

## 1. Inżynieria odwrotna w środowisku NetBeans – na przykładzie p. 3 instrukcji

- Należy uruchomić środowisko NetBeans 6.7.1, utworzyć rozpakowany wcześniej projekt Katalog4. Wykonać za pomocą inżynierii odwrotnej projekt typu UML.
- Dokonać analizy zawartości podkatalogu Model. Wybrać w tym podkatalogu pakiet katalog4, następnie uruchomić wyskakujące menu prawym klawiszem myszy, wybrać opcję Create Diagram From Selected Elements i następnie wybrać Class Diagram i wykonać diagram klas o nazwie Katalog4 (zatwierdzić komunikat OK, dotyczący tworzenia diagramu klas).
- W klasie Uchwyt w podkatalogu Model w zakładce Operations zaznaczyć prawym klawiszem myszy metodę *dodaj\_tytul* i z wyskakującego menu wybrać opcję Reverse Engineer Operation i wykonać diagram sekwencji.
- Podobnie wykonać diagram sekwencji dla metody *dodaj\_książke* w klasie *Uchwyt*, następnie dla operacji *dodaj\_książke* w klasie *Tytul\_książki* oraz *addKsiążka* w klasie *Tytul\_książki*. Należy przeanalizować związek między elementami wszystkich diagramów a kodem źródłowym metod i klas -
- zwrócić uwagę na związek pomiędzy wywoływanymi metodami jednych obiektów przez inne obiekty i powiązaniem klas tych obiektów na diagramie klas.

