Języki i metody programowania Java
Lab5 – pliki

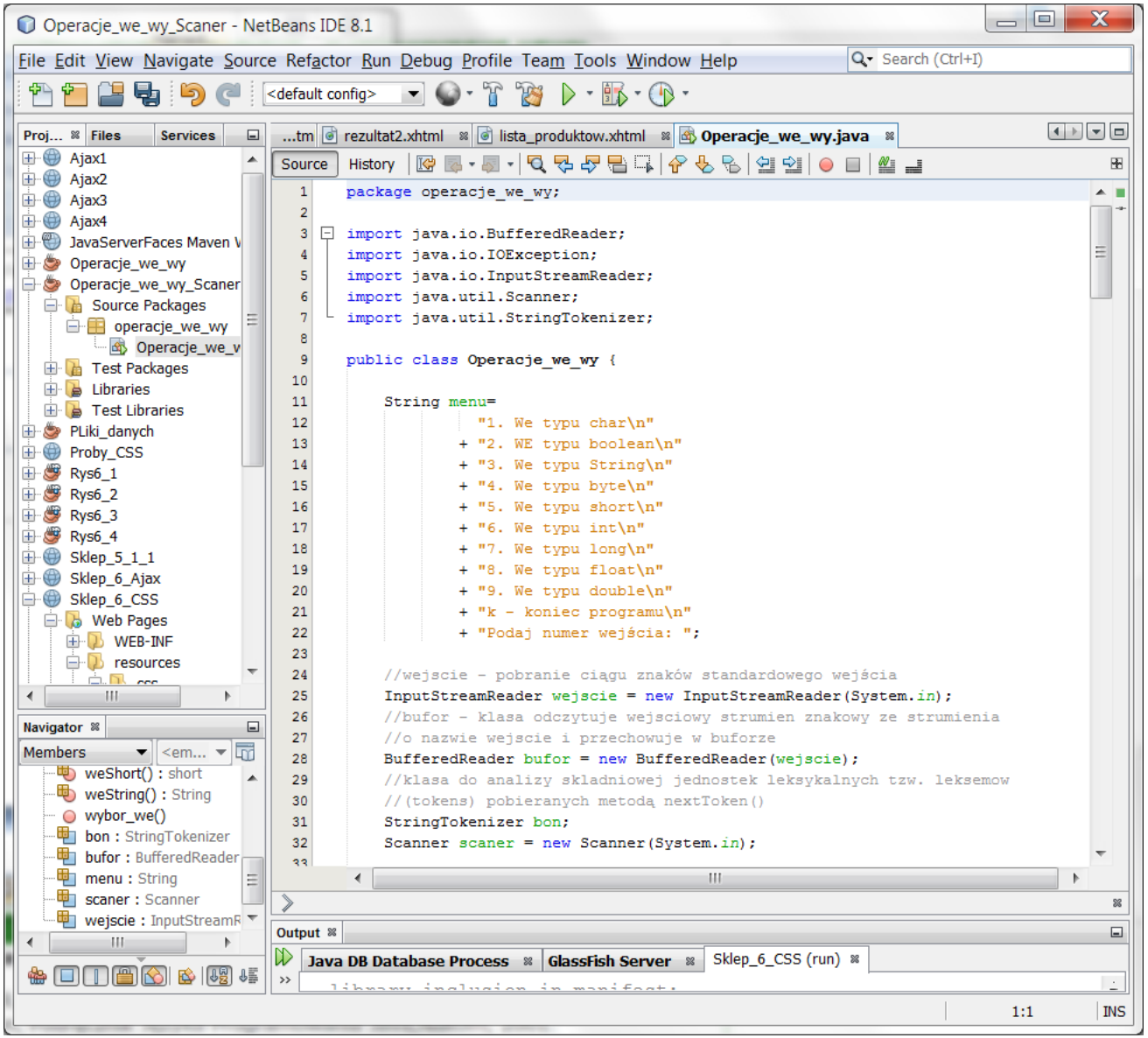
https://docs.oracle.com/javase/tutorial/

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/java_wyklad4.pdf

Zofia Kruczkiewicz

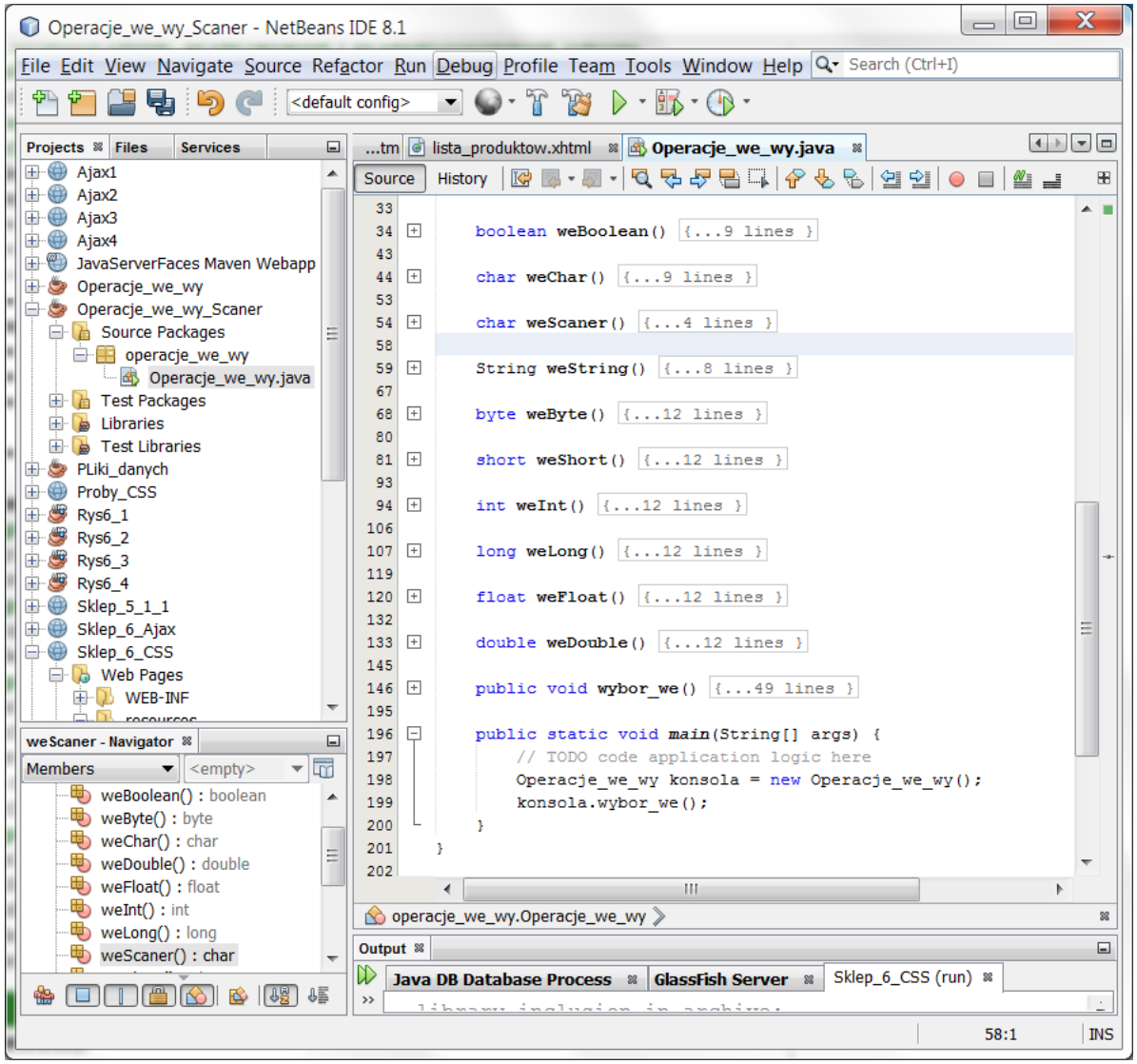
1. Należy dokonać analizy kodu programu Operacje_we_wy_Scaner

