }  
}
```

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

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

public void dodaj_ksiazke(int numer_) {
    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);
    }
}
```


Nowa metoda **ksiazki()** w klasie Tytul_ksiazki – zwraca kolekcję obiektów typu **String**, każdy z nich zawierający łańcuch zwracany przez metodę **toString** elementu kolekcji **mKsiazka**

```
public ArrayList<String> ksiazki() {  
    ArrayList<String> ksiazki = new ArrayList<String>();  
    Iterator<Ksiazka> it = mKsiazka.iterator();  
    while (it.hasNext()) {  
        ksiazki.add(it.next().toString());  
    }  
    return ksiazki;  
}
```

Zmiana definicji klasy Uchwył - dodane metody **tytul() oraz **ksiazki()** do klasy Uchwył stosowane do prezentacji tytułów i książek.**

```
public class Uchwył {  
  
    private ArrayList<Tytuł_ksiazki> mTytuł_ksiazki = new ArrayList<Tytuł_ksiazki>();  
  
    public Uchwył() { }  
  
    public void dodaj_tytuł(String a, String b, String c, String d, String e) {  
        Tytuł_ksiazki tytuł_ksiazki = new Tytuł_ksiazki();  
        tytuł_ksiazki.setTytuł(a);  
        tytuł_ksiazki.setNazwisko(b);  
        tytuł_ksiazki.setImię(c);  
        tytuł_ksiazki.setISBN(d);  
        tytuł_ksiazki.setWydawnictwo(e);  
        addTytuł_ksiazki(tytuł_ksiazki);  
    }  
  
    public ArrayList<Tytuł_ksiazki> getTytuł_ksiazki() { return mTytuł_ksiazki; }  
    public void setTytuł_ksiazki(ArrayList<Tytuł_ksiazki> val) { this.mTytuł_ksiazki = val; }
```

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

```
public void dodaj_książke(String ISBN_, int numer_) {  
    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  
}
```

Nowe metody w klasie **Uchwyt** – zwracają kolekcję obiektów typu String. Metoda **tytuly()** zwraca kolekcję łańcuchów, każdy z nich zawierający łańcuch zwracany przez metodę toString elementu kolekcji mTytul_książki. Metoda **ksiazki()** zwraca kolekcję łańcuchów, reprezentującą sumę kolekcji książek każdego z tytułów, zwracanych metodą **ksiazki()** z klasy typu **Tytul_książki**.

```
public ArrayList<String> tytuly() {
    ArrayList<String> tytuly = new ArrayList<String>();
    Iterator<Tytul_książki> it = mTytul_książki.iterator();
    while (it.hasNext()) {
        tytuly.add(it.next().toString());
    }
    return tytuly;
}

public ArrayList<String> ksiazki() {
    ArrayList<String> ksiazki = new ArrayList<String>();
    Iterator<Tytul_książki> it = mTytul_książki.iterator();
    while (it.hasNext()) {
        ksiazki.addAll(it.next().ksiazki()); //łączenie kolekcji łańcuchów
    } //reprezentujących dane
    return ksiazki; //zbioru książek kolejnego
} // tytułu, zwracanymi metodą
} //ksiazki()
```

```
import java.sql.*;
import javax.swing.*;
import java.util.*;
import java.io.*;
import java.lang.*;
import java.awt.event.*;
```

```
public class GUI extends JFrame
```

```
    implements ActionListener
```

```
{ JLabel elista_tytulow = new JLabel ("Tytuly ksiazek");
  JComboBox tytuly = new JComboBox();
  JLabel elista_ksiazek = new JLabel ("Ksiazki");
  JComboBox ksiazki = new JComboBox();
  JLabel etytul = new JLabel ("          Tytul ksiazki");
  JTextField Ttytul = new JTextField(30);
  JLabel enazwisko = new JLabel ("Nazwisko autora ksiazki");
  JTextField TNazwisko = new JTextField(30);
```

```
JLabel eimie=    new JLabel ("        Imie autora ksiazki");
JTextField TImie = new JTextField(30);
JLabel eISBN=  new JLabel ("        ISBN tytułu");
JTextField TISBN= new JTextField(30);
JLabel ewydawnictwo =
        new JLabel ("        Wydawnictwo");
JTextField TWydawnictwo=new JTextField(30);
JLabel enumer=
        new JLabel ("        Numer ksiazki");
JTextField TNumer= new JTextField(30);
JButton zapisz_tytul= new JButton("Zapisz tytul");
JButton zapisz_ksiazke= new JButton("Zapisz ksiazke");
JButton wyswietl_tytuly= new JButton("Wyswietl tytuly");
JButton wyswietl_ksiazki= new JButton("Wyswietl ksiazki");
Uchwyt dane;
```