## 2. Dokonaj analizy diagramu klas z rys. 2, uzyskany za pomocą inżynierii odwrotnej z programu Wektor2, którego kod źródłowy podano na rys.1, podanego jako przykład w ramach zestawu 1 laboratorium 2.

```
import javax.swing.*;
import java.util.*;

public class Dane1 implements Comparable <Dane1>
{ String nazwisko;
  float srednia;
  String uwagi;
  public void Nadaj_nazwisko(String lan)    { nazwisko=lan;}
  public String Podaj_nazwisko()           { return nazwisko;}
  public void Nadaj_uwagi(String lan)      { uwagi=lan;}
  public String Podaj_uwagi()              { return uwagi;}
  public void Nadaj_srednia(float srednia_) { srednia=srednia_;}
  public float Podaj_srednia()             { return srednia;}
  public static Dane1 Wstaw(String nazwisko_, String uwagi_, String srednia_)
  { Dane1 d=new Dane1();
    d.Nadaj_nazwisko(nazwisko_);
    d.Nadaj_srednia(Float.parseFloat(srednia_));
    d.Nadaj_uwagi(uwagi_);
    return d; }
  public String toString()
  { String napis="";
    napis+=" Nazwisko: "+nazwisko;
    napis+=" Srednia: "+srednia;
    napis+=" Uwagi: "+uwagi+"\n";
    return napis; }

  public boolean equals(Object o)           //metoda umozliwiająca wstawianie do pojemników typu Hash
  { Dane1 d=(Dane1)o;
    return nazwisko.equals(d.nazwisko) && srednia==d.srednia && uwagi.equals(d.uwagi); }

  public int hashCode()                    //metoda umozliwiająca wstawianie do pojemników typu Hash
  { return nazwisko.hashCode()+((int)srednia+uwagi.hashCode()); }

  public int compareTo(Dane1 o)            //metoda umozliwiająca sortowanie, wyszukiwanie w pojemnikach List
  { return nazwisko.compareTo(o.nazwisko); } //oraz przetwarzanie w pojemnikach typu TreeSet i TreeMap
}

//-----
import java.awt.Dimension;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.DefaultTableModel;
//-----
class Panel extends JPanel implements ListSelectionListener
{ JTable tabela;
  DefaultTableModel model;
  Vector <Dane1> dane;
  int wiersz;
```

```

public Panel(Vector <Dane1> v)
{ super();
  dane=v;
  final String[] columnNames = {"Nazwisko","Uwagi","Srednia"};
  model=new DefaultTableModel(columnNames,0);
  tabela= new JTable(model);
  model.addRow(new Vector<String>(3));
  tabela.setPreferredScrollableViewportSize(new Dimension(250,150));
  tabela.setSelectionMode(ListSelectionModel.SINGLE_SELECTION );
  ListSelectionModel selekcja = tabela.getSelectionModel();
  selekcja.addListSelectionListener(this);
  JScrollPane suwak = new JScrollPane(tabela);
  add(suwak);
}
public void valueChanged(ListSelectionEvent e)
{ if (e.getValueAdjusting()) return;
  ListSelectionModel lsm = (ListSelectionModel)e.getSource();
  if (!lsm.isSelectionEmpty())
  { wiersz = lsm.getMinSelectionIndex();
    System.out.println("Wiersz " + wiersz+ " jest wybrany."); }
}
public void wyswietl()
{ Iterator iterator = dane.iterator();
  model.setRowCount(0);
  int i=0;
  while(iterator.hasNext())
  { Dane1 t=(Dane1)iterator.next();
    Vector <String>vv=new Vector<String>(3);
    model.addRow(vv);
    model.setValueAt(t.Podaj_nazwisko(),i,0);
    model.setValueAt(t.Podaj_uwagi(),i,1);
    model.setValueAt(""+t.Podaj_srednia(),i,2);
    i++; }
  Vector <String>vv=new Vector<String>(3);
  model.addRow(vv);
}
void obslugaactionPerformed()
{ if(tabela.isRowSelected(wiersz))
  {String nazwisko=(String)model.getValueAt(wiersz, 0); //pobranie z wybranych
  String uwagi=(String)model.getValueAt(wiersz, 1); // komorek lancuchy
  String srednia=(String)model.getValueAt(wiersz, 2);
  if(nazwisko!=null&&uwagi!=null&&srednia!=null&& !nazwisko.equals("")&& !uwagi.equals("") &&!srednia.equals(""))
  { try
    { //Float.parseFloat(srednia);
      Dane1 d=Dane1.Wstaw(nazwisko,uwagi, srednia);
      dane.add(d);
      Collections.sort(dane);
      wyswietl();
    } catch(NumberFormatException e)
    { JOptionPane.showMessageDialog(this,
      "Wprowadz poprawna srednia"); }
  }
}
}
}
}
}
//-----
class Okno extends JFrame implements ActionListener
{ Panel panel; //skladniki interfejsu graficznego uzytkownika
  JButton osobaOK ;
  JButton repaintOK ;

public Okno(Vector <Dane1>dane)
{ super("Dane osob");
  setSize(350,250);
  osobaOK=new JButton("Osoba OK");
  osobaOK.addActionListener(this);
  repaintOK=new JButton("Repaint OK");
  repaintOK.addActionListener(this);
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  panel = new Panel(dane);
  panel.setOpaque(true);
  panel.add(osobaOK);
  panel.add(repaintOK);
  setContentPane(panel);
  setVisible(true); }
}

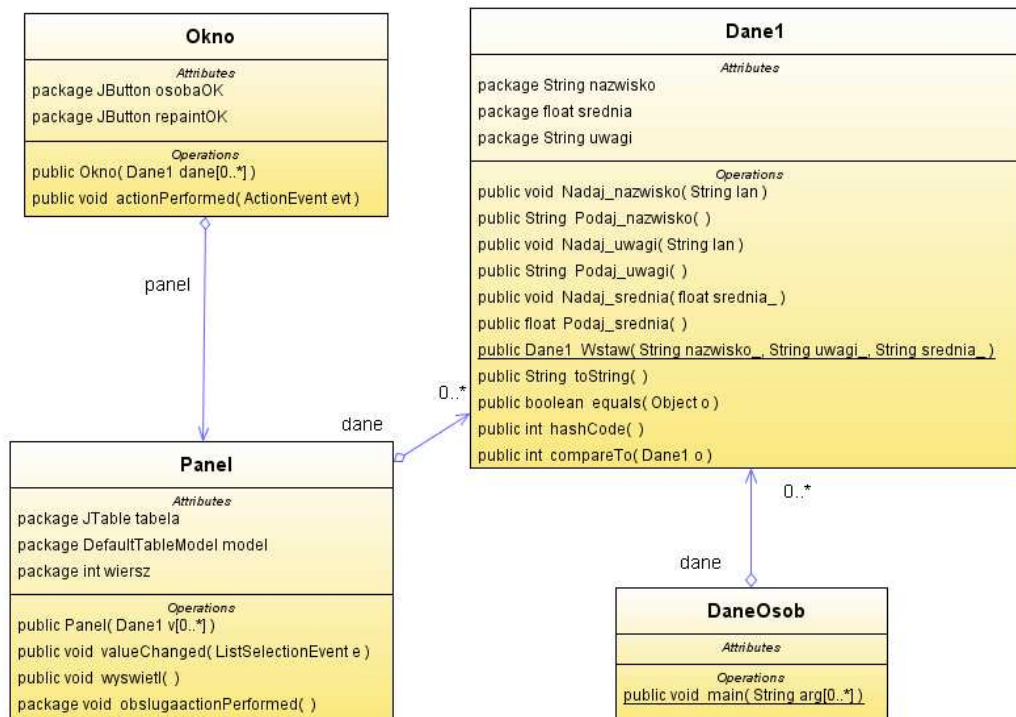
```

```

public void actionPerformed(ActionEvent evt)
{ Object zrodlo=evt.getSource();
  if (zrodlo==osobaOK)   panel.obsługaactionPerformed();
  else
  if (zrodlo==repaintOK)
    panel.wyświetl();
  repaint();           // wywołanie funkcji odświeżającej zawartosc tabeli i pokazanie jej na ekranie
}
}
//-----
public class DaneOsob
{ Vector<Dane1> dane=new Vector<Dane1>(5,5);
  static public void main(String arg[])
  { DaneOsob baza = new DaneOsob();
    Okno okno = new Okno(baza.dane); }
}

```

Rys. 1. Kod źródłowy programu Wektor2



Rys.2. Diagram klas wygenerowany za pomocą inżynierii odwrotnej z kodu źródłowego z rys.1 – projekt Wektor2