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Operacje_we_wy_Scaner.rar



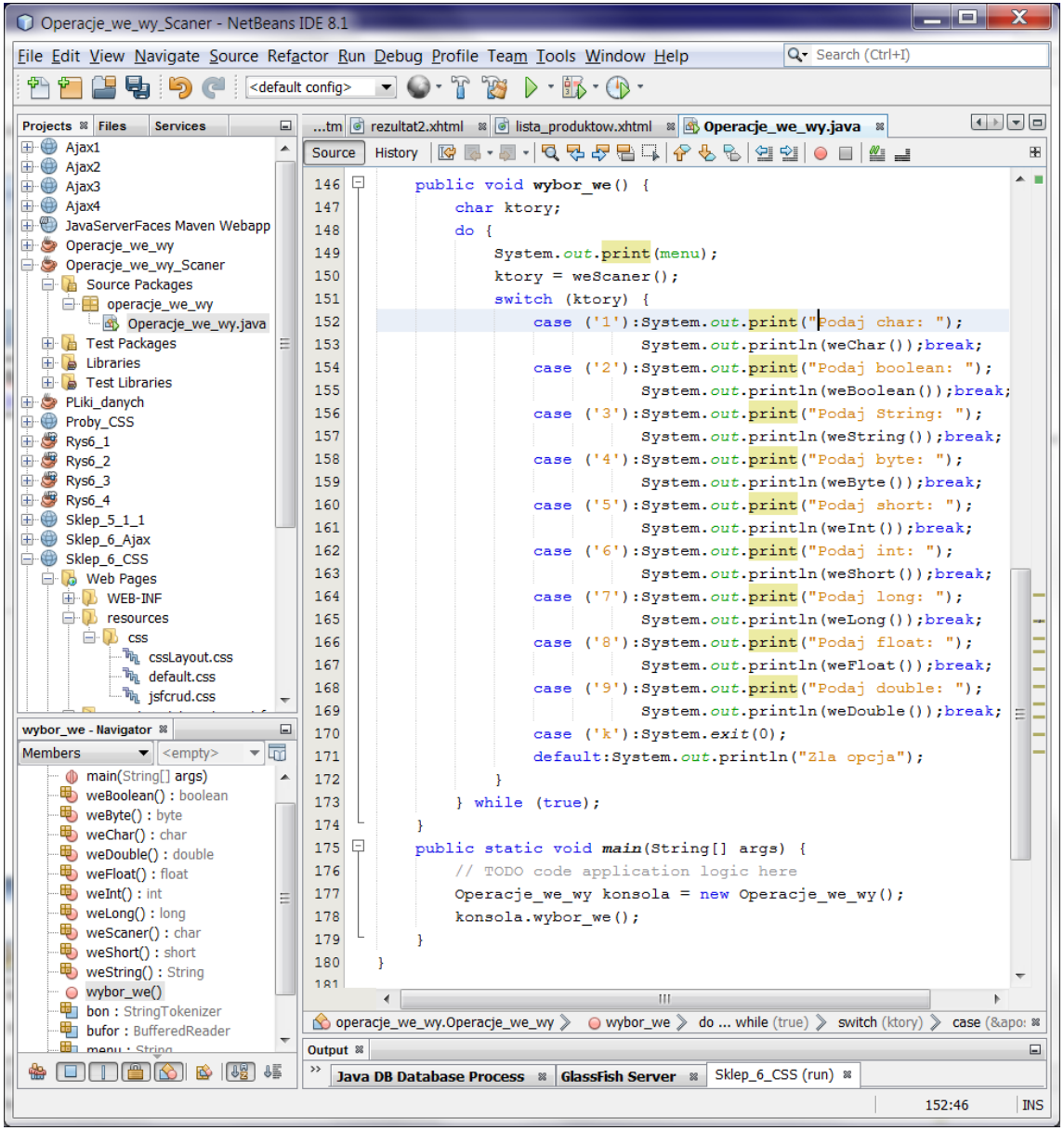
1. Należy dokonać analizy kodu programu Operacje_we_wy_Scaner (cd)

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Operacje_we_wy_Scaner.rar



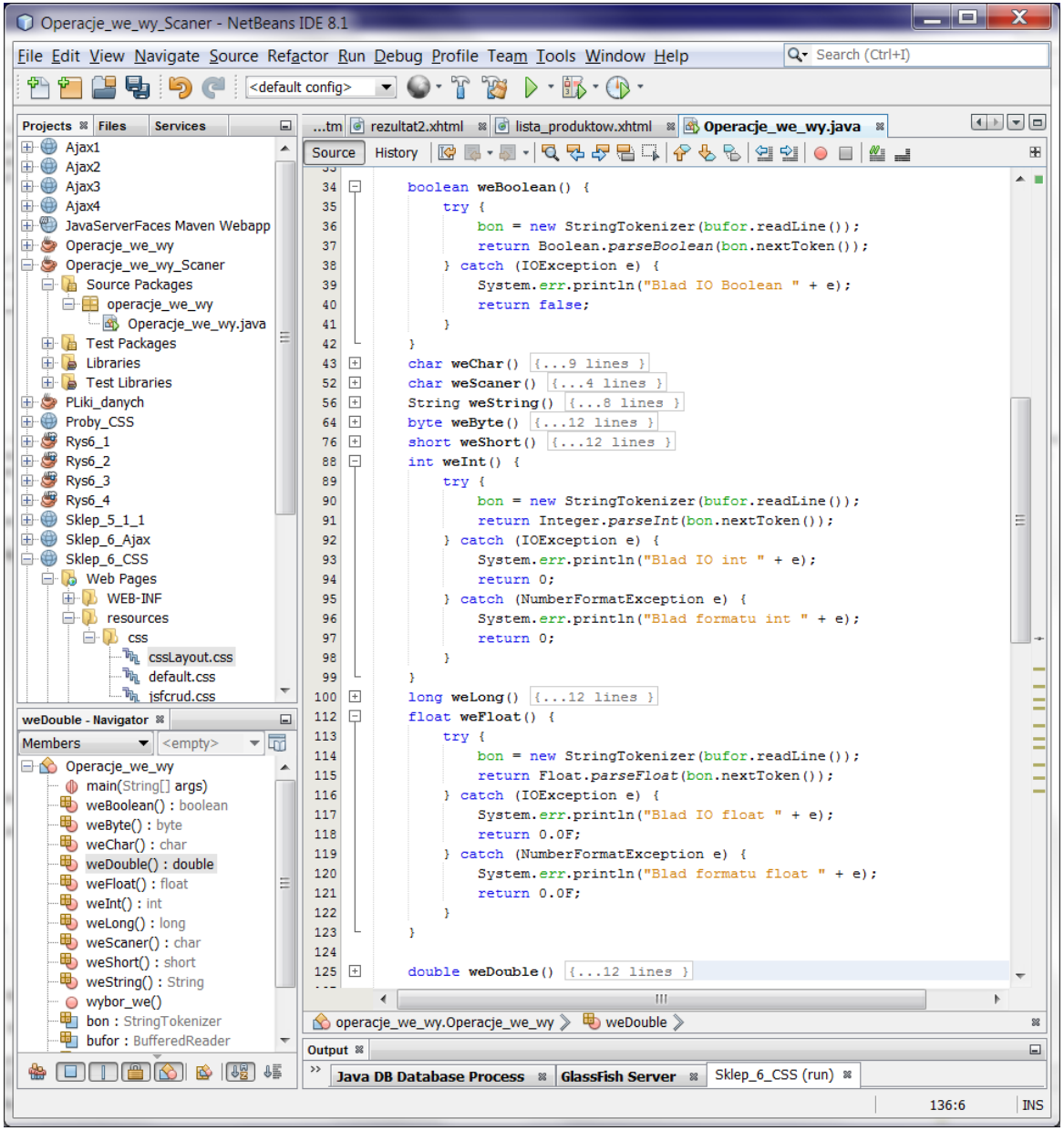
1. Należy dokonać analizy kodu programu Operacje_we_wy_Scaner (cd)

