

## LABORATORIUM 7

**Cel:** Budowa klas z wykorzystaniem podstawowych pojemników na dane typu String. W zadaniach wykorzystuje się dwa typy pojemników: typu Collection rozróżniając wśród nich pojemniki typu List (ArrayList oraz LinkedList) oraz Set (HashSet, TreeSet) oraz typu Map (HashMap oraz TreeMap)

1. Wykonaj program (klasa *arraylist\_*), który opiera się na programie *Tytuły*, podanym w projekcie *Wektor*. Zamiast pojemnika *Vector* zastosować pojemnik typu *ArrayList* – należy poprawić lub usunąć wskazane przez kompilator metody, które pojemnik *ArrayList* nie posiada (projekty: *Pojemniki*, *Iteratory*). Należy uzupełnić program o sortowanie uruchamiane za pomocą dodanego przycisku „Sortuj” (typu *JButton*) i wyszukiwanie połówkowe uruchamiane za pomocą dodanego przycisku „Szukaj” (typu *JButton*) po wprowadzeniu wyszukiwanego łańcucha w polu tekstowym oznaczonym etykietą „Podaj tytuł książki” i wyświetlając w dodanym polu tekstowym typu *JTextField* numer znalezionej elementu- obie funkcje zrealizować za pomocą metod klasy usługowej *Collections* w metodzie *actionPerformed*. Przykład zastosowania klasy *Collections* podano w projekcie *Pojemniki*.

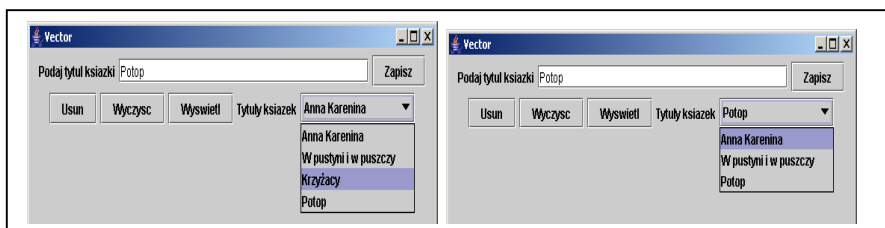
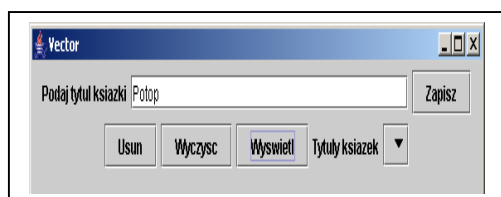
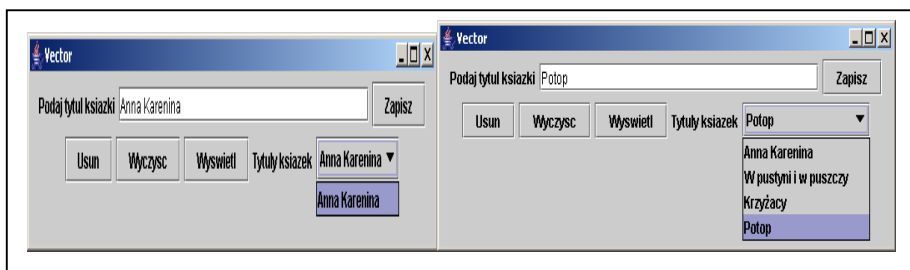
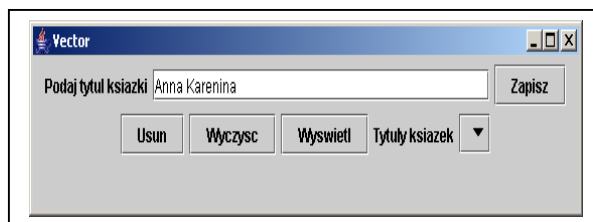
```
import java.sql.*;
import javax.swing.*;
import java.util.*;
import java.io.*;
import java.lang.*;
import java.awt.event.*;
```

```
class Okno extends JFrame implements ActionListener
```

```
{ JLabel elista= new JLabel ("Tytuły książek");
  JComboBox lista;
  JLabel ewiersz= new JLabel ("Podaj tytuł książki");
  JTextField wiersz;
  JButton zapisz=new JButton("Zapisz");
  JButton usun=new JButton("Usun");
  JButton wyczysc=new JButton("Wyczysc");
  JButton wyswietl=new JButton("Wyswietl");
  Vector <String> dane;
  Okno(Vector <String> vector)
  { super("Vector");
    lista=new JComboBox();
    wiersz=new JTextField(30);
    dane= vector;
    setSize(550,250);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JPanel panel= new JPanel();
    panel.add(ewiersz);
    zapisz.addActionListener(this);
    usun.addActionListener(this);
    wyczysc.addActionListener(this);
    wyswietl.addActionListener(this);
    panel.add(elista);
    lista.addActionListener(this);
    setContentPane(panel); }
    panel.add(wiersz);
    panel.add(zapisz);
    panel.add(usun);
    panel.add(wyczysc);
    panel.add(wyswietl);
    panel.add(lista);
```

```
public void actionPerformed (ActionEvent evt)
```

```
{ String s;
  Object zrodlo = evt.getSource();
  if( zrodlo==zapisz)
  { s=wiersz.getText();
    if (!s.equals(""))
    { dane.add( s);
      lista.addItem(s);}
  }
  else
  if (zrodlo==wyczysc)
  {dane.clear(); } //nie usuwa wyświetlanych danych
  else
  if(zrodlo==wyswietl)
  { lista.removeAllItems();
    Iterator iterator = dane.iterator();
    while(iterator.hasNext())
    { s=(String)iterator.next();
      lista.addItem(s); } }
  else
  if (zrodlo==usun)
  { s=(String)lista.getSelectedItem();
    lista.removeItem(s);
    dane.remove(s);
```



```

        dane.trimToSize();
    }
    repaint(); } }
public class baza9_0
{ Vector <String> dane=new Vector<String> (5,5);
  static public void main(String arg[])
  { baza9_0 baza = new baza9_0();
    try
    { Okno okno = new Okno(baza.dane);
      okno.setVisible(true);
    } catch(Exception e)
    { System.out.println("Blad bazy "+e); }
  }
}
}