3. Dokonaj analizy diagramu klas z rys. 4, uzyskany za pomocą inżynierii odwrotnej z programu Baza0, którego kod źródłowy podano na rys.3, podanego jako przykład w ramach zestawu 2 z laboratorium 2.

```
package baza0;
import javax.swing.*;

class Osoba
{
    String nazwisko;
    float srednia;
    String uwagi;

    public void Nadaj_nazwisko(String lan)    { nazwisko=lan;}
    public String Podaj_nazwisko()           { return nazwisko;}
    public void Nadaj_uwagi(String lan)      { uwagi=lan;}
    public String Podaj_uwagi()              { return uwagi;}
    public void Nadaj_srednia(float srednia_) { srednia=srednia_;}
    public float Podaj_srednia()             {return srednia;}
    public String toString()
    {String napis="";
     napis+="Nazwisko: "+nazwisko;
     napis+=" srednia: "+srednia;
     napis+=" uwagi: "+uwagi;
     return napis;
    }
    public boolean Szukaj(String s)          { return nazwisko.equals(s); }
}
//-----

class Tablica
{
    Osoba Dane[]=null;
    int ile=0;

    public boolean Pusta()                   { return ile==0; }
    public boolean Pelna()                   { return ile==Dane.length; }
    public int Podaj_ile()                   { return ile; }

    public void Wykonaj_tablice(int N)
    { ile=0;
      Dane=new Osoba[N]; //utworzenie tablicy
    }
    public void Wstaw (String nazwisko_, float srednia, String uwagi_)
    {
        Osoba dana=new Osoba();
        dana.Nadaj_nazwisko(nazwisko_);
        dana.Nadaj_srednia(srednia);
        dana.Nadaj_uwagi(uwagi_);
        Dane[ile]=dana;
        ile++;
    }
    public Osoba Wyszukaj(String s)
    {
        for (int i=0; i<ile;i++)
            if (Dane[i].Szukaj(s))
                return Dane[i];
        return null;
    }
    public String toString()
    { String s="";
      for (int i=0; i<ile; i++)
          s+=Dane[i].toString()+"\n";
      return s; }
}
//-----

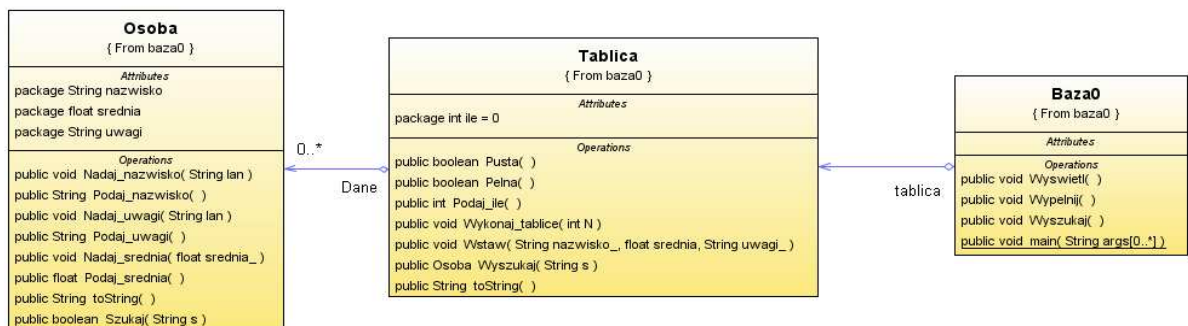
public class Baza0
{
    protected Tablica tablica=new Tablica();
    public void Wyszukaj()
    { if (tablica.Pusta())//jesli tablica istnieje, to mozna wyswietlac
      JOptionPane.showMessageDialog(null,"Brak danych");
      else
        JOptionPane.showMessageDialog(null,tablica.toString());
    }
}
```

```

public void Wypelnij()
{
    String S;
    S=JOptionPane.showInputDialog(null,"Podaj rozmiar tablicy");
    int N=Integer.parseInt(S);
    tablica.Wykonaj_tablice(N); //utworzenie tablicy
    String S1,S2,S3;
    while(!tablica.Pelna()) //jeśli tablica jest zapełniona, wyrażenie dla while jest równe false
    {
        S1 = JOptionPane.showInputDialog(null, "Podaj nazwisko");
        S2 = JOptionPane.showInputDialog(null, "Podaj srednia");
        S3 =JOptionPane.showInputDialog(null, "Podaj uwagi");
        tablica.Wstaw(S1, Float.parseFloat(S2), S3); //dopisanie danych na koncu
    }
}
public void Wyszukaj()
{
    if (tablica.Pusta())//jesli tablica istnieje, to mozna wyszukiwac
        JOptionPane.showMessageDialog(null,"Brak danych");
    else
    {
        String S = JOptionPane.showInputDialog(null,"Podaj nazwisko");
        Object osoba=tablica.Wyszukaj(S);
        if (osoba!=null)
            JOptionPane.showMessageDialog(null,osoba.toString()); }
}
static public void main(String args[])
{
    Baza0 baza1=new Baza0(); //referencja do tablicy, ktora jest tworzona w opcji 1
    char ch;
    String s;
    do
    {
        s=JOptionPane.showInputDialog(null, "Progarm nie jest zabezpieczony przed\n" +
            "złym formatem danych\n i naciskaniem Cancel w okienkach dialogowych\n"+ "Podaj wybor"
            +"\n1 - Wypelnij tablice,"
            +"\n2 - Wyszukaj dane osob"
            +"\n3 - Wyszukaj osobe i wyswietl jej dane"
            +"\nk - Koniec programu");
        ch = s.charAt(0); //pobranie opcji
        switch(ch)
        {
            case '1' : baza1.Wypelnij(); break;
            case '2' : baza1.Wyswietl(); break;
            case '3' : baza1.Wyszukaj(); break;
            case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu"); break;
            default : JOptionPane.showMessageDialog(null,"Zla opcja"); }
    } while (ch != 'k');
    System.exit(0);
}
}
}

```

Rys.3. Kod źródłowy programu Baza0.