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Operacje_we_wy_Scaner.rar



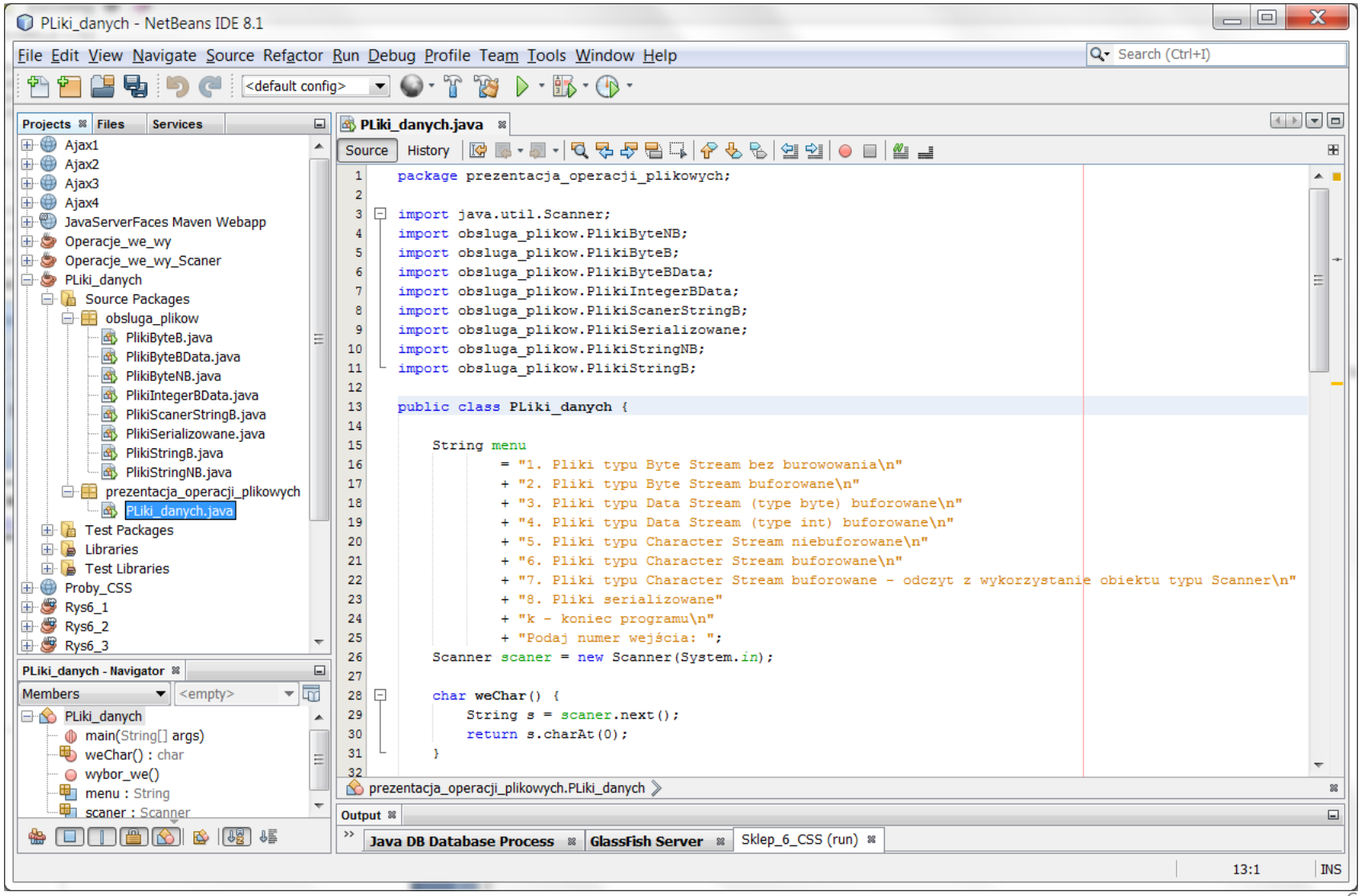
1. Należy dokonać analizy kodu programu Operacje_we_wy_Scaner (cd)