GUI(Uchwyt uchwyt)

```
{ super("Aplikacja UML");  
  setSize(500,350);  
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  JPanel panel= new JPanel();  
  panel.add(etytul);  
  panel.add(TTytul);  
  panel.add(enazwisko);  
  panel.add(TNazwisko);  
  panel.add(eimie);  
  panel.add(TImie);  
  panel.add(eISBN);  
  panel.add(TISBN);  
  panel.add(ewydawnictwo);  
  panel.add(TWydawnictwo);  
  panel.add(enumer);  
  panel.add(TNumer);
```

```
zapisz_tytul.addActionListener(this);  
panel.add(zapisz_tytul);  
zapisz_ksiazke.addActionListener(this);  
panel.add(zapisz_ksiazke);  
wyswietl_tytuly.addActionListener(this);  
panel.add(wyswietl_tytuly);  
wyswietl_ksiazki.addActionListener(this);  
panel.add(wyswietl_ksiazki);  
panel.add(tytuly);  
panel.add(elista_tytulow);  
panel.add(ksiazki);  
panel.add(elista_ksiazek);  
setContentPane(panel);  
dane= uchwyt;  
}
```



```
private void zawartosc_listy(ArrayList<String> kol,  
                                JComboBox lista) {  
    String s;  
    lista.removeAllItems();  
    Iterator<String> iterator = kol.iterator();  
    while (iterator.hasNext()) {  
        s = iterator.next();  
        lista.addItem(s);  
    }  
}
```

