

## Pliki dyskowe

### 1. Procedura korzystania ze strumieni tekstowych wejściowych powiązanych ze standardowym wejściem (konsola) – wprowadzanie danych z klawiatury

- 1) Należy utworzyć obiekt (np. typu *InputStreamReader*), powiązany ze źródłem danych (np. *System.in*)  
*InputStreamReader wejście = new InputStreamReader( System.in );*
- 2) W celu buforowania znaków pochodzących z obiektu powiązanego ze źródłem danych np. typu *InputStreamReader* należy utworzyć obiekt klasy *BufferedReader*  
*BufferedReader bufor = new BufferedReader( wejście );*  
lub *BufferedReader bufor = new BufferedReader( wejście, int );* //int-rozmiar bufora
- 3) Znaki mogą być odczytywane metodami obiektu buforującego:  
*int read()*  
np. *int dana = bufor.read()*
- 4) Całe ciągi znaków mogą być odczytane za pomocą:  
*int read(char[] cbuf, int off, int len)* – metoda, która czyta do tablicy *cbuf* od indeksu *off* liczbę *len* znaków i zwraca przez **return** liczbę faktycznie odczytanych znaków
- 5) Całe wiersze mogą być odczytywane za pomocą metody obiektu buforującego:  
*String readLine()*  
np. *String dana = bufor.readLine(),*  
która czyta łańcuch typu *String* lub jego wartość **null**, gdy będzie osiągnięty koniec łańcucha (czyli koniec linii oznaczony znakami np.: `'\n', '\r'` )
- 6) Analiza odczytanego łańcucha może być przeprowadzona za pomocą obiektu klasy typu *StringTokenizer*  
*StringTokenizer bon = new StringTokenizer(bufor.readLine());*
- 7) Innym sposobem jest analiza odczytanego łańcucha typu *String* bezpośrednio za pomocą metod klasy *String*

```

import java.io.*;
import java.util.*;

public class WEWY           //plik WEWY.java
{
    //wejściowy strumień tekstowy odczytuje strumien stdin (System.in)
    static InputStreamReader wejście = new InputStreamReader( System.in );

    //bufor – klasa odczytuje wejściowy strumień znakowy wejście i przechowuje odczytywane znaki w buforze

    static BufferedReader bufor = new BufferedReader( wejście );

    StringTokenizer bon;     //klasa do analizy składniowej jednostek leksykalnych tzw. Leksemów (tokens) pobieranych metodą nextToken()

boolean weBoolean()
{ try
    { bon = new StringTokenizer(bufor.readLine());
      return new Boolean(bon.nextToken()).booleanValue();  }
  catch (IOException e)
    { System.err.println("Bład IO Boolean "+e);
      return false;  }
  catch (Exception e)
    { System.err.println( "Bład Boolean "+e);
      return false;  }
}

String weString()
{ try
    { return bufor.readLine(); }
  catch (IOException e)
    { System.err.println("Bład IO String");
      return ""; }
  catch (Exception e)
    { System.err.println( "Bład String "+e);
      return "";  } }

```

**char** weChar()

```
{ try
  { String s = bufor.readLine();
    return s.charAt(0); }
  catch (IOException e)
  { System.err.println("Blad IO char "+e);
    return 0; }
  catch (Exception e)
  { System.err.println( "Blad char "+e);
    return 0; }
}
```

**byte** weByte()

```
{ try
  { bon = new StringTokenizer(bufor.readLine());
    return Byte.parseByte(bon.nextToken());
  }
  catch (IOException e)
  { System.err.println("Blad IO byte "+e);
    return 0; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu byte "+e);
    return 0; }
  catch (Exception e)
  { System.err.println( "Blad formatu byte "+e);
    return 0; }
}
```

## **short** weShort()

```
{ try
    { bon = new StringTokenizer(bufor.readLine());
      return Short.parseShort(bon.nextToken()); }
  catch (IOException e)
  { System.err.println("Blad IO short: "+e);
    return 0; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu short "+e);
    return 0; }
  catch (Exception e)
  { System.err.println( "Blad formatu short "+e);
    return 0; }
}
```

## **int** weInt()

```
{ try
    { bon = new StringTokenizer(bufor.readLine());
      return Integer.parseInt(bon.nextToken()); }
  catch (IOException e)
  { System.err.println("Blad IO int "+e);
    return 0; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu int "+e);
    return 0; }
  catch (Exception e)
  { System.err.println( "Blad formatu int "+e);
    return 0; }
}
```

## **long** weLong()

```
{ try
    { bon = new StringTokenizer(bufor.readLine());
      return Long.parseLong(bon.nextToken()); }
  catch (IOException e)
  { System.err.println("Blad IO "+e);
    return 0L; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu long "+e);
    return 0L; }
  catch (Exception e)
  { System.err.println( "Blad formatu long "+e);
    return 0L; }
}
```

```

float weFloat()
{ try
  { bon = new StringTokenizer(bufor.readLine());
    return new Float(bon.nextToken()).floatValue(); }
  catch (IOException e)
  { System.err.println("Blad IO float "+e);
    return 0.0F; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu float "+e);
    return 0.0F; }
  catch (Exception e)
  { System.err.println( "Blad formatu float "+e);
    return 0.0F; }
}

```

```

double weDouble()
{ try
  { bon = new StringTokenizer(bufor.readLine());
    return new Double(bon.nextToken()).doubleValue(); }
  catch (IOException e)
  { System.err.println("Blad IO double "+e);
    return 0.0; }
  catch (NumberFormatException e)
  { System.err.println( "Blad formatu double "+e);
    return 0; }
  catch (Exception e)
  { System.err.println( "Blad formatu double "+e);
    return 0.0; }
}