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Operacje_we_wy_Scaner.rar



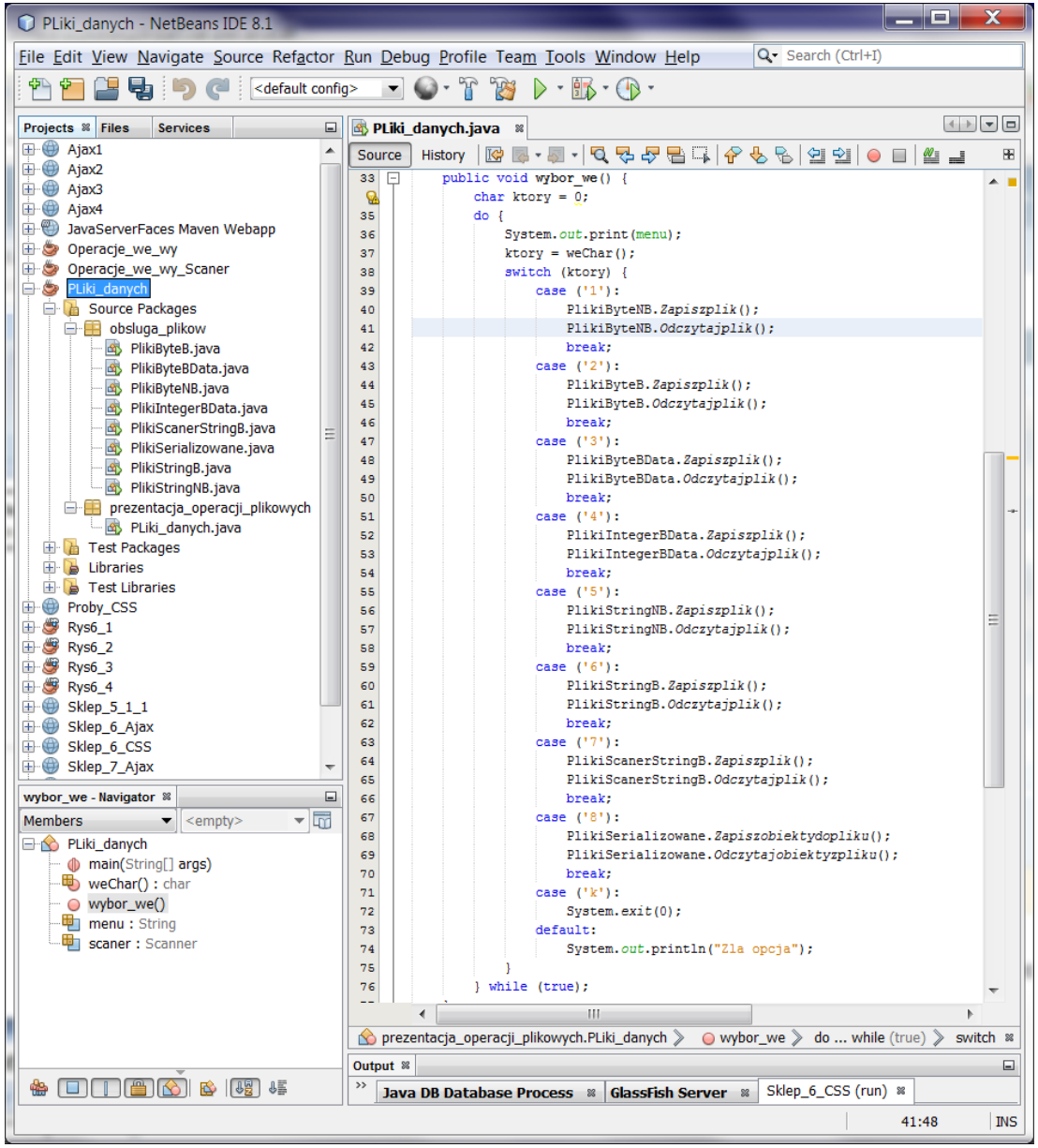
2. Należy dokonać analizy kodu programu Pliki_danych