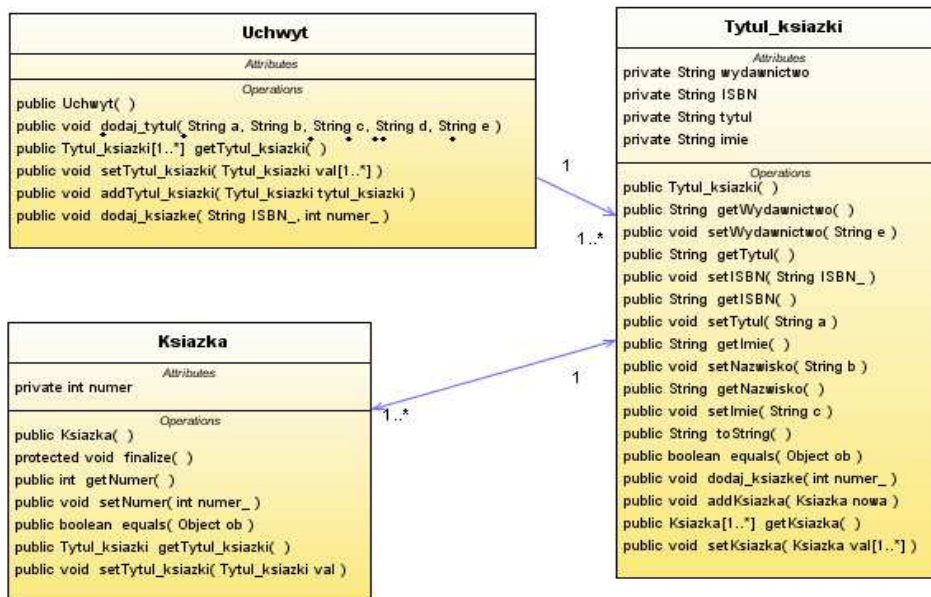


Rys.4. Diagram klas wygenerowany za pomocą inżynierii odwrotnej z kodu źródłowego z rys.3

#### 4. Wykonaj zadanie:

Wykonaj program obsługujący informacje o tytułach i książkach w bibliotece (wstawianie tytułów, wstawianie książek, usuwanie tytułów, usuwanie książek, wyszukiwanie wg tytułu, wyświetlanie tytułów, wyświetlanie książek dla wybranego tytułu, zapis w formacie serializowanych obiektów do jednego pliku danych o tytułach i do drugiego pliku dane o książkach, odczyt danych z plików i wydruk na ekran) oraz odtworzenie danych z plików i wstawienie ich do kolekcji w programie. Dodatkowo należy wykonać zapis w formacie tekstowym do jednego pliku danych o tytułach i do drugiego pliku dane o książkach, odczyt danych z plików i wydruk na ekran) \* oraz odtworzenie danych z plików i wstawienie ich do kolekcji w programie ( ta funkcja na ocenę 5.5).

Diagram klas z rys.5 (projekt Katalog4) należy rozszerzyć o podane funkcje wg zasady podanej na diagramie. Klasa Uchwyt jest fasadą modelu danych (projekt Katalog4). Można dołożyć klasę typu Baza0 umożliwiającą pobieranie danych z klawiatury i wyświetlanie ich na ekranie (klasa ta powinna posiadać atrybut tablica typu Uchwyt zamiast atrybutu typu Tablica) – załącznik do zestawu 2 z laboratorium 2. Przykład obsługi plików tekstowych podano w załączonym projekcie Baza1 do laboratorium 3, natomiast obsługa plików z serializowanymi obiektami w załączonym projekcie Baza4 do laboratorium 3. Wykonaj diagram klas poszerzony o podane metody w projekcie UML i wygeneruj szkielet kodu do uzupełnienia. Następnie wykonaj diagram klas wykonanego programu za pomocą inżynierii odwrotnej. Sposób zainstalowania dodatku UML podano na stronie www, gdzie pobrano instrukcję.



Rys.5. Diagram klas wykonany za pomocą inżynierii wprost – projekt Katalog4.