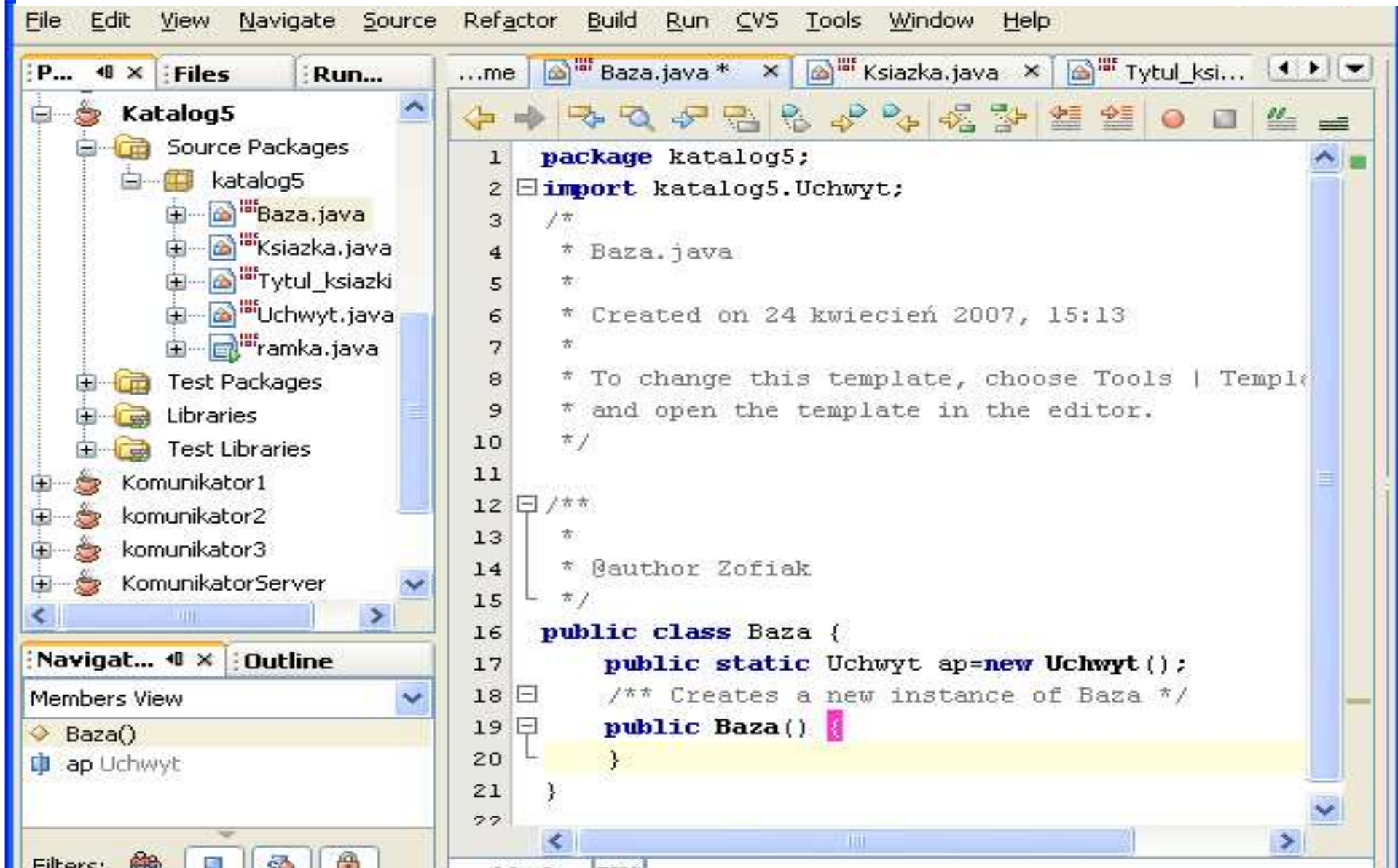
```
public void actionPerformed (ActionEvent evt)
{   String s1,s2,s3,s4,s5;
    Tytul_ksiazki t;
    Object zrodlo = evt.getSource();
if ( zrodlo==zapisz_tytul)
    {
        s1=TTytul.getText();
        s2=TNazwisko.getText();
        s3=TImie.getText();
        s4=TISBN.getText();
        s5=TWydawnictwo.getText();
if (!s1.equals("") && !s2.equals("") && !s3.equals("")
        && !s4.equals("") && !s5.equals("")) {
            dane.dodaj_tytul(s1, s2, s3, s4, s5);
        }
    }
}
```

```
else if (zrodlo == zapisz_ksiazke)
{ s1=TISBN.getText();
  s2=TNumer.getText();
  if (!s1.equals("")&&!s2.equals("")) {
    dane.dodaj_ksiazke(s1, Integer.parseInt(s2));
  }
}
else if(zrodlo == wyswietl_tytuly)
{
  zawartosc_listy(dane.tytuly(), tytuly);
}
```

```
else if (zrodlo == wyswietl_książki)
    {
        zawartosc_listy(dane.książki(), książki);
    }
    repaint();
} // koniec metody actionPerformed
} // koniec klasy GUI
```

II. Przykład drugi programu z warstwą klienta (interfejs graficzny użytkownika) i warstwą biznesową dostępną za pomocą metod klasy **Uchwyt**, należącej do warstwy biznesowej.

Klasa warstwy biznesowej – Baza



The screenshot displays an IDE window with the following components:

- Project Explorer (Left):** Shows a project named 'Katalog5' with a 'Source Packages' folder containing 'katalog5'. Inside 'katalog5', there are files: 'Baza.java', 'Ksiazka.java', 'Tytul_ksiazki', 'Uchwyt.java', and 'ramka.java'. There are also 'Test Packages', 'Libraries', and 'Test Libraries' folders, and several other projects like 'Komunikator1', 'komunikator2', 'komunikator3', and 'KomunikatorServer'.
- Code Editor (Right):** Shows the source code for 'Baza.java'. The code is as follows:

```
1 package katalog5;
2 import katalog5.Uchwyt;
3 /*
4  * Baza.java
5  *
6  * Created on 24 kwiecień 2007, 15:13
7  *
8  * To change this template, choose Tools | Template
9  * and open the template in the editor.
10 */
11
12 /**
13  *
14  * @author Zofiak
15  */
16 public class Baza {
17     public static Uchwyt ap=new Uchwyt();
18     /** Creates a new instance of Baza */
19     public Baza()
20     {
21     }
22 }
```
- Outline (Bottom Left):** Shows the 'Members View' for the 'Baza' class, listing 'Baza()' and 'ap Uchwyt'.