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

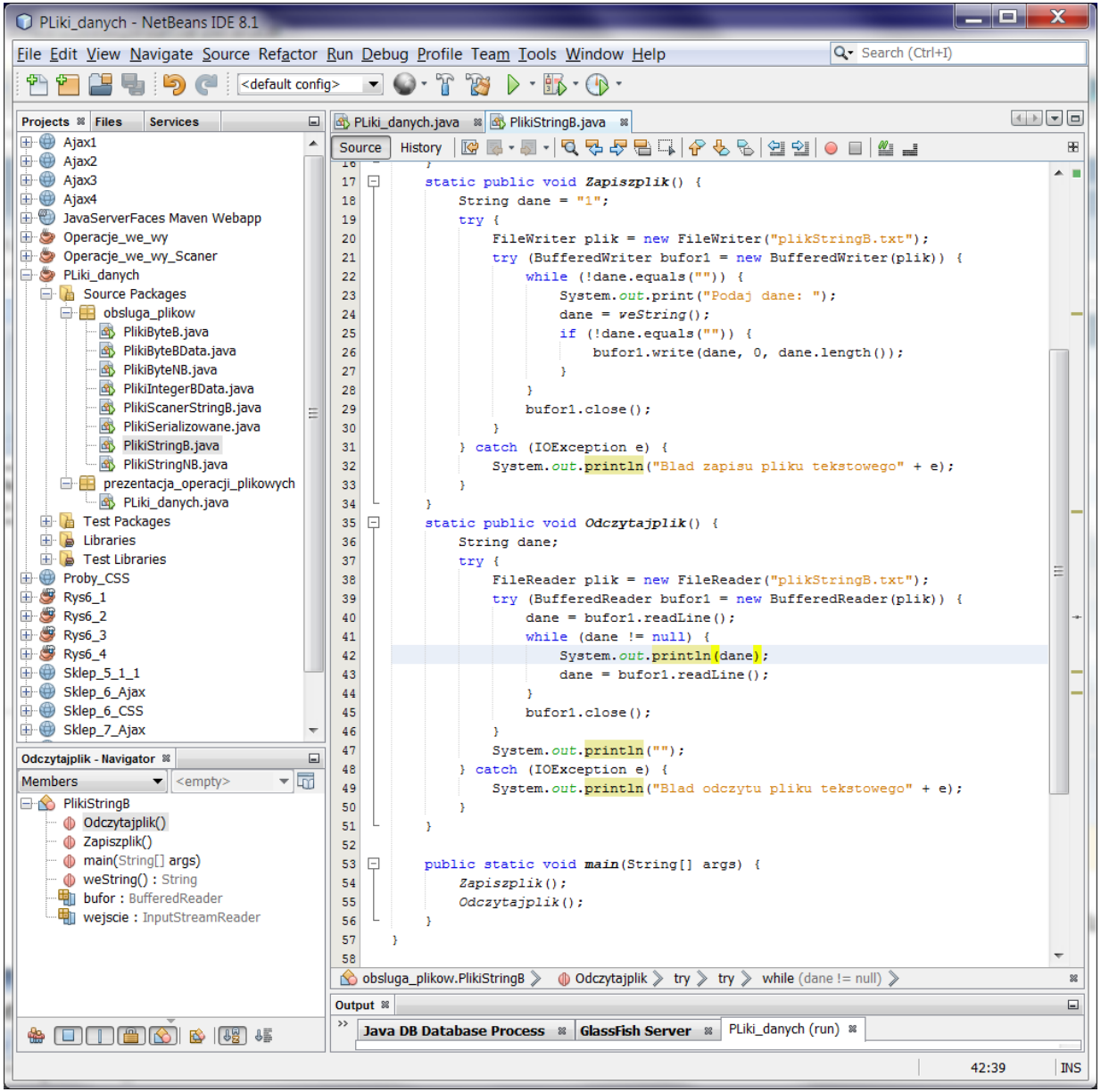


2. Należy dokonać analizy kodu programu Pliki_danych (cd)

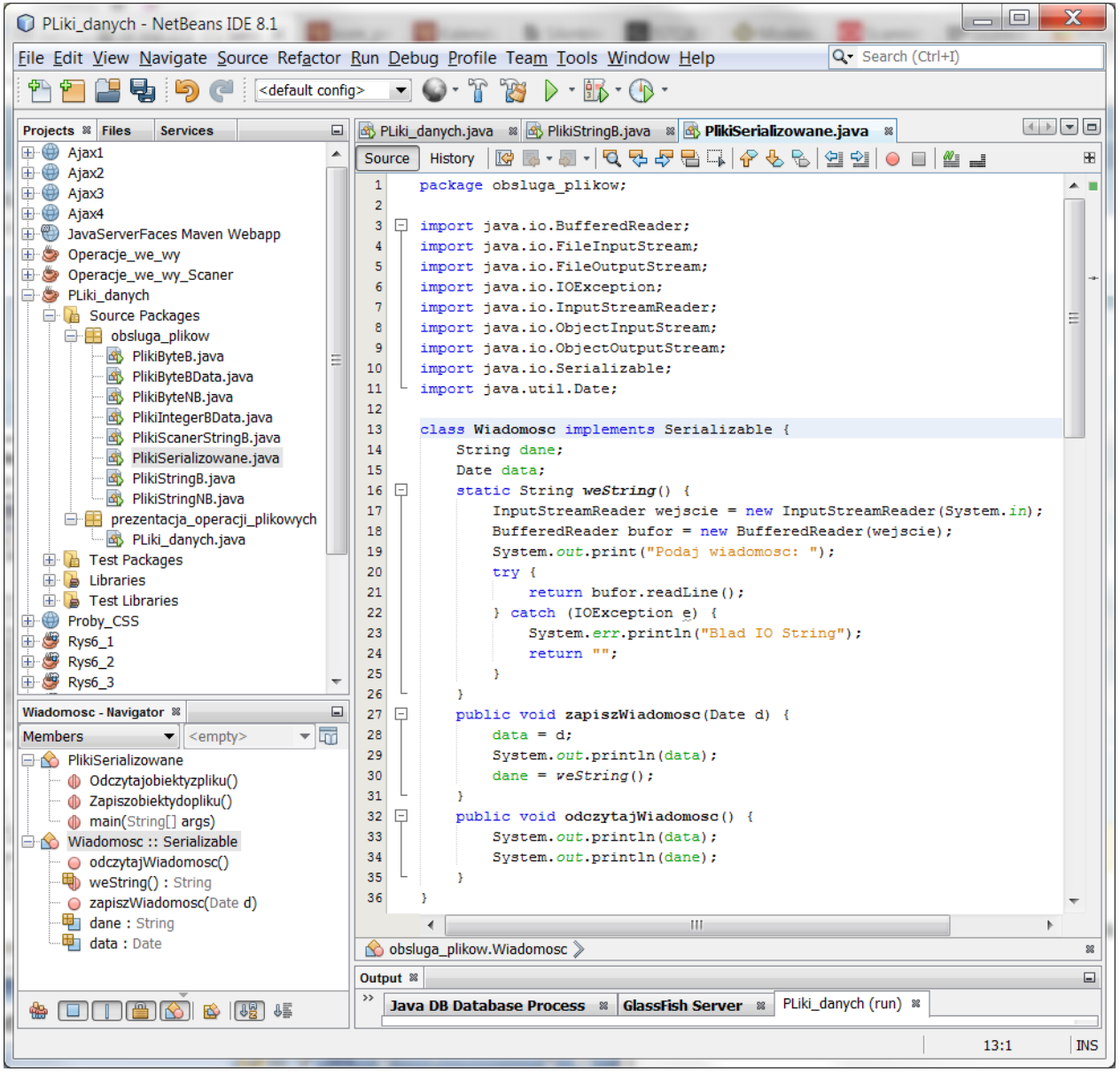
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar



2. Należy dokonać analizy kodu programu Pliki_danych (cd) – opcja 6 (klasa PlikiStringB)
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar



2. Należy dokonać analizy kodu programu Pliki_danych (cd) – opcja 8(klasa PlikiSerializowane)
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar



2. Należy dokonać analizy kodu programu Pliki_danych (cd) – opcja 8(klasa PlikiSerializowane)
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

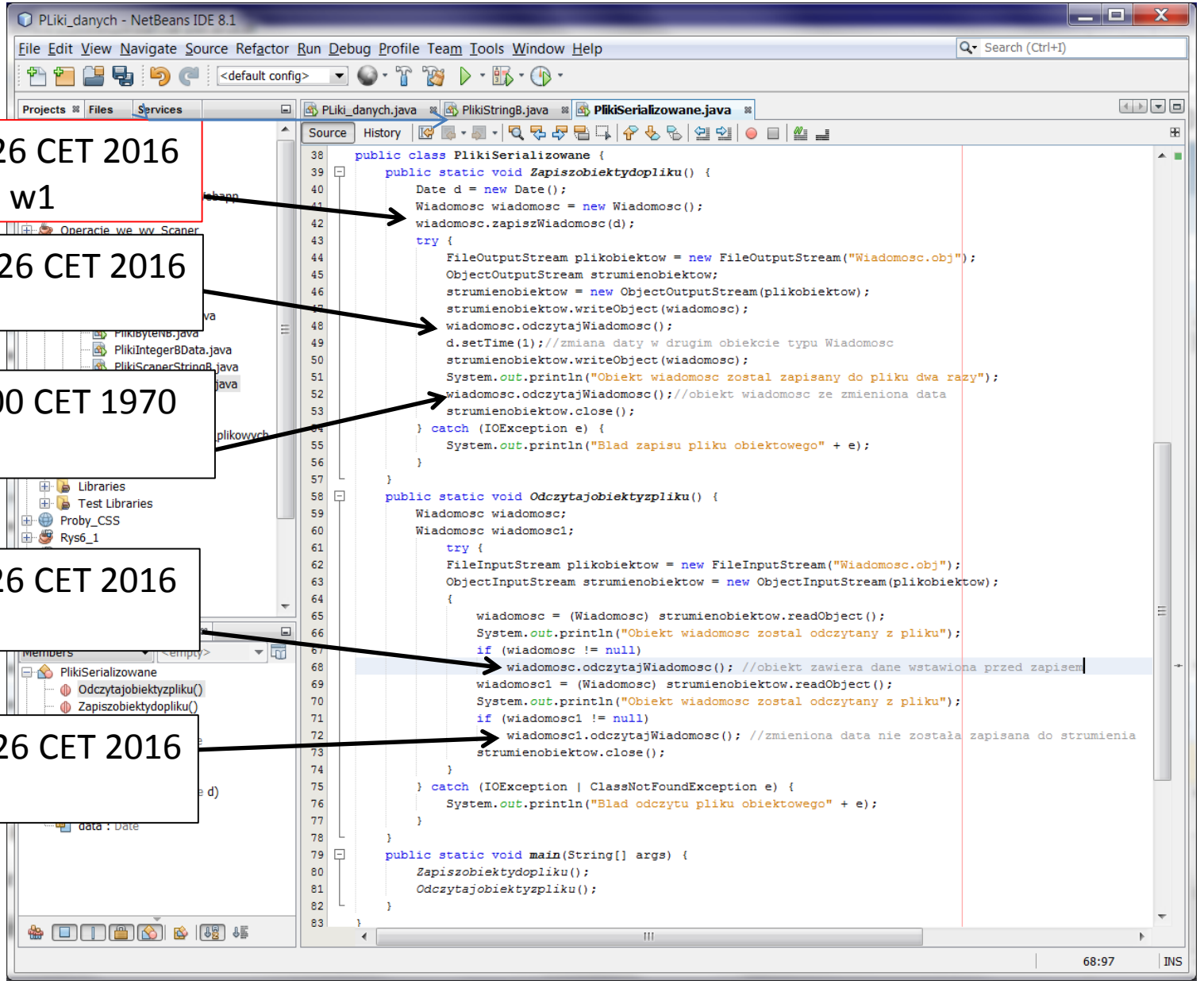
Sat Dec 17 17:10:26 CET 2016
Podaj wiadomosc: w1