#### Kod źródłowy dotyczący diagramu klas z rys.5.:

```
package katalog4;
import katalog4.Tytul_książki;

public class Książka {
    private int numer;
    private Tytul_książki mTytul_książki;

    public Książka()
    protected void finalize()

    public int getNumer()
    public void setNumer(int numer_)

    public Tytul_książki getTytul_książki()
    public void setTytul_książki(Tytul_książki val)

    public boolean equals(Object ob)

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

```

package katalog4;
import java.util.ArrayList;
import katalog4.Ksiazka;

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 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 ArrayList<Ksiazka> getKsiazka() { return mKsiazka; }
    public void setKsiazka(ArrayList<Ksiazka> val) { mKsiazka = val; }

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

    public void addKsiazka(Ksiazka nowa) {
        if (!mKsiazka.contains(nowa))
            { mKsiazka.add(nowa);
              nowa.setTytul_ksiazki(this);}
    }
}
//-----
package katalog4;
import java.util.ArrayList;

public class Uchwyt {

    private ArrayList<Tytul_ksiazki> mTytul_ksiazki= new ArrayList<Tytul_ksiazki>();

    public Uchwyt() { }

    public void dodaj_tytul(String a, String b, String c, String d, String e) {
        Tytul_ksiazki tytul_ksiazki=new Tytul_ksiazki();
        tytul_ksiazki.setTytul(a);
        tytul_ksiazki.setNazwisko(b);
        tytul_ksiazki.setImie(c);
        tytul_ksiazki.setISBN(d);
        tytul_ksiazki.setWydawnictwo(e);
        addTytul_ksiazki(tytul_ksiazki);
    }

    public ArrayList<Tytul_ksiazki> getTytul_ksiazki() { return mTytul_ksiazki; }

    public void setTytul_ksiazki(ArrayList<Tytul_ksiazki> val) { mTytul_ksiazki = 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_) {
    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[])    // your code her
{
    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_książki().toString();
    System.out.println(lan);

    ap.dodaj_książke("1",1);
    ap.dodaj_książke("1",2);
    ap.dodaj_książke("1",2);
    ap.dodaj_książke("2",1);
}
}

```

**Rys. 6. Kod źródłowy dla diagramu z rys. 5 – projekt Katalog4.**

```

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]

```

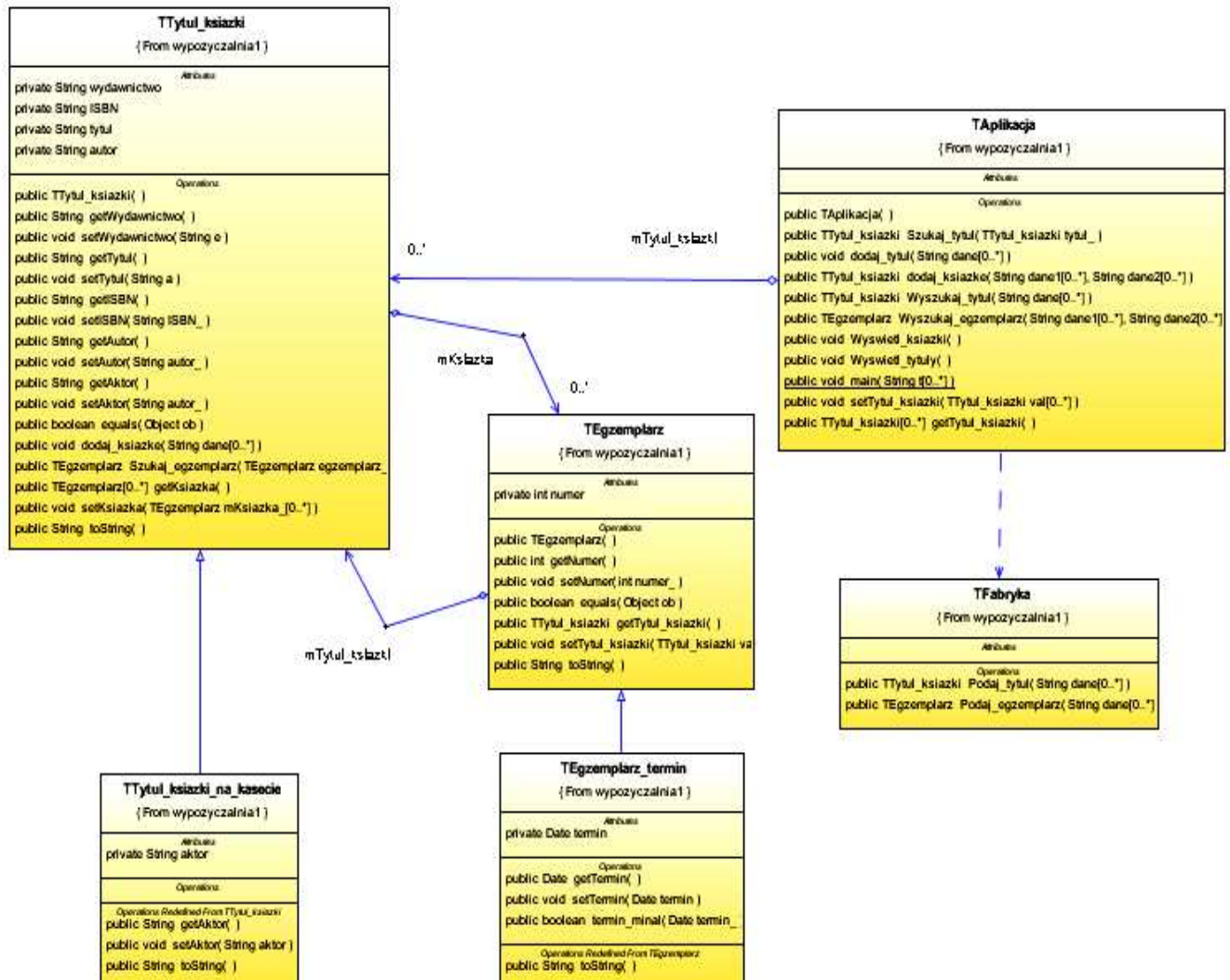
**Rys.7. Wynik działania programu przedstawionego na rys.6.**



## 5. Wykonaj zadanie:

Wykonaj program obsługujący informację o tytułach i książkach w bibliotece (wstawianie tytułów, wstawianie książek, usuwanie tytułów, usuwanie książek, wyszukiwanie wg tytułu, wyświetlanie tytułów, wyświetlanie książek dla wybranego tytułu, zapis w formacie serializowanych obiektów do jednego pliku danych o tytułach i do drugiego pliku dane o książkach, odczyt danych z plików i wydruk na ekran) oraz odtworzenie danych z plików i wstawienie ich do kolekcji w programie. Dodatkowo należy wykonać zapis w formacie tekstowym do jednego pliku danych o tytułach i do drugiego pliku dane o książkach, odczyt danych z plików i wydruk na ekran) \* oraz odtworzenie danych z plików i wstawienie ich do kolekcji w programie ( ta funkcja na ocenę 5.5).

Diagram klas z rys.8 (projekt Wypożyczalnia1app\_1) należy rozszerzyć o podane funkcje wg zasady podanej na diagramie. Klasa TAplikacja jest fasadą modelu danych (projekt Wypożyczalnia1app\_1). Można dołożyć klasę typu Baza0 umożliwiającą pobieranie danych z klawiatury i wyświetlanie ich na ekranie (klasa ta powinna posiadać atrybut tablica typu TAplikacja zamiast atrybutu typu Tablica) – załącznik do zestawu 2 z laboratorium 2. Przykład obsługi plików tekstowych podano w załączonym projekcie Baza1 do laboratorium 3, natomiast obsługa plików z serializowanymi obiektami w załączonym projekcie Baza4 do laboratorium 3. Wykonaj diagram klas poszerzony o podane metody w projekcie UML i wygeneruj szkielet kodu do uzupełnienia. Następnie wykonaj diagram klas wykonanego programu za pomocą inżynierii odwrotnej. Sposób zainstalowania dodatku UML podano na stronie www, gdzie pobrano instrukcję.



Rys. 8. Diagram klas rozszerzający diagram klas z rys. 5 o dziedziczenie klas - projekt Wypożyczalnia1app\_1.