```

2. **Wykonaj program** (klasa *linkedlist\_*), który opiera się na programie *Tytuly*, podanym w projekcie *Wektor*. Zamiast pojemnika *Vector* zastosować pojemnik typu *LinkedList* – należy poprawić lub usunąć wskazane przez kompilator metody, które pojemnik *LinkedList* nie posiada (projekty: *Pojemniki*, *Iteratory*).

2.1. Należy uzupełnić program o sortowanie uruchamiane za pomocą dodanego przycisku „Sortuj” (typu *JButton*) i wyszukiwanie połówkowe uruchamiane za pomocą dodanego przycisku „Szukaj” (typu *JButton*) po wprowadzeniu wyszukiwanego łańcucha w polu tekstowym oznaczonym etykietą „Podaj tytuł książki” i wyświetlając w dodanym polu tekstowym typu *JTextField* numer znalezionej elementu- obie funkcje zrealizować za pomocą metod klasy usługowej *Collections* w metodzie *actionPerformed*. Przykład zastosowania klasy *Collections* podano w projekcie *Pojemniki*.

2.2. Należy w metodzie *actionPerformed* uzupełnić obsługę usuwania wybranego elementu przez wyświetlanie numeru usuwanego elementu, wyznaczony za pomocą metody *indexOf* pojemnika *LinkedList*, która jako parametr otrzymuje łańcuch znaków przeznaczony do usuwania przez wybór z listy i zwraca numer elementu. Metodę tę należy wywołać przed wywołaniem metody *remove*. Otrzymany numer elementu należy wyświetlić w dodanym polu tekstowym, w którym wyświetlany jest wynik wyszukiwania połówkowego.

3. **Wykonaj program** (klasa *hashset\_*), który opiera się na programie *Tytul*, podanym w projekcie *Wektor*. Zamiast pojemnika *Vector* zastosować pojemnik typu *HashSet* – należy poprawić lub usunąć wskazane przez kompilator metody, które pojemnik *HashSet* nie posiada (projekty: *Zbiory*, *Pojemniki*, *Iteratory*). Należy w metodzie *actionPerformed* dodać obsługę wyszukiwania podanego łańcucha w polu tekstowym oznaczonym etykietą „Podaj tytuł książki” przez naciśnięcie dodanego przycisku „Szukaj” typu *JButton*. Wyszukiwanie należy zrealizować za pomocą metody *contains* pojemnika *HashSet*, która jako parametr otrzymuje podany łańcuch i zwraca wartość *true*, jeśli znaleziony został element lub *false*, jeśli nie znaleziono elementu. Wynik wyszukiwania należy wyświetlać w dodanym polu tekstowym typu *JTextField*.

4. **Wykonaj program** (klasa *hashmap\_*), który opiera się na programie *Tytul*, podanym w projekcie *Wektor*. Zamiast pojemnika *Vector* zastosować pojemnik typu *HashMap* – należy poprawić lub usunąć wskazane przez kompilator metody, które pojemnik *HashMap* nie posiada. Przykład zastosowania pojemnika *HashMap* podano w projektach *Pojemniki* (klasa *Kolekcja2*), *Mapy* (klasa *Mapy*), *Iteratory* (klasa *Kolekcje3*).

4.1. Należy w metodzie *actionPerformed* zmienić obsługę wstawiania elementów do pojemnika przez wylosowanie łańcucha, który jest kluczem dla wstawianej z klawiatury wartości (pierwszy parametr w metodzie *put*, która teraz zastępuje *add*). Sposób losowania klucza można oprzeć na przykładzie podanym w projekcie *Pojemniki* (klasa *Kolekcja2*).

4.2. Należy w metodzie *actionPerformed* zmienić obsługę wyświetlania, która powinna dodatkowo wyświetlać wylosowany klucz w liście typu *JComboBox*. Jest to możliwe, jeśli w obsłudze wyświetlania *iterator* dla pojemnika typu *HashMap* zostanie pobrany za pomocą metody *entrySet().iterator()* (projekt *Iteratory*)

4.3. Należy w metodzie *actionPerformed* dodać obsługę wyszukiwania podanego łańcucha w polu tekstowym oznaczonym etykietą „Podaj tytuł książki” przez naciśnięcie dodanego przycisku „Szukaj” typu *JButton*. Wyszukiwanie należy zrealizować za pomocą metody *containsValue* pojemnika *HashMap* (wyszukiwanie według wartości), która jako parametr otrzymuje podany łańcuch i zwraca wartość *true*, jeśli znaleziony został element lub *false*, jeśli nie znaleziono elementu. Wynik wyszukiwania należy wyświetlać w dodanym polu tekstowym typu *JTextField*.