```

```

public static void main (String args[])
{
    WEWY we = new WEWY();
    System.out.print("Podaj char: ");
    System.out.println(we.weChar());

    System.out.print("Podaj String: ");
    System.out.println(we.weString());

    System.out.print("Podaj boolean: ");
    System.out.println(we.weBoolean());

    System.out.print("Podaj byte: ");
    System.out.println(we.weByte());

    System.out.print("Podaj short: ");
    System.out.println(we.weShort());

    System.out.print("Podaj int: ");
    System.out.println(we.weInt());

    System.out.print("Podaj long: ");
    System.out.println(we.weLong());

    System.out.print("Podaj float: ");
    System.out.println(we.weFloat());

    System.out.print("Podaj double: ");
    System.out.println(we.weDouble());
}
}

```

```

c:\Program Files\Xinox Software\JCreator LE\GE2001.exe
Podaj char: a
a
Podaj String: asd
asd
Podaj boolean: tu
false
Podaj byte: f
Blad formatu byte java.lang.NumberFormatException: For input string: "f"
0
Podaj short: 2
2
Podaj int: 5
5
Podaj long: o
Blad formatu long java.lang.NumberFormatException: For input string: "o"
0
Podaj float: 3
3.0
Podaj double: 123.5
123.5
Press any key to continue...

```

## 2. Strumienie tekstowe plikowe

### Procedura korzystania ze strumieni tekstowych buforowanych powiązanych z plikami tekstowymi

#### Aby utworzyć plik:

- 1) Należy utworzyć obiekt (np. typu *FileWriter*), powiązany ze plikiem danych tekstowych (np. "plik2.txt");  
*FileWriter plik = new FileWriter("plik2.txt");*
- 2) W celu buforowania znaków pochodzących z obiektu powiązanego ze źródłem danych np. typu *FileWriter* służy obiekt klasy *BufferedWriter*  
*BufferedWriter bufor = new BufferedWriter( plik );*
- 3) Znaki mogą być zapisywane do pliku za pomocą metody bufora:  
*void write(int c)*
- 4) Całe ciągi znaków mogą być zapisywane do pliku za pomocą metody bufora:  
*void write(char[] cbuf, int off, int len)* – metoda, która czyta z tablicy *cbuf* od indeksu *off* liczbę *len* znaków i zapisuje do pliku
- 5) Część łańcucha można zapisać do pliku za pomocą metody bufora:  
*void write(String str, int off, int len)* – metoda, która czyta z łańcucha *str* od indeksu *off* liczbę *len* znaków i zapisuje do pliku

#### Aby odczytać plik:

- 6) Należy utworzyć obiekt (np. typu *FileReader*), powiązany ze plikiem danych tekstowych (np. "plik2.txt");  
*FileReader plik = new FileReader("plik2.txt");*
- 7) Do buforowania znaków pochodzących z obiektu powiązanego ze źródłem danych np. typu *FileWriter* służy obiekt klasy *BufferedReader*  
*BufferedReader bufor = new BufferedReader( plik );*
- 8) Znaki mogą być odczytywane metodą bufora:  
*int read ();* np. *int dane = plik.read()*
- 9) Całe ciągi znaków mogą być odczytane za pomocą metody bufora:  
*int read(char[] cbuf, int off, int len)* – metoda, która czyta plik i zapisuje do tablicy *cbuf* od indeksu *off* liczbę *len* znaków i zwraca przez **return** liczbę faktycznie odczytanych znaków
- 10) Całe wiersze mogą być odczytywane za pomocą metody obiektu buforującego:  
*String readLine()* np. *String dana = bufor.readLine(),*  
która czyta w pliku łańcuch typu *String* lub jego wartość **null**, gdy osiągnięto koniec łańcucha (koniec linii oznaczony znakami np.: '\n', '\r' )
- 11) Po zapisie i odczycie *bufor* należy zamknąć metodą *close()*

**package** baza1;

//przykład prezentowany na wykładach z tablic (Przykład 6– część 1)

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;
import java.util.*;
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); }
}
```



//przykład prezentowany na wykładach z tablic (Przykład 6 – część 1)

```
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 Baza1

```
{ protected Tablica tablica=new Tablica();
```

```
public void Wyswietl()
```

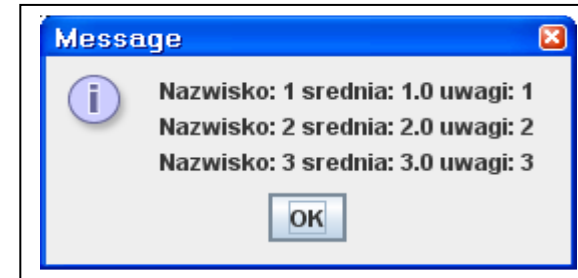
```
{ if (tablica.Pusta()) //jeśli tablica istnieje, to można wyświetlać
    JOptionPane.showMessageDialog(null, "Brak danych");
else
    JOptionPane.showMessageDialog(null, tablica.toString());
}
```

```
public void Wypelnij()
```

```
{ String S;
  S=JOptionPane.showInputDialog(null,"Podaj rozmiar tablicy");
  int N=Integer.parseInt(S);
  tablica.Wykonaj_tablice(N);
  String S1,S2,S3;
  while(!tablica.Pelna())
  {
    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);
  }
}
```

```
public void Wyszukaj()
```

```
{
  if (tablica.Pusta()) //jeśli tablica istnieje, to można wyszukiwać
    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());
  }
}
```