Klasa **ramka** reprezentująca warstwę klienta (GUI)

The screenshot shows the NetBeans IDE 5.5.1 interface. The main editor window displays the source code for the `ramka.java` file. The code is as follows:

```
package katalog5;
import java.util.Iterator;
import katalog5.Tytul_ksiazki;

class ramka extends javax.swing.JFrame {
    Uchwyty dane=Baza.ap;
    /** Creates new form ramka */
    public ramka() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    Generated Code
}
```

A context menu is open over the `ramka` class, with the `JFrame Form...` option selected. The menu also includes options like `File/Folder...`, `Java Class...`, `Java Package...`, `Java Interface...`, `JPanel Form...`, `Entity Class...`, `Entity Classes from Database...`, and `Web Service Client...`. The `Tools` menu is also visible at the bottom of the context menu.

The `Navigator` window on the left shows the project structure, including the `Katalog5` package and its sub-packages. The `Properties` window on the right shows the `Sort Mode` set to `By Name (...)`. The `Outline` window at the bottom left shows `<No View Available>`. The status bar at the bottom indicates `1:1` and `INS`.

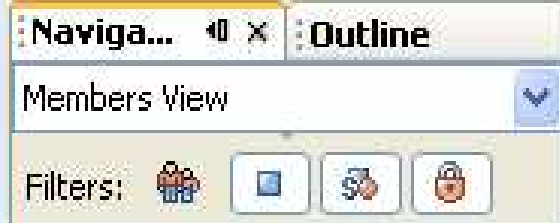
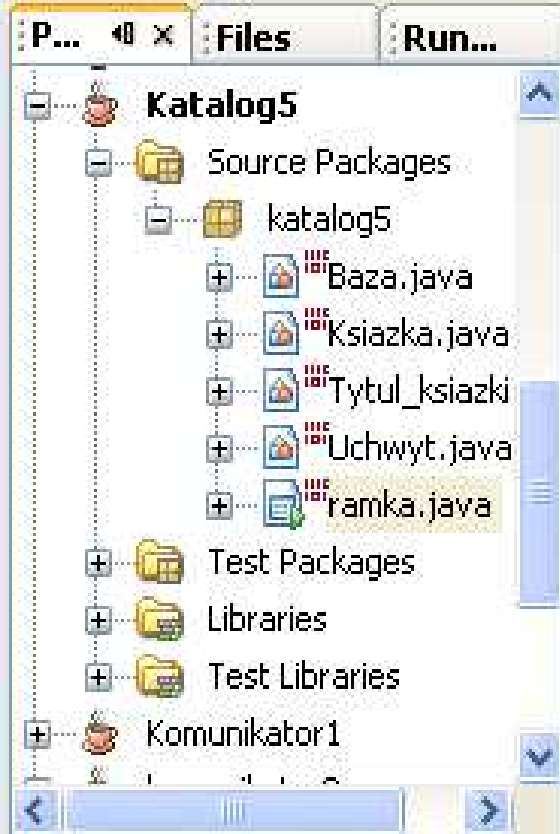
Wykonanie interfejsu graficznego metodą wizualną

The screenshot displays the NetBeans IDE 5.5 interface. The main workspace shows a visual design of a Java Swing window titled "Form ramka". The design includes several text input fields for book details: "Tytuł książki", "Nazwisko autora książki", "Imię autora książki", "ISBN tytułu", "Wydawnictwo", and "Numer książki". Below these fields are four buttons: "Zapisz tytuł", "Zapisz książkę", "Wyświetl tytuły", and "Wyświetl książki". There are also two dropdown menus labeled "Tytuły książek" and "Książki", each with "Item 1" selected. The left sidebar shows the project structure for "Katalog5" and the "Outline" view listing components like JLabel, JTextField, and JButton. The right sidebar contains the "Palette" of Swing components and the "Projects - Properties" and "Dynamic Help" panels. The status bar at the bottom indicates "Save All finished."