```
package wypożyczalnia1app;
```

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

```
public class TAplikacja {
    private ArrayList<TTytuł_książki> Tytuł_książki = new ArrayList<TTytuł_książki>();
    public TAplikacja() {
    }
}
```

```

public TTytul_książki Szukaj_tytul(TTytul_książki tytul_) {
    int idx;
    if ((idx = Tytul_książki.indexOf(tytul_)) != -1) {
        tytul_ = Tytul_książki.get(idx);
        return tytul_;
    }
    return null;
}

public TTytul_książki dodaj_tytul(String dane[]) {
    TFabryka fabryka = new TFabryka();
    TTytul_książki tytul_książki = fabryka.Podaj_tytul(dane);
    if (Szukaj_tytul(tytul_książki) == null) {
        Tytul_książki.add(tytul_książki);
        return tytul_książki;
    }
    return null;
}

public ArrayList<TTytul_książki> getTytul_książki()           { return Tytul_książki; }
public void setTytul_książki(ArrayList<TTytul_książki> val) { Tytul_książki = val; }

public TTytul_książki dodaj_książke(String dane1[], String dane2[]) {
    TTytul_książki pom, pom1 = null;
    TFabryka fabryka = new TFabryka();
    pom = fabryka.Podaj_tytul(dane1);
    if ((pom1 = Szukaj_tytul(pom)) != null) {
        pom1.dodaj_książke(dane2);    }
    return pom1;
}

public TTytul_książki Wyszukaj_tytul(String dane[]) {
    TFabryka fabryka = new TFabryka();
    TTytul_książki tytul_książki = fabryka.Podaj_tytul(dane);
    return Szukaj_tytul(tytul_książki);
}

public TEgzemplarz Wyszukaj_egzemplarz(String dane1[], String dane2[]) {
    TFabryka fabryka = new TFabryka();
    TTytul_książki tytul_książki = fabryka.Podaj_tytul(dane1);
    TTytul_książki tytul = Szukaj_tytul(tytul_książki);
    if (tytul != null) {
        TEgzemplarz egzemplarz = fabryka.Podaj_egzemplarz(dane2);
        return tytul.Szukaj_egzemplarz(egzemplarz);    }
    return null;
}

public void Wyswietl_książki() {
    System.out.println("\nKsiążki");
    TTytul_książki tytul;
    TEgzemplarz egz;
    Iterator<TTytul_książki> it1 = Tytul_książki.iterator();
    while (it1.hasNext()) {
        tytul = it1.next();
        Iterator<TEgzemplarz> it2 = tytul.getMKsiążka().iterator();
        while (it2.hasNext()) {
            egz = it2.next();
            System.out.println(egz.toString());    }
    }
}

public void Wyswietl_tytuly() {
    System.out.println("\nTytuly książek");
    TTytul_książki tytul;
    Iterator<TTytul_książki> it = Tytul_książki.iterator();
    while (it.hasNext()) {
        tytul = it.next();
        System.out.println(tytul.toString());    }
}

```