Sat Dec 17 17:10:26 CET 2016
w1

Thu Jan 01 01:00:00 CET 1970
w1

Sat Dec 17 17:10:26 CET 2016
w1

Sat Dec 17 17:10:26 CET 2016
w1




```
38  public class PlikiSerializowane {
39      public static void Zapiszobiektydopliku() {
40          Date d = new Date();
41          Wiadomosc wiadomosc = new Wiadomosc();
42          wiadomosc.zapiszWiadomosc(d);
43          try {
44              FileOutputStream plikobiektow = new FileOutputStream("Wiadomosc.obj");
45              ObjectOutputStream strumienobiektow;
46              strumienobiektow = new ObjectOutputStream(plikobiektow);
47              strumienobiektow.writeObject(wiadomosc);
48              wiadomosc.odczytajWiadomosc();
49              d.setTime(1);//zmiana daty w drugim obiekcie typu Wiadomosc
50              strumienobiektow.writeObject(wiadomosc);
51              System.out.println("Obiekt wiadomosc zostal zapisany do pliku dwa razy");
52              wiadomosc.odczytajWiadomosc();//obiekt wiadomosc ze zmieniona data
53              strumienobiektow.close();
54          } catch (IOException e) {
55              System.out.println("Blad zapisu pliku obiektowego" + e);
56          }
57      }
58      public static void Odczytajobiektyzpliku() {
59          Wiadomosc wiadomosc;
60          Wiadomosc wiadomosc1;
61              try {
62              FileInputStream plikobiektow = new FileInputStream("Wiadomosc.obj");
63              ObjectInputStream strumienobiektow = new ObjectInputStream(plikobiektow);
64              {
65                  wiadomosc = (Wiadomosc) strumienobiektow.readObject();
66                  System.out.println("Obiekt wiadomosc zostal odczytany z pliku");
67                  if (wiadomosc != null)
68                      wiadomosc.odczytajWiadomosc(); //obiekt zawiera dane wstawiona przed zapisem
69                  wiadomosc1 = (Wiadomosc) strumienobiektow.readObject();
70                  System.out.println("Obiekt wiadomosc zostal odczytany z pliku");
71                  if (wiadomosc1 != null)
72                      wiadomosc1.odczytajWiadomosc(); //zmieniona data nie została zapisana do strumienia
73                  strumienobiektow.close();
74              }
75          } catch (IOException | ClassNotFoundException e) {
76              System.out.println("Blad odczytu pliku obiektowego" + e);
77          }
78      }
79      public static void main(String[] args) {
80          Zapiszobiektydopliku();
81          Odczytajobiektyzpliku();
82      }
83  }
```

3. Wykonanie pliku tekstowego (typu Character Stream)

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/PLiki_danych.rar

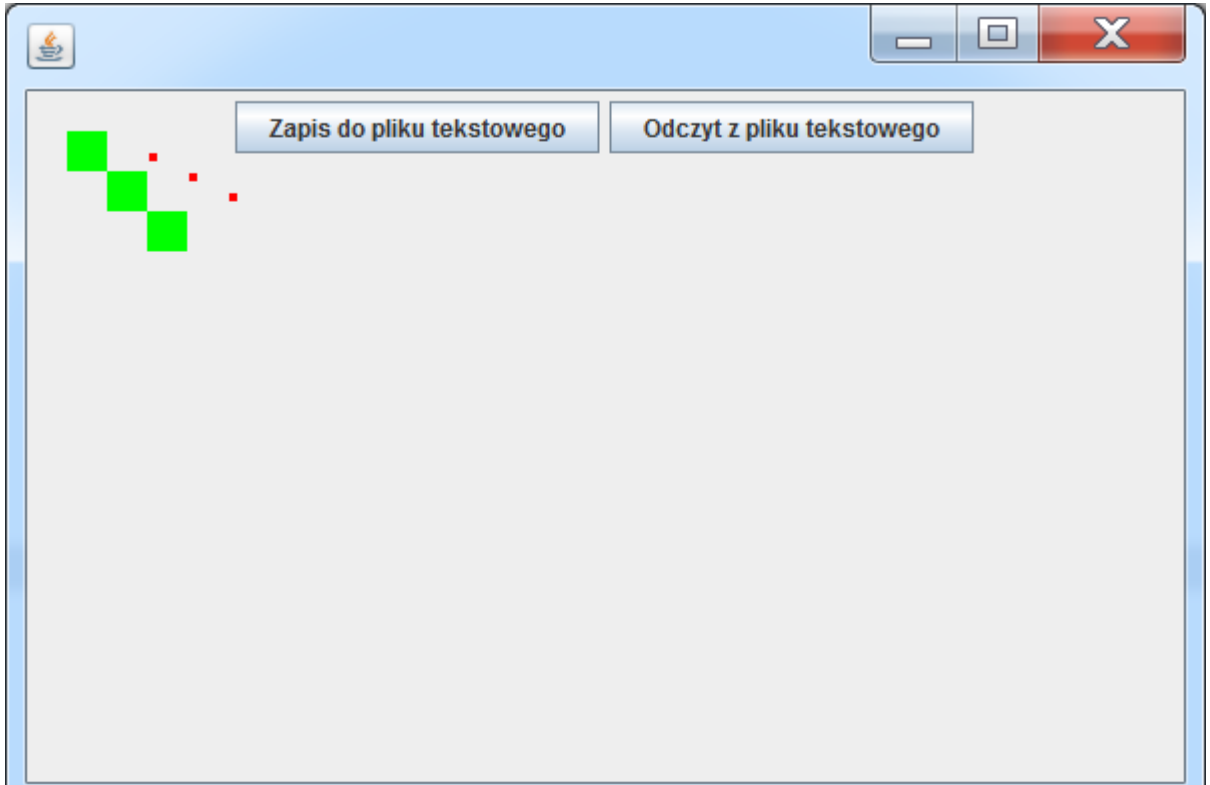
Wykorzystanie kodu klasy PlikiStringB

1. Należy w kopii programu wykonanego w lab4, jako rozwinięcie programu Rys1_1 dodać dwa przyciski JButton: jeden do zapisu danych o figurach, drugi do odczytu z pliku danych o figurach i wydruk na ekranie.