Obsługa zdarzeń – wywołanie metod klasy **Uchwyt** podczas obsługi zdarzeń

The screenshot displays an IDE window with the following components:

- File Explorer:** Shows a project structure with packages like `Katalog5` and source files including `Uchwyt.java`.
- Design View:** Shows a GUI form with text fields for book details (Title, Author Name, Author Name, ISBN, Publisher, Book Number) and buttons for `Zapisz tytuł`, `Zapisz książkę`, `Wyświetl tytuły`, and `Wyświetl książki`.
- Context Menu:** Opened over the `Zapisz tytuł` button, showing the **Events** tab. The **Action** event is selected, with the method `actionPerformed [JButton1ActionPerformed]` assigned to it.
- Palette:** Lists various Swing components such as `JLabel`, `JButton`, `JTextField`, and `JFrame`.
- Properties Window:** Shows the properties for the selected `JButton1` component.
- Outline:** Lists the components in the design view, including `JTextField2`, `JLabel3`, and `JButton1`.



```
Source Design
13 /**
14  *
15  * @author Zofiak
16  */
17 public class ramka extends javax.swing.JFrame {
18     Uchwyt dane=Baza.ap;
19     /** Creates new form ramka */
20     public ramka() {
21         initComponents();
22     }
23
24     /** This method is called from within the con:
25     * initialize the form.
26     * WARNING: Do NOT modify this code. The cont:
27     * always regenerated by the Form Editor.
```

244:6 INS

Metoda klasy `ramka`: pomocnicza metoda obsługi zdarzenia dodawania tytułu (GUI dla przypadku użycia „`dodaj_tytul`”)

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:hfjhgkdgkh
    String s1, s2, s3, s4, s5;
    s1 = jTextField1.getText();
    s2 = jTextField2.getText();
    s3 = jTextField3.getText();
    s4 = jTextField4.getText();
    s5 = jTextField5.getText();
if (!s1.equals("") && !s2.equals("") && !s3.equals("")
    && !s4.equals("") && !s5.equals("")) {
        dane.dodaj_tytul(s1, s2, s3, s4, s5);
    }
}
```

Metoda klasy `ramka`: pomocnicza metoda obsługi zdarzenia dodawania książki (GUI dla przypadku użycia „dodaj_książke”

```
private void jButton2ActionPerformed(  
                                java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    String s1, s2;  
    s1 = jTextField4.getText();  
    s2 = jTextField6.getText();  
    if (!s1.equals("") && !s2.equals("")) {  
        dane.dodaj_książke(s1, Integer.parseInt(s2));  
    }  
}
```

Pomocnicza metoda w klasie **ramka** do wypełniania komponentu typu **JComboBox** zawartością kolekcji typu **ArrayList<String>**

```
private void zawartosc_listy(ArrayList<String> kol,  
                               JComboBox lista) {  
  
    String s;  
    lista.removeAllItems();  
    Iterator<String> iterator = kol.iterator();  
    while (iterator.hasNext()) {  
        s = iterator.next();  
        lista.addItem(s);  
    }  
}
```

Metoda klasy **ramka**: pomocnicza metoda obsługi zdarzenia prezentacji tytułów książek

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    zawartosc_listy(dane.tytuly(), jComboBox1);
}
```

Metoda klasy **ramka**: pomocnicza metoda obsługi zdarzenia prezentacji książek

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    zawartosc_listy(dane.ksiazki(), jComboBox2);
}
```

```
private void initComponents() {  
  
    jLabel1 = new javax.swing.JLabel();  
    jTextField1 = new javax.swing.JTextField();  
    jLabel2 = new javax.swing.JLabel();  
    jTextField2 = new javax.swing.JTextField();  
    jLabel3 = new javax.swing.JLabel();  
    jTextField3 = new javax.swing.JTextField();  
    jLabel4 = new javax.swing.JLabel();  
    jTextField4 = new javax.swing.JTextField();  
    jLabel5 = new javax.swing.JLabel();  
    jTextField5 = new javax.swing.JTextField();  
    jLabel6 = new javax.swing.JLabel();  
    jTextField6 = new javax.swing.JTextField();  
    jButton1 = new javax.swing.JButton();  
    jButton2 = new javax.swing.JButton();  
    jButton3 = new javax.swing.JButton();  
    jButton4 = new javax.swing.JButton();  
    jComboBox1 = new javax.swing.JComboBox();  
    jComboBox2 = new javax.swing.JComboBox();  
    jLabel7 = new javax.swing.JLabel();  
    jLabel8 = new javax.swing.JLabel();  
}
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jLabel1.setText("Tytu\u0142 ksi\u0105\u017cki");