//utworzenie tablicy

//jeśli tablica jest zapełniona, tak jest równe false

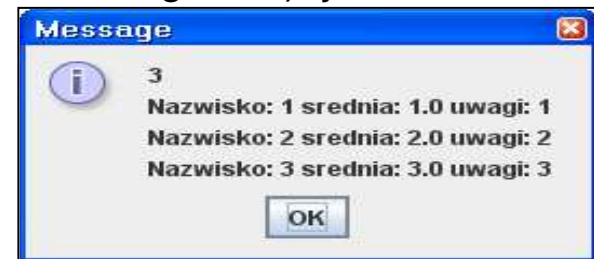
//dopisanie danych na końcu danych

Przykład  
prezentowany  
na wykładach  
z tablic  
(Przykład 6 –  
część 1

```

public void Zapisz_do_pliku_tekstowego()
{ String S;
  if (tablica.Pusta()) JOptionPane.showMessageDialog(null,"Brak danych");//jesli tablica istnieje, to mozna zapisać do pliku
  else
  try
  { FileWriter plik = new FileWriter ("baza1.txt"); //utworzenie strumienia wyjściowego
    BufferedWriter bufor = new BufferedWriter (plik); //utworzenie bufora wejściowego
    S=tablica.Podaj_ile()+"\n"; //zapisanie liczby elementów w pierwszym wierszu
    S+=tablica.toString(); //zapisanie zawartości tablicy w postaci łańcucha
    bufor.write(S, 0, S.length()); //zapisanie do pliku przygotowanego łańcucha
    bufor.close(); //zamknięcie pliku i strumieni
  } catch (IOException e)
    { JOptionPane.showMessageDialog(null, "Bład zapisu pliku tekstowego" +e); }
}

```



```

public void Odczytaj_plik_tekstowy_na_ekran()
{ String dane,S="";
  try //utworzenie strumienia wejściowego
  { FileReader plik = new FileReader ("baza1.txt");
    BufferedReader bufor = new BufferedReader (plik); //utworzenie bufora wejściowego
    dane=bufor.readLine(); //czytanie 1 linii pliku tekstowego bez znaków końca linii
    while (dane!=null) //czytanie pliku aż do wyczerpania
    { S+=dane+"\n"; //zapisanie kolejnego wiersza odczytanego z pliku do łańcucha S
      dane=bufor.readLine(); } //czytanie kolejnej linii pliku tekstowego bez znaków końca linii
    bufor.close(); //zamknięcie pliku i strumieni
    JOptionPane.showMessageDialog(null,S); //wyświetlenie zawartości pliku na ekranie
  } catch (IOException e)
    { JOptionPane.showMessageDialog(null, "Bład odczytu pliku tekstowego " +e); } }

```

```

public void Odczytaj_plik_tekstowy_do_tablicy()
{ String dane="0";
  try
  { FileReader plik = new FileReader ("baza1.txt");           //utworzenie strumienia wejściowego
    BufferedReader bufor = new BufferedReader (plik);         //utworzenie bufora wejściowego
    dane=bufor.readLine();                                   //odczytanie liczby elementów
    int ile=Integer.parseInt(dane);                         //zamiana liczby wierszy z postaci łańcucha na postać typu int
    tablica.Wykonaj_tablice(ile);                           //wykonanie tablicy o rozmiarze równym liczbie wierszy, czyli liczbie danych obiektów
    dane=bufor.readLine();                                  //odczytanie pierwszego wiersza
    while (dane!=null)                                     //czytanie pliku aż do wyczerpania
    {
      String pom[]=dane.split(" ");                        //rozebrania zawartości wiersza po pojedyncze słowa, oddzielone spacjami
      tablica.Wstaw(pom[1],Float.parseFloat(pom[3]),pom[5]); //zapis do tablicy jedynie danych z łańcucha
      dane=bufor.readLine();                               //odczytanie kolejnego wiersza
    }
    bufor.close();                                        //zamknięcie pliku i strumieni
  } catch (IOException e)
  { JOptionPane.showMessageDialog(null, "Bład odczytu pliku tekstowego "+e); }
}

```

Zawartość pliku w formacie tekstowym

The screenshot shows a Notepad window titled "Lister - [c:\Settings\dydaktyka\Komponenty\_1\bazy\Baza1\baza1.txt]". The menu bar includes "Plik", "Edytuj", "Opcje", and "Pomoc". The text content is as follows:

```

3
Nazwisko: 1 srednia: 1.0 uwagi: 1
Nazwisko: 2 srednia: 2.0 uwagi: 2
Nazwisko: 3 srednia: 3.0 uwagi: 3

```

A tooltip box is overlaid on the text, displaying the following information:

```

1 znak + '\n' = 2 bajty
33 znaki + '\n' = 34 bajty
33 znaki + '\n' = 34 bajty
33 znaki + '\n' = 34 bajty