```

public static void main(String t[]) {
    TAplikacja ap = new TAplikacja();
    String t1[] = {"1", "1", "1", "1", "1"};
    String t2[] = {"1", "2", "2", "2", "2"};
    String t3[] = {"1", "3", "3", "3", "3"};
    String t4[] = {"3", "1", "1", "1", "1", "1"};
    String t5[] = {"3", "2", "2", "2", "2", "2"};
    String t6[] = {"3", "4", "4", "4", "4", "4"};
    ap.dodaj_tytul(t1);
    ap.dodaj_tytul(t2);
    ap.dodaj_tytul(t2);
    ap.dodaj_tytul(t3);
    ap.dodaj_tytul(t4);
    ap.dodaj_tytul(t5);
    ap.dodaj_tytul(t5);
    ap.dodaj_tytul(t6);
    String lan = ap.getTytul_ksiazki().toString();
    System.out.println(lan);
    String d1[] = {"0", "1"};
    String d2[] = {"0", "2"};
    String d3[] = {"0", "5"};
    String d4[] = {"2", "1", "1"};
    String d5[] = {"2", "4", "4"};
    String tr1[] = {"0", "1"};
    String tr2[] = {"0", "2"};
    String tr3[] = {"1", "3", "3"};
    String tr4[] = {"1", "2", "5"};
    TTYtul_ksiazki pom = ap.dodaj_ksiazke(d1, tr1);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d2, tr1);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d2, tr1);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d2, tr2);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d3, tr2);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d4, tr3);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d4, tr3);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d4, tr4);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    pom = ap.dodaj_ksiazke(d5, tr2);
    if (pom != null) { System.out.println(pom.getMKsiazka().toString()); }
    ap.Wyswietl_tytuly();
    ap.Wyswietl_ksiazki();
    System.out.println("Wyszukiwanie");
    System.out.println(ap.Wyszukaj_tytul(t5).toString());
    System.out.println(ap.Wyszukaj_egzemplarz(d4, tr4).toString());
}
}

```

```

package wypożyczalnia1app;
import java.util.Date;

```

```

public class TFabryka {
    public TTYtul_ksiazki Podaj_tytul(String dane[])
    { TTYtul_ksiazki tytul = null;
    switch(Integer.parseInt(dane[0]))
    { case 0: tytul= new TTYtul_ksiazki();
        tytul.setISBN(dane[1]);
        break;
        case 1: tytul= new TTYtul_ksiazki();
        tytul.setAutor(dane[1]);
        tytul.setTytul(dane[2]);
        tytul.setISBN(dane[3]);
        tytul.setWydawnictwo(dane[4]);
        break;
        case 2: tytul= new TTYtul_ksiazki_na_kascecie();
        tytul.setISBN(dane[1]);
        ((TTYtul_ksiazki_na_kascecie)tytul).setAktor(dane[2]);
        break;
    }
}

```

```

        case 3: tytul= new TTytul_ksiazki_na_kasecie();
            tytul.setAutor(dane[1]);
            tytul.setTytul(dane[2]);
            tytul.setISBN(dane[3]);
            tytul.setWydawnictwo(dane[4]);
            ((TTytul_ksiazki_na_kasecie)tytul).setAktor(dane[4]);          break;
    }
    return tytul;
}

public TEgzemplarz Podaj_egzemplarz(String dane[])
{
    TEgzemplarz egzemplarz=null;
    switch(Integer.parseInt(dane[0]))
    {
        case 0: egzemplarz = new TEgzemplarz();
            egzemplarz.setNumer(Integer.parseInt(dane[1]));
            break;
        case 1: egzemplarz = new TEgzemplarz_termin();
            egzemplarz.setNumer(Integer.parseInt(dane[1]));
            Date pom=new Date();
            pom.setTime(pom.getTime()+Integer.parseInt(dane[2])*1000*60*60*24);
            ((TEgzemplarz_termin)egzemplarz).setTermin(pom);
            break;
    }
    return egzemplarz;
}

}

package wypożyczalnia1app;
import java.util.ArrayList;

public class TTytul_ksiazki {
    private String wydawnictwo;
    private String ISBN;
    private String tytul;
    private String autor;
    private ArrayList<TEgzemplarz> mKsiazka = new java.util.ArrayList<TEgzemplarz>();
    public ArrayList<TEgzemplarz> getMKsiazka() { return mKsiazka; }
    public void setMKsiazka(ArrayList<TEgzemplarz> mKsiazka) { this.mKsiazka = mKsiazka; }
    public TTytul_ksiazki() { }
    public String getWydawnictwo() { return wydawnictwo; }
    public void setWydawnictwo(String e) { this.wydawnictwo = e; }
    public String getTytul() { return tytul; }
    public void setTytul(String a) { this.tytul = a; }
    public String getISBN() { return ISBN; }
    public void setISBN(String ISBN_) { this.ISBN = ISBN_; }
    public String getAutor() { return autor; }
    public void setAutor(String autor_) { this.autor = autor_; }
    public String getAktor() { return ""; }
    public void setAktor(String autor_) { }
}