    jTextField1.setMinimumSize(new java.awt.Dimension(110, 19));

    jLabel2.setText("Nazwisko autora ksi\u0105\u017cki");

    jLabel3.setText("Imi\u0119 autora ksi\u0105\u017cki");

    jLabel4.setText("ISBN tytu\u0142u");

    jLabel5.setText("Wydawnictwo");

    jLabel6.setText("Numer ksi\u0105\u017cki");

    jButton1.setText("Zapisz tytu\u0142");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
```

```
jButton2.setText("Zapisz ksi\u0105\u017ck\u0119");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
```

```
jButton3.setText("Wy\u015bwietl tytu\u0142");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
```

```
jButton4.setText("Wy\u015bwietl ksi\u0105\u017cki");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
```



```
jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[]  
{ "Item 1", "Item 2", "Item 3", "Item 4" }));  
jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new  
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
jLabel7.setText("Tytu\u0142y ksi\u0105\u017cek");  
jLabel8.setText("Ksi\u0105\u017cki");  
org.jdesktop.layout.GroupLayout layout = new  
    org.jdesktop.layout.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)  
        .add(layout.createSequentialGroup()  
            .addContainerGap()  
            .add(layout.createParallelGroup(  
                org.jdesktop.layout.GroupLayout.LEADING)  
                .add(layout.createSequentialGroup()  
                    .add(layout.createParallelGroup(  
                        org.jdesktop.layout.GroupLayout.LEADING)
```

```
.add(jLabel2)
.add(jLabel1)
.add(jLabel4)
.add(jLabel3)
.add(jLabel5)
.add(jLabel6))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(layout.createParallelGroup(
    org.jdesktop.layout.GroupLayout.LEADING, false)
.add(jTextField6)
.add(jTextField5)
.add(jTextField3)
.add(jTextField2)
.add(jTextField1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    182, Short.MAX_VALUE)
.add(jTextField4)))
.add(layout.createSequentialGroup()
.add(jButton1)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jButton2).
```

```
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jButton3)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jButton4))
.add(layout.createSequentialGroup()
.add(layout.createParallelGroup(
    org.jdesktop.layout.GroupLayout.LEADING)
.add(jLabel7)
.add(jLabel8))
.add(48, 48, 48)
.add(layout.createParallelGroup(
    org.jdesktop.layout.GroupLayout.LEADING)
.add(jComboBox2,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(jComboBox1,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))))
.addContainerGap())
);
```

```
layout.setVerticalGroup(  
    layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)  
        .add(layout.createSequentialGroup()  
            .addContainerGap()  
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)  
                .add(jLabel1)  
                .add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,  
                    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,  
                    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))  
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)  
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)  
                .add(jLabel2)  
                .add(jTextField2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,  
                    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,  
                    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))  
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)  
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)  
                .add(jTextField3, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,  
                    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,  
                    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
```

```
.add(jLabel3))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
.add(jTextField4, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(jLabel4))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
.add(jLabel5)
.add(jTextField5, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
.add(jLabel6)
.add(jTextField6, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
    org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
    org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
.add(16, 16, 16)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
```

```
.add(jButton1)
.add(jButton2)
.add(jButton3)
.add(jButton4))
.add(26, 26, 26)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
.add(jLabel7)
.add(jComboBox1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
      org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
      org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(
      org.jdesktop.layout.LayoutStyle.RELATED, 54, Short.MAX_VALUE)
.add(layout.createParallelGroup(
      org.jdesktop.layout.GroupLayout.TRAILING)
.add(jComboBox2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
      org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
      org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(jLabel8))
.add(90, 90, 90))
);
pack();
} // </editor-fold>
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
} // End of variables declaration
```

Przykład wywołania programu - interfejsu użytkownika (warstwy klienta)
reprezentowanego przez klasę **ramka**