```

```

static public void main(String args[])
{ Baza1 baza1=new Baza1(); //referencja do tablicy, ktora jest tworzona w opcji 1
  char ch;
  String s;
  do
  { s=JOptionPane.showInputDialog(null, "Progam nie jest zabezpieczony przed\n" +
    "zlym formatem danych\n i naciskaniem Cancel w okienkach dialogowych\n"+
    "Podaj wybor"
    +"\n1 - Wypelnij tablice,"
    +"\n2 - Wyswietl dane osob"
    +"\n3 - Wyszukaj osobe i wyswietl jej dane"
    +"\n4 - Zapisz plik w formacie tekstowym"
    +"\n5 - Odczytaj plik w formacie tekstowym i wyswietl na ekranie"
    +"\n6 - Odczytaj plik w formacie tekstowym i zapisz do tablicy"
    +"\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 '4' : baza1.Zapisz_do_pliku_tekstowego(); break;
    case '5' : baza1.Odczytaj_plik_tekstowy_na_ekran(); break;
    case '6' : baza1.Odczytaj_plik_tekstowy_do_tablicy(); break;
    case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu"); break;
    default : JOptionPane.showMessageDialog(null,"Zla opcja");
  }
}
while (ch != 'k') ;
System.exit(0); } }

```

### 3. Strumienie bajtowe plikowe

#### Procedura korzystania ze strumieni bajtowych buforowanych powiązanych z plikami binarymi

##### Aby utworzyć plik:

- 1) Należy utworzyć obiekt (np. typu *FileOutputStream*), powiązany ze plikiem danych binarnych (np. "plik2.dat");  
*FileOutputStream plik = new FileOutputStream("plik2.dat");*
- 2) W celu buforowania bajtów pochodzących z obiektu powiązanego ze źródłem danych np. typu *FileOutputStream* należy utworzyć obiekt klasy *BufferedOutputStream*  
*BufferedOutputStream bufor = new BufferedOutputStream( plik );*
- 3) Pojedyncze bajty mogą być zapisywane do pliku za pomocą metody bufora:  
*void write(int c)*
- 4) Całe ciągi bajtów mogą być zapisywane do pliku za pomocą metody bufora:  
*void write(byte[] cbuf, int off, int len)* – metoda, która czyta z tablicy *cbuf* od indeksu *off* liczbę *len* bajtów i zapisuje do pliku

##### Aby odczytać plik:

- 5) Należy utworzyć obiekt (np. typu *FileInputStream*), powiązany ze plikiem danych binarnych (np. "plik2.dat");  
*FileInputStream plik = new FileInputStream("plik2.dat");*
- 6) W celu buforowania bajtów pochodzących z obiektu powiązanego ze źródłem danych np. typu *FileInputStream* należy utworzyć obiekt klasy *BufferedInputStream*  
*BufferedInputStream bufor = new BufferedInputStream (plik);*
- 7) Pojedyncze bajty mogą być odczytywane metodą bufora:  
*int read ();*  
np. *int dane = plik.read()*
- 8) Całe ciągi bajtów mogą być odczytane za pomocą metody bufora:  
*int read (byte[] cbuf, int off, int len)* – metoda, która czyta plik i zapisuje do tablicy *cbuf* od indeksu *off* liczbę *len* bajtów i zwraca przez **return** liczbę faktycznie odczytanych bajtów
- 9) Po zapisie i odczycie bufor należy zamknąć metodą *close()*

```

package baza2;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import javax.swing.*;
import java.util.*;
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); }

```

```

public byte[] Zamien_na_bajty()

```

```

{ byte pom[];

```

```

  String S,S1;

```

```

  S1=Float.toString(srednia);

```

```

  S=" "+ " "+ " "+nazwisko+S1+uwagi;

```

```

  pom=S.getBytes();

```

```

  pom[0]=(byte)nazwisko.length();

```

```

  pom[1]=(byte)S1.length();

```

```

  pom[2]=(byte)uwagi.length();

```

```

  return pom; } }

```

```

//zamiana danej typu float na obiekt S1 typu String

```

```

//utworzenie łańcucha zawierającego trzy spacje oraz nazwisko, S1 i uwagi

```

```

//zamiana łańcucha na tablicę bajtów, gdzie każdy bajt reprezentuje znak

```

```

//zapis do pierwszego bajtu, pustego, liczbę bajtów zawierającej nazwisko

```

```

//zapis do drugiego bajtu, pustego, liczbę bajtów zawierającej srednia

```

```

//zapis do trzeciego bajtu, pustego, liczbę bajtów zawierającej uwagi

```

```

//zwrócenie tablicy bajtów przez metodę Zamien_na_bajty

```

Przykład prezentowany na wykładach z tablic (Przykład 6 – część 1

```

class Tablica //przykład prezentowany na wykładach z tablic (Przykład 6– część 1)
{
    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 Baza2

```
{ protected Tablica tablica=new Tablica();
```

```
public void Wyswietl()
```

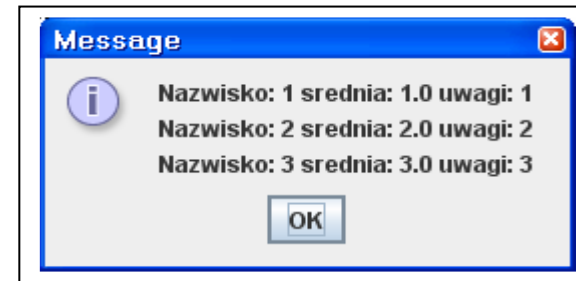
```
{ if (tablica.Pusta()) //jeśli tablica istnieje, to można wyświetlać
    JOptionPane.showMessageDialog(null, "Brak danych");
else
    JOptionPane.showMessageDialog(null, tablica.toString());
}
```

```
public void Wypelnij()
```

```
{ String S;
  S=JOptionPane.showInputDialog(null,"Podaj rozmiar tablicy");
  int N=Integer.parseInt(S);
  tablica.Wykonaj_tablice(N);
  String S1,S2,S3;
  while(!tablica.Pelna())
  {
    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);
  }
}
```