@Override
public String toString() {
    String pom = "Tytul: " + getTytul();
    pom += " Autor: " + getAutor();
    pom += " ISBN: " + getISBN();
    pom += " Wydawnictwo: " + getWydawnictwo();
    return pom;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += Integer.parseInt(ISBN);
    return hash;
}

@Override
public boolean equals(Object ob) {
    boolean a = false;
    if (getISBN().equals(((TTytul_ksiazki) ob).getISBN())) {
        if (getAktor().equals(((TTytul_ksiazki) ob).getAktor())) {
            a = true;
        }
    }
    return a;
}
}

```

```

public void dodaj_ksiazke(String dane[]) {
    TFabryka fabryka = new TFabryka();
    TEgzemplarz nowa;
    nowa = fabryka.Podaj_egzemplarz(dane);
    if (Szukaj_egzemplarz(nowa) == null) {
        mKsiazka.add(nowa);
        nowa.setMTytul_ksiazki(this);
    }
}
public TEgzemplarz Szukaj_egzemplarz(TEgzemplarz egzemplarz_) {
    int idx;
    if ((idx = mKsiazka.indexOf(egzemplarz_)) != -1) {
        egzemplarz_ = (TEgzemplarz)mKsiazka.get(idx);
        return egzemplarz_;
    }
    return null;
}
}

```

```
package wypożyczalnia1app;
```

```

public class TTytul_ksiazki_na_kascecie extends TTytul_ksiazki {
    private String aktor;

    @Override
    public String getAktor()          { return aktor; }

    @Override
    public void setAktor(String aktor) { this.aktor = aktor; }

    @Override
    public String toString() {
        String pom = super.toString();
        pom += " Aktor: " + getAktor();
        return pom; }
}

```

```
package wypożyczalnia1app;
import java.util.Date;
```

```

public class TEgzemplarz {

    private int numer;
    private TTytul_ksiazki mTytul_ksiazki;

    public TEgzemplarz()          { }
    public TTytul_ksiazki getMTytul_ksiazki() { return mTytul_ksiazki; }

    public void setMTytul_ksiazki(TTytul_ksiazki mTytul_ksiazki) { this.mTytul_ksiazki = mTytul_ksiazki; }
    public Date getTermin()      { return null; }
    public void setTermin(Date termin) { }
    public int getNumer()        { return numer; }
    public void setNumer(int numer_) { this.numer = numer_; }

    @Override
    public boolean equals(Object ob) { return numer == ((TEgzemplarz) ob).getNumer(); }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += numer;
        return hash;
    }

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

```

```

package wypożyczalnia1app;

import java.util.Date;

public class TEgzemplarz_termin extends TEgzemplarz    {

    private Date termin;

    @Override
    public Date getTermin()                {    return termin;    }

    @Override
    public void setTermin(Date termin)      {    this.termin = termin;    }

    public boolean termin_minal(Date termin_) {    return termin.compareTo(termin_) >= 0;    }

    @Override
    public String toString() // your code here
    {
        String pom = super.toString();
        pom += " termin: " + termin.toString();
        return pom;
    }
}

```

**Rys.9. Kod źródłowy programu, którego diagram klas pokazano na rys. 7 - projekt Wypożyczalnia1app\_1.**

```

[Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1, Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2, Tytuł: 3 Autor: 3 ISBN: 3 Wydawnictwo: 3, Tytuł: 1 Autor:
1 ISBN: 1 Wydawnictwo: 1 Aktor: 1, Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2, Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4 Aktor: 4]
[Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Numer: 1]
[Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 1]
[Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 1]
[Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 1, Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 2]
[Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 3 termin: Mon Oct 19 21:47:37 CEST 2009]
[Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 3 termin: Mon Oct 19 21:47:37 CEST 2009]
[Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 3 termin: Mon Oct 19 21:47:37 CEST 2009, Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo:
1 Aktor: 1 Numer: 2 termin: Wed Oct 21 21:47:37 CEST 2009]
[Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4 Aktor: 4 Numer: 2]

```

Tytuły książek

```

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1
Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2
Tytuł: 3 Autor: 3 ISBN: 3 Wydawnictwo: 3
Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1
Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4 Aktor: 4

```

Książki

```

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Numer: 1
Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 1
Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Numer: 2
Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 3 termin: Mon Oct 19 21:47:37 CEST 2009
Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 2 termin: Wed Oct 21 21:47:37 CEST 2009
Tytuł: 4 Autor: 4 ISBN: 4 Wydawnictwo: 4 Aktor: 4 Numer: 2

```

Wyszukiwanie

```

Tytuł: 2 Autor: 2 ISBN: 2 Wydawnictwo: 2 Aktor: 2
Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1 Aktor: 1 Numer: 2 termin: Wed Oct 21 21:47:37 CEST 2009

```

**Rys. 10. Wynik działania programu - metody main z klasy TAplikacja.**