1.1. Należy wykonać kopię wybranego programu z p.1 (lub jego rozwinięcia)

1.2. Należy zmodyfikować kod metody toString() w klasach z rodziny Punkt lub pozostawić kod istniiejącej metody

1.3. Należy dodać dwa przyciski typu JButton w klasie Figury_panel (p.1.4)



3. Wykonanie pliku tekstowego (typu Character Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

1.4. Należy dodać w klasie Figury_panel dwa przyciski i obsługę zdarzeń dla tych przycisków.

```
public class Figury_panel extends JPanel implements ActionListener {
   FiguryHashSet kontroler;
   protected JButton zapis = new JButton("Zapis do pliku tekstowego");
   protected JButton odczyt = new JButton("Odczyt z pliku tekstowego");

@Override
   protected void paintComponent(Graphics g) {
      super.paintComponent(g);
      kontroler.rysuj_figury(g);
   }
   public void init() {
      kontroler = new FiguryHashSet();
      kontroler.pojemnik();
      kontroler.wypelnij();
      zapis.addActionListener(this);
      odczyt.addActionListener(this);
      add(zapis);
      add(odczyt);
   }
```

3. Wykonanie pliku tekstowego (typu Character Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

1.5. Należy dodać metodę obsługi zdarzeń **actionPerformed(ActionEvent evt)** w klasie Figury_panel.

```
public void actionPerformed(ActionEvent evt) {
        Object zrodlo = evt.getSource();
       if (zrodlo == zapis) {
          kontroler.Zapis_do_pliku();
        }
        else if (zrodlo == odczyt) {
          kontroler.Odczytaj_plik();
        }
        requestFocus();
                    //przywrocenie zdolnosci do obslugi zdarzen
        repaint();
    }
```

3. Wykonanie pliku tekstowego (typu Character Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

Wykorzystanie kodu klasy PlikiStringB

1.6**. Należy wykonać dwie nowe metody w klasie FiguryHashSet, stosując linię try do otwierania i zamykania strumieni (wg wykładu** http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/java_wyklad4.pdf **):**

```
public void Zapis_do_pliku() {
      try {
         FileWriter plik = new FileWriter("Figury_text.txt");
         BufferedWriter bufor1 = new BufferedWriter(plik);
/*   zapis w pętli łańcucha znaków zwracanego  z metody toString() wywołanej od
każdego elementu z tablicy  figury*/
}

 public void Odczytaj_plik() {
      String dane;
      try {
         FileReader plik = new FileReader("Figury_text.txt");
         BufferedReader bufor1 = new BufferedReader(plik);
   /* odczyt zawartości pliku tekstowego na ekran  */
}
```

3. Wykonanie pliku tekstowego (typu Character Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

1.7. W klasie Obraz1_1 należy zmodyfikować rozmiar ramki

```
void rysunek_Swing() {
        JFrame ramka = new JFrame();
        Figury_panel panel = new Figury_panel();
        panel.init();
      ramka.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        ramka.setSize(600, 400);
        ramka.setContentPane(panel);
        ramka.setVisible(true);
    }
```

1.8. Przykład wyniku tworzenia i wyświetlania zawartości pliku z informacją o figurach

```
Punkt{x=60, y=30}
Kwadrat{dlugosc=20} i dziedzicze od
Punkt{x=20, y=20}
Kwadrat{dlugosc=20} i dziedzicze od
Punkt{x=40, y=40}
Kwadrat{dlugosc=20} i dziedzicze od
Punkt{x=60, y=60}
Punkt{x=100, y=50}
Punkt{x=80, y=40}
```

4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd

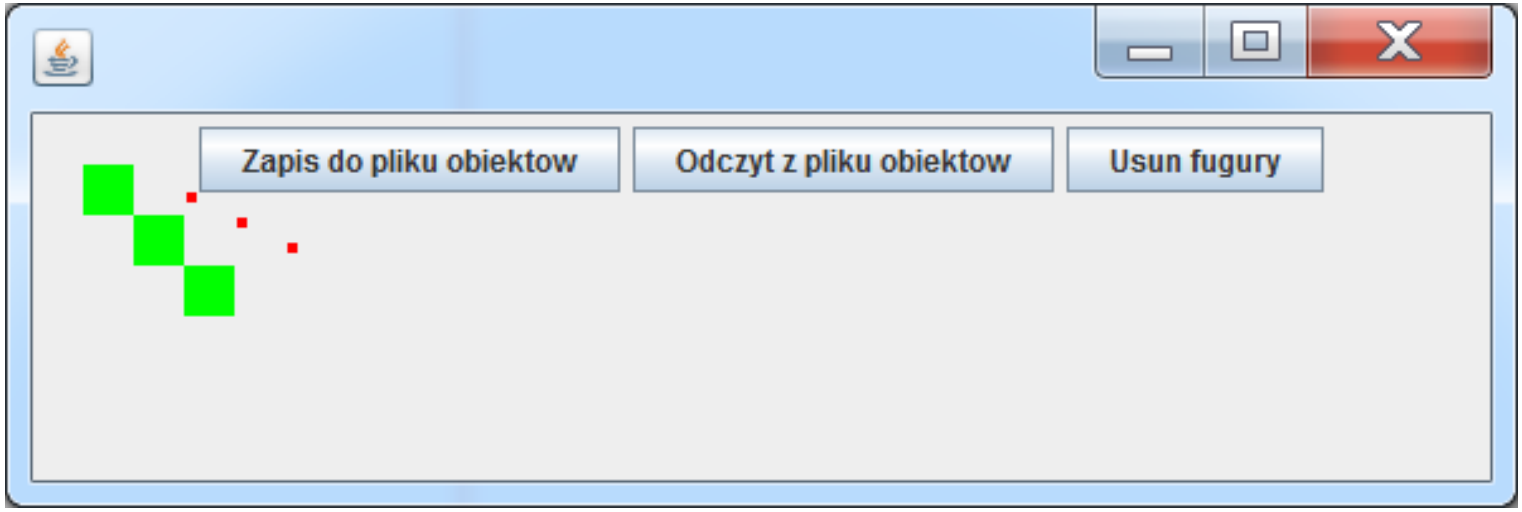
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

Wykorzystanie kodu klasy **PlikiSerializowane**

1. Należy w kopii programu wykonanego w lab4, jako rozwinięcie programu Rys1_2 dodać trzy przyciski JButton:jeden do zapisu obiektów z kolekcji **figury** z rodziny Punkt, drugi do odczytu z pliku kolekcji **figury** z obiektami z rodziny Punkt, a trzeci do usunięcia wszystkich obiektów z kolekcji figury.

1.1. Należy wykonać kopię wybranego programu z p.1 (lub jego rozwinięcia)

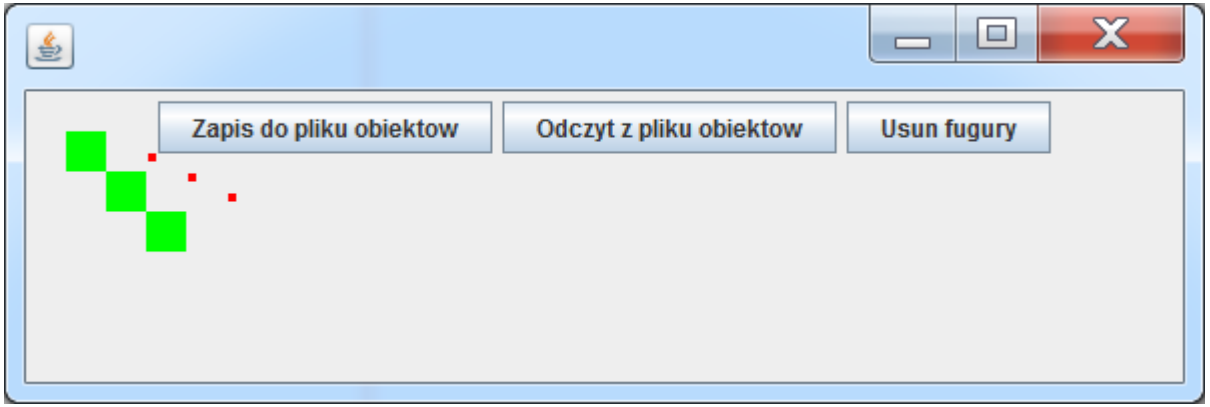
1.2. Należy dodać trzy przyciski typu JButton w klasie Figury_panel (p.1.4)