```
public void Wyszukaj()
```

```
{ if (tablica.Pusta())
    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());
}
}
```



```
//utworzenie tablicy
```

```
//jeśli tablica jest zapełniona, tak jest równe false
```

```
//dopisanie danych na końcu danych
```

```
//jeśli tablica istnieje, to można wyszukiwać
```

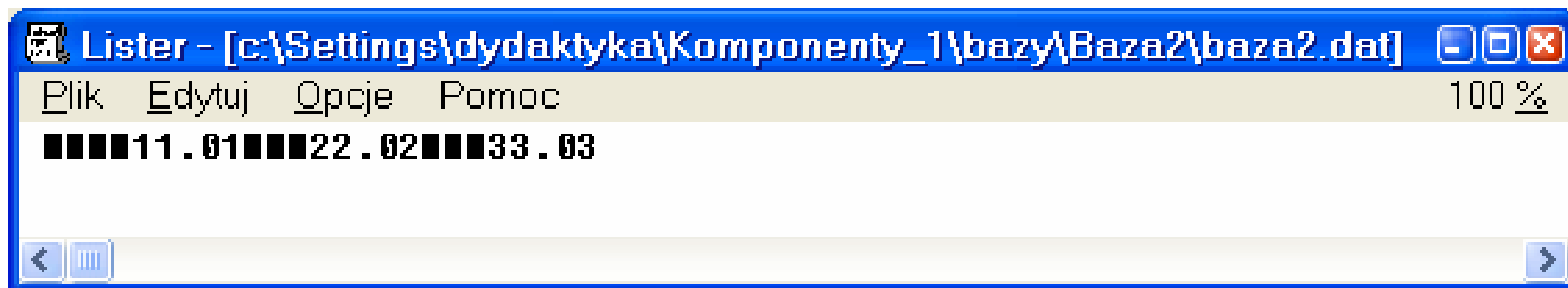
Przykład  
prezentowany  
na wykładach  
z tablic  
(Przykład 6 –  
część 1

```

public void Zapisz_do_pliku_binarnego()
{
    byte pom[];
    if (tablica.Pusta()) //jeśli tablica istnieje, to można zapisać do pliku
        JOptionPane.showMessageDialog(null,"Brak danych");
    else
    try
    {
        FileOutputStream plik = new FileOutputStream ("baza2.dat"); //utworzenie strumienia wyjściowego
        BufferedOutputStream bufor = new BufferedOutputStream (plik); //utworzenie bufora wyjściowego
        bufor.write(tablica.ile); //zapisanie do pliku liczby elementów
        for (int i=0; i<tablica.ile;i++)
        {
            pom=tablica.Dane[i].Zamien_na_bajty(); //utworzenie tablicy bajtów zawierającej nazwiska, średniej i imienia
            bufor.write(pom); //zapis tablicy bajtów do pliku, zawierającej dane jednego obiektu
        }
        bufor.close(); //zamknięcie pliku i strumieni
    } catch (IOException e)
        { JOptionPane.showMessageDialog(null, "Bład zapisu pliku binarnego" +e); }
}

```

## Zawartość pliku w formacie binarnym



Nr bajtu	1	2	3	4	5	6	7	8	9
Zawartość bajtu [B]	Ile	Rozmiar Nazwiska	Rozmiar średniej	Rozmiar uwag					
Przykład	3	1	3	1	1	1	.	0	1

Nr bajtu	10	11	12	13	14	15	16	17
Zawartość bajtu [B]	Rozmiar Nazwiska	Rozmiar średniej	Rozmiar uwag					
Przykład	1	3	1	2	2	.	0	2

Nr bajtu	18	19	20	21	22	23	24	25
Zawartość bajtu [B]	Rozmiar Nazwiska	Rozmiar średniej	Rozmiar uwag					
Przykład	1	3	1	3	3	.	0	3

```
public void Odczytaj_plik_binarny_na_ekran()
```

```
{ byte pom1[],pom2[],pom3[];
```

```
String S="",S1;
```

```
int rozmiar;
```

```
try
```

```
{
```

```
FileInputStream plik = new FileInputStream ("baza2.dat");
```

```
BufferedInputStream bufor = new BufferedInputStream (plik);
```

```
rozmiar=bufor.read();
```

```
rozmiar=bufor.read();
```

```
while(rozmiar!=-1)
```

```
{ pom1=new byte[rozmiar];
```

```
rozmiar=bufor.read();
```

```
pom2=new byte[rozmiar];
```

```
rozmiar=bufor.read();
```

```
pom3=new byte[rozmiar];
```

```
bufor.read(pom1,0,pom1.length);
```

```
bufor.read(pom2,0,pom2.length);
```

```
bufor.read(pom3,0,pom3.length);
```

```
S1=new String(pom1);
```

```
S1=new String(pom2);
```

```
S1=new String(pom3);
```

```
rozmiar=bufor.read(); }
```

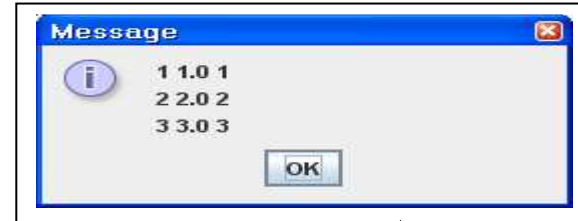
```
bufor.close();
```

```
JOptionPane.showMessageDialog(null, S); //wyswietlenie pliku w postaci łańcucha zawierającego w wierszach dane obiektów
```

```
} catch (IOException e)
```

```
{ JOptionPane.showMessageDialog(null,"Bład odczytu pliku binarnego "+e); }
```

```
}
```



//utworzenie strumienia wejściowego

//utworzenie bufora wejściowego

//odczytanie liczby obiektów

//odczytanie rozmiaru nazwiska

//tak długo czytaj, aż plik zostanie wyczerpany

//utworzenie tablicy pom1 bajtów do odczytu nazwisko

//odczytanie rozmiaru srednia

// utworzenie tablicy pom2 bajtów do odczytu srednia

// odczytanie rozmiaru uwagi

// utworzenie tablicy pom3 bajtów do odczytu uwagi

//odczytanie nazwisko do pom1

//odczytanie srednia do pom2

//odczytanie uwagi do pom3

//utworzenie łańcucha S zawierającego nazwisko

//dodanie do łańcucha srednia

//dodanie do łańcucha uwagi

//odczytanie rozmiaru nazwisko należącego do danych kolejnego obiektu

//zamknięcie pliku I strumieni

```

public void Odczytaj_plik_binarny_do_tablicy()
{ byte pom1[],pom2[],pom3[];
  String S1,S2,S3;
  int rozmiar;
  try
  { FileInputStream plik = new FileInputStream ("baza2.dat");
    BufferedInputStream bufor = new BufferedInputStream (plik); //utworzenie bufora wejściowego
    rozmiar=bufor.read(); //odczytanie liczby danych obiektów
    tablica.Wykonaj_tablice(rozmiar); // utworzenie nowej tablicy osób
    rozmiar=bufor.read(); //odczytanie rozmiaru nazwiska
    while(rozmiar!=-1) //tak długo czytaj, aż plik zostanie wyczerpany
    { pom1=new byte[rozmiar]; //utworzenie tablicy pom1 bajtów do odczytu nazwisko
      rozmiar=bufor.read(); //odczytanie rozmiaru srednia
      pom2=new byte[rozmiar]; // utworzenie tablicy pom2 bajtów do odczytu srednia
      rozmiar=bufor.read(); // odczytanie rozmiaru uwagi
      pom3=new byte[rozmiar]; // utworzenie tablicy pom3 bajtów do odczytu uwagi
      bufor.read(pom1,0,pom1.length); //odczytanie nazwisko do pom1
      bufor.read(pom2,0,pom2.length); //odczytanie srednia do pom2
      bufor.read(pom3,0,pom3.length); //odczytanie uwagi do pom3
      S1=new String(pom1); //utworzenie łańcucha S1 zawierającego nazwisko z danych tablicy bajtów
      S2=new String(pom2); //utworzenie łańcucha S2 zawierającego srednia z danych tablicy bajtów
      S3=new String(pom3); //utworzenie łańcucha S3 zawierającego uwagi z danych tablicy bajtów
      tablica.Wstaw(S1,Float.parseFloat(S2),S3); //wstawienie do tablicy nowej danej
      rozmiar=bufor.read(); } //odczytanie rozmiaru nazwiska następnego obiektu
    bufor.close(); //zamknięcie pliku i strumieni
  } catch (IOException e)
  { JOptionPane.showMessageDialog(null, "Bład odczytu pliku binarnego "+e); }
}

```

```

static public void main(String args[])
{
    Baza2 baza1=new Baza2(); //referencja do tablicy, ktora jest tworzona w opcji 1
    char ch;
    String s;
    do
    {
        s=JOptionPane.showInputDialog(null, "Progarn nie jest zabezpieczony przed\n" +
            "zlym formatem danych\n i naciskaniem Cancel w okienkach dialogowych\n"+
            "Podaj wybor"
            +"\n1 - Wypelnij tablice,"
            +"\n2 - Wyszwiatl dane osob"
            +"\n3 - Wyszukaj osobe i wyswiatl jej dane"
            +"\n4 - Zapisz plik w formacie binarnym"
            +"\n5 - Odczytaj plik w formacie binarnym i wyswiatl na ekranie"
            +"\n6 - Odczytaj plik w formacie binarnym i zapisz do tablicy"
            +"\nk - Koniec programu");
        ch = s.charAt(0); //pobranie opcji
        switch(ch)
        {
            case '1' : baza1.Wypelnij();
            case '2' : baza1.Wyszwiatl();
            case '3' : baza1.Wyszukaj();
            case '4' : baza1.Zapisz_do_pliku_binarnego();
            case '5' : baza1.Odczytaj_plik_binarny_na_ekran();
            case '6' : baza1.Odczytaj_plik_binarny_do_tablicy();
            case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu");
            default : JOptionPane.showMessageDialog(null,"Zla opcja");
        }
    }
    while (ch != 'k') ;
    System.exit(0); } }

```

```

break;
break;
break;
break;
break;
break;
break;

```

### 3. Strumienie obiektów

- Jest to mechanizm szeregowego zapisu do pliku związanego ze strumieniem wyjściowym ciągu bajtów po wykonaniu konwersji obiektu do postaci szeregowej i
- Odczytu szeregowego ciągu bajtów z pliku związanego ze strumieniem wejściowym i dokonanie konwersji do postaci danej (obektu, typu podstawowego: **int**, **float** itp.)
- Mechanizm ten pozwala zachować całe obiekty w pliku po zakończeniu programu
- Obiekty zapisywane do pliku muszą implementować pusty interfejs *Serializable* (obiekty są serializowane) – **implements** *Serializable* umieszczone po nazwie klasy, przez ciałem klasy
- Obiekty z zagnieżdżonymi obiektami są w całości zapisywane do pliku pod warunkiem, że zagnieżdżone obiekty też są serializowane
- Obiekty zagnieżdżone w serializowanych klasach mogą być pomijane przy zapisie do strumienia, jeśli to konieczne, za pomocą słowa kluczowego **transient**  
Np. **public transient** String s = "Kowalski";
- Identyfikator wersji serializowanego obiektu  
**private static final long** serialVersionUID = 1L;

# Procedura korzystania ze strumieni obiektowych powiązanych z plikami binarymi

## Aby utworzyć plik:

- 1) Należy utworzyć obiekt (np. typu *FileOutputStream*), powiązany ze plikiem danych binarnych (np. "Wiadomosc.obj");