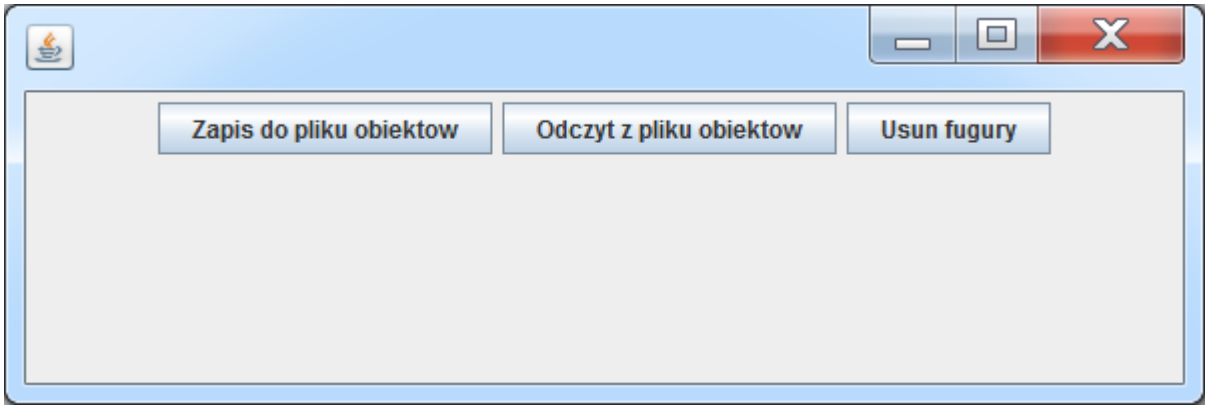
4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

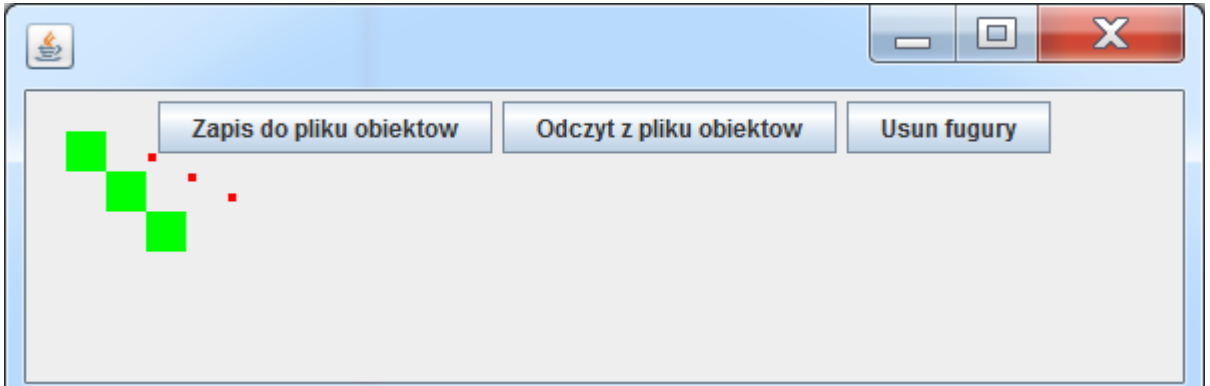
Wykorzystanie kodu klasy **PlikiSerializowane**



Zapis figur do pliku za pomocą przycisku **Zapisz do pliku obiektow**



Usunięcie figur z programu za pomoca przycisku **Usun figury**



Odczyt figur z pliku i zapis w kolekcji figury **Odczyt z pliku obiektow**

4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

Wykorzystanie kodu klasy **PlikiSerializowane**

1.3. Należy dodać w klasie Figury_panel trzy przyciski i obsługę zdarzeń dla tych przycisków.

```
public class Figury_panel extends JPanel implements ActionListener {
   FiguryHashSet kontroler;
   protected JButton zapis = new JButton("Zapis do pliku obiektow");
   protected JButton odczyt = new JButton("Odczyt z pliku obiektow");
   protected JButton usun=new JButton("Usun fugury");

@Override
   protected void paintComponent(Graphics g) {
      super.paintComponent(g);
      kontroler.rysuj_figury(g); }

 public void init()
   {   kontroler=new FiguryArrayList();
      kontroler.pojemnik();
      kontroler.wypelnij();
      zapis.addActionListener(this);
      odczyt.addActionListener(this);
      usun.addActionListener(this);
      add(zapis);
      add(odczyt);
      add(usun);  }
```

4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

Wykorzystanie kodu klasy **PlikiSerializowane**

1.4. Należy dodać metodę obsługi zdarzeń **actionPerformed(ActionEvent evt)** w klasie Figury_panel.

```
public void actionPerformed(ActionEvent evt) {
        Object zrodlo = evt.getSource();
        if (zrodlo == zapis) {
            kontroler.Zapis_do_pliku();
        } else if (zrodlo == odczyt) {
            kontroler.Odczytaj_plik();
        }else if (zrodlo == usun) {
            kontroler.figury.clear();
         }
        requestFocus();
        repaint();
    }
```

4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd
http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar
Wykorzystanie kodu klasy **PlikiSerializowane**

1.5**. Należy wykonać dwie nowe metody w klasie FiguryArrayList stosując linię try do otwierania i zamykania strumieni (wg wykładu** http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/java_wyklad4.pdf **)**

```
public void Zapis_do_pliku() {
  try {
     FileOutputStream plikobiektow = new FileOutputStream("Wiadomosc.obj");
     ObjectOutputStream strumienobiektow = new ObjectOutputStream(plikobiektow);

   /*   Zapis kolekcji figury*/
}

public void Odczytaj_plik() {
       try {
          FileInputStream plikobiektow = new FileInputStream("Wiadomosc.obj");
          ObjectInputStream strumienobiektow = new ObjectInputStream(plikobiektow);

    /* Odczyt kkolekcji figury.  To pozwala odtworzyć stan programu przed zapisem figur
do pliku.*/
}
```

4. Wykonanie pliku serializowanych obiektów (typu Object Stream) cd

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/pojava/Pliki_danych.rar

Wykorzystanie kodu klasy **PlikiSerializowane**

1.7. W klasie Obraz1_1 należy zmodyfikować rozmiar ramki

```
void rysunek_Swing() {
        JFrame ramka = new JFrame();
        Figury_panel panel = new Figury_panel();
        panel.init();
       ramka.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        ramka.setSize(600, 400);
        ramka.setContentPane(panel);
        ramka.setVisible(true);
    }
```