```
FileOutputStream plikobiekto = new FileOutputStream("Wiadomosc.obj");
```

- 2) W celu utworzenia wyjściowego strumienia obiektowego powiązanego z obiektem związanym ze źródłem danych np. typu *FileOutputStream* należy utworzyć obiekt klasy *ObjectOutputStream*

```
ObjectOutputStream strumienobiekto =  
new ObjectOutputStream (plikobiekto);
```

- 3) obiekty dziedziczące po *Object* i implementujące interfejs *Serializable* są zapisywane do pliku w postaci szeregowej za pomocą metody

```
void writeObject(Object ob)
```



## Dalej podano część metod strumienia obiektów do zapisu różnych danych do pliku:

4) Pojedyncze bajty mogą być zapisywane do pliku za pomocą metody:

**void** write(**int** b)

5) Całe ciągi bajtów mogą być zapisywane do pliku za pomocą metody:

**void** write(byte[] cbuf) – metoda, która czyta zawartość tablicy bajtów i zapisuje ją do strumienia

6) Całe ciągi bajtów mogą być zapisywane do pliku za pomocą metody:

**void** write(**int**[] cbuf, **int** off, **int** len) – metoda, która czyta z tablicy *cbuf* od indeksu *off* liczbę *len* bajtów i zapisuje do pliku

7) **void** writeBoolean(**boolean** v) – zapisuje do pliku 1-bajtową wartość

8) **void** writeByte(**int** v) – zapisuje do pliku 1-bajtową wartość

9) **void** writeChar(**int** v) – zapisuje znak jako 2-bajtową wartość

10) **void** writeDouble(**double** v) – zapisuje 8-bajtową wartość do pliku

11) **void** writeFloat(**float** v) – zapisuje 4-bajtową wartość do pliku

12) **void** writeInt(**int** v) – zapisuje 4 bajty do pliku

13) **void** writeLong(**long** v) – zapisuje 8 bajtów do pliku

14) **void** writeShort(**int** v) – zapisuje 2 bajty do pliku

## ***Aby odczytać plik:***

1) Należy utworzyć obiekt (np. typu *FileInputStream*), powiązany ze plikiem danych binarnych (np. "Wiadomosc.obj");

```
FileInputStream plik = new FileInputStream("Wiadomosc.obj");
```

2) W celu odczytu obiektów pochodzących z obiektu powiązanego ze źródłem danych np. typu *FileInputStream* należy utworzyć obiekt klasy *ObjectInputStream*

```
ObjectInputStream bufor = new ObjectInputStream (plik);
```

3) Odczytu obiektów z strumienia należy wykonać za pomocą metody `Object readObject()`

## **Dalej podano niektóre metody strumienia obiektów do odczytu danych z pliku:**

- 4) **boolean** readBoolean() – czyta z pliku 1 bajt i wraca wartość true lub false
- 5) **byte** readByte() – czyta z pliku 1 bajt i zwraca wartość typu byte
- 6) **char** readChar() – czyta 1 znak (2 bajty ) i zwraca 1 znak
- 7) **double** readDouble() – czyta 8 bajtów z pliku i zwraca wartość **double**
- 8) **float** readFloat() – czyta 4 bajtów z pliku i zwraca wartość **float**
- 9) **int** readInt() – czyta 4 bajty z pliku i zwraca wartość typu **int**
- 10) **long** readLong() – czyta 8 bajtów z pliku i zwraca wartość typu **long**
- 11) **short** readShort() – czyta 2 bajty z pliku i zwraca wartość typu **short**
- 12) Po zapisie i odczycie strumień obiektów należy zamknąć metodą *close()*

**package** baza4;

//przykład prezentowany na wykładach z tablic (Przykład 6)

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import javax.swing.*;
import java.util.*;
```

```
class Osoba implements Serializable //przygotowanie do serializacji
```

```
{ 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 implements Serializable //przygotowanie do serializacji - przykład prezentowany na wykładach z tablic (Przykład 6)
{
    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 Baza4

```
{ protected Tablica tablica=new Tablica();
```

```
public void Wyswietl()
```

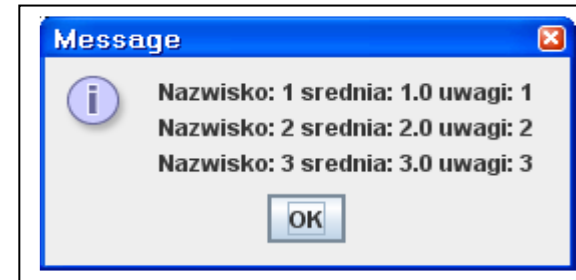
```
{ if (tablica.Pusta()) //jeśli tablica istnieje, to można wyświetlać
    JOptionPane.showMessageDialog(null, "Brak danych");
else
    JOptionPane.showMessageDialog(null, tablica.toString());
}
```

```
public void Wypelnij()
```

```
{ String S;
  S=JOptionPane.showInputDialog(null,"Podaj rozmiar tablicy");
  int N=Integer.parseInt(S);
  tablica.Wykonaj_tablice(N);
  String S1,S2,S3;
  while(!tablica.Pelna())
  {
    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);
  }
}
```

```
public void Wyszukaj()
```

```
{ if (tablica.Pusta())
    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());
}
}
```



//utworzenie tablicy

//jeśli tablica jest zapełniona, tak jest równe false

//dopisanie danych na końcu danych

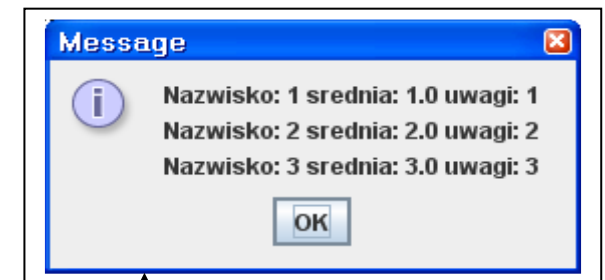
//jeśli tablica istnieje, to można wyszukiwać

Przykład  
prezentowany  
na wykładach  
z tablic  
(Przykład 6 –  
część 1

```

public void Zapisz_do_pliku_obiektow()
{
    if (tablica.Pusta())
        JOptionPane.showMessageDialog(null,"Brak danych");
    else
    try
    {
        FileOutputStream plik = new FileOutputStream ("baza3.obj"); //otwarcie strumienia wyjściowego
        ObjectOutputStream strumienobiektow = new ObjectOutputStream (plik); // warstwa serializująca obiekty
        strumienobiektow.writeObject(tablica); //zapis obiektu do pliku
        strumienobiektow.close(); //zamknięcie pliku i strumieni
    }
    catch (IOException e)
    {
        JOptionPane.showMessageDialog(null, "Bład zapisu pliku obiektowego" +e); }
    }
}

```



```

public void Odczytaj_plik_obiektow_na_ekran()
{
    try
    {
        FileInputStream plik = new FileInputStream ("baza3.obj"); //otwarcie strumienia wyjściowego
        ObjectInputStream strumienobiektow = new ObjectInputStream (plik); //otwarcie bufora wyjściowego
        Tablica tablica_=(Tablica)strumienobiektow.readObject();//odczyt obiektu ze strumienia jako obiektu lokalnego
        JOptionPane.showMessageDialog(null,tablica_.toString()); //wyświetlenie obiektu na ekran
        strumienobiektow.close(); //zamknięcie pliku i strumieni
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, "Bład odczytu pliku obiektowego "+e); }
    }
}

```

```

public void Odczytaj_plik_objektow_do_tablicy()
{
    try
    {
        FileInputStream plik = new FileInputStream ("baza3.obj"); //otwarcie strumienia wyjściowego
        ObjectInputStream strumienobjektow = new ObjectInputStream (plik); //otwarcie bufora wyjściowego
        tablica=(Tablica)strumienobjektow.readObject(); //odczyt obiektu ze strumienia jako składowej obiektu Baza4
        strumienobjektow.close(); //zamknięcie pliku i strumieni
    } catch (Exception e)
        { JOptionPane.showMessageDialog(null, "Bład odczytu pliku obiektowego "+e); }
    }
}

```

Zawartość pliku obiektów

Notepad window titled "Lister - [c:\Settings\dydaktyka\Komponenty\_1\bazy\Baza4\baza3.obj]". The content is a Java object representation of a Tablica object. The text is as follows:

```

-ísrbaza4.Tablica>Z#a1&9Iile[Danet[Lbaza4/Osoba;xpur[Lt
aza4.Osoba;\IzfZĆxp sr baza4.Osoba%úKÍDkřF sredniaL nazwisko
tL java/lang/String;L uwagiq~ xp? t 1q~ sq~ @ t 2q~ sq~ @
@ t 3q~

```



```

static public void main(String args[])
{
    Baza4 baza1=new Baza4();           //referencja do tablicy, ktora jest tworzona w opcji 1
    char ch;
    String s;
    do
    {
        s=JOptionPane.showInputDialog(null, "Progama nie jest zabezpieczony przed\n" +
            "zlym formatem danych\n i naciskaniem Cancel w okienkach dialogowych\n"+
            "Podaj wybor"
            +"\n1 - Wypelnij tablice,"
            +"\n2 - Wyszwiatl dane osob"
            +"\n3 - Wyszukaj osobe i wyswiatl jej dane"
            +"\n4 - Zapisz plik obiektow"
            +"\n5 - Odczytaj plik obiektow i wyswiatl na ekranie"
            +"\n6 - Odczytaj plik obiektow i zapisz do tablicy"
            +"\nk - Koniec programu");
        ch = s.charAt(0);                //pobranie opcji
        switch(ch)
        {
            case '1' : baza1.Wypelnij();           break;
            case '2' : baza1.Wyszwiatl();          break;
            case '3' : baza1.Wyszukaj();           break;
            case '4' : baza1.Zapisz_do_pliku_obiektow(); break;
            case '5' : baza1.Odczytaj_plik_obiektow_na_ekran(); break;
            case '6' : baza1.Odczytaj_plik_obiektow_do_tablicy(); break;
            case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu"); break;
            default : JOptionPane.showMessageDialog(null,"Zla opcja");
        }
    }
while (ch != 'k') ;
System.exit(0); }}

```