



JAVA

Pliki - zapis i odczyt

Opracował: Andrzej Nowak

Bibliografia: **JAVA Szkoła programowania**, D. Trajkowska
Ćwiczenia praktyczne **JAVA**. Wydanie III, M. Lis



Strumienie wejścia i wyjścia

Strumienie wejścia w Javie pozwalają na odczyt danych z pliku. Służą do tego klasy ułożone hierarchicznie, pozwalające na operowanie na strumieniach danych.

W Javie obiektem, z którego można pobrać sekwencję bajtów jest strumień wejścia.

Klasa **InputStream** jest podstawową klasą abstrakcyjną, definiującą metody odczytu bajtów ze strumienia.

Klasa **Reader** jest podstawową klasą abstrakcyjną, definiującą metody odczytu ciągu znaków z obsługą Unicode.

Klasy pochodne od klas **InputStream** i **Reader** dostarczają metod pozwalających na pobieranie danych z wejścia, a więc odczyt danych

Klasy odczytu danych

Nazwa klasy strumienia wejściowego	Rodzaj pobieranych danych ze strumienia wejściowego
FileInputStream	Pobiera dane bajtowe z podanego pliku
BufferedReader	Pobiera łańcuch znaków
DataInputStream	Pobiera dane typów prostych oraz łańcucha znakowego
ObjectInputStream	Pobiera obiekt
ZipInputStream	Pobiera dane bajtowe z podanego archiwum ZIP

Strumienie wyjścia – obiekt, do którego można wysłać sekwencję bajtów.

Klasa **OutputStream** jest podstawową klasą abstrakcyjną, definiującą metody zapisu bajtów do strumienia.

Klasa **Writer** jest podstawową klasą abstrakcyjną, definiującą metody zapisu ciągu znaków z obsługą Unicode.

Klasy pochodne od klas **OutputStream** i **Writer** dostarczają metod pozwalających na zapisywanie danych na wyjście.

Każda z klas jest przeznaczona do zapisu określonego typu danych.

Strumienie wyjściowe są tak samo skonstruowane jak strumienie wejściowe, a więc mogą przekazywać do siebie strumienie w postaci obiektów, czyli stosować filtrowanie. Dzięki filtrowaniu możemy zapisywać na wyjście nie tylko bajty lub znaki, ale także obiekty czy typy proste.

Klasy zapisu danych

Nazwa klasy strumienia wejściowego	Rodzaj pobieranych danych ze strumienia wejściowego
FileOutputStream	Zapisuje dane bajtowe z podanego pliku
BufferedWriter	Zapisuje łańcuch znaków
DataOutputStream	Zapisuje dane typów prostych oraz łańcucha znakowego
ObjectOutputStream	Zapisuje obiekt
ZipOutputStream	Zapisuje dane bajtowe z podanego archiwum ZIP



Odczyt tekstu z pliku

Aby odczytać dane wystarczy zastosować **strumienie wejściowe**.

zadanie

Napisz aplikację odczytującą dane tekstowe z pliku tekstowego.

/zadanie5.txt/

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class Odczyt {
    public static void main (String args[]){
        Okno okno = new Okno();
        okno.setSize(280,400);
        okno.setTitle("Odczyt z pliku");
        okno.setDefaultCloseOperation(3);
        okno.setLocationRelativeTo(null);
        okno.setVisible(true);
    }
}

class Okno extends JFrame implements ActionListener{
    JTextField pole;
    JTextArea area;

    public Okno() {
        JPanel panel = new JPanel();
        panel.setBackground(new Color(125,125,0));
        JLabel etykieta1 = new JLabel("Nazwa pliku: ");
        pole = new JTextField(15);
        panel.add(etykieta1);
        panel.add(pole);
        JLabel etykieta2 = new JLabel("Tresc");
        panel.add(etykieta2);
```

```
        area = new JTextArea(16,20);
        panel.add(area);
        JButton przycisk = new JButton("Odczytaj");
        przycisk.addActionListener(this);
        panel.add(przycisk);
        getContentPane().add(panel);
    }

    public void actionPerformed(ActionEvent e) {
        String nazwa = pole.getText();
        String linia;
        try{
            BufferedReader br = new BufferedReader(new FileReader(nazwa));
            while ((linia = br.readLine()) != null){
                area.append(linia+"\n");
            }
            br.close();
        }catch(Exception ek){};
    }
}
```



Zapis tekstu do pliku

Aby odczytać dane wystarczy zastosować **strumień wyjściowe**.

zadanie

Napisz aplikację zapisującą dane tekstowe do pliku tekstowego.

/zadanie6.txt/

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class Zapis {
    public static void main (String args[]){
        Okno okno = new Okno();
        okno.setSize(280,400);
        okno.setTitle("Zapis do pliku");
        okno.setDefaultCloseOperation(3);
        okno.setLocationRelativeTo(null);
        okno.setVisible(true);
    }
}

class Okno extends JFrame implements ActionListener{
    JTextField pole;
    JTextArea area;

    public Okno() {
        JPanel panel = new JPanel();
        panel.setBackground(new Color(125,125,125));
        JLabel etykieta1 = new JLabel("Nazwa pliku: ");
        pole = new JTextField(15);
        panel.add(etykieta1);
        panel.add(pole);
        JLabel etykieta2 = new JLabel("Tresc");
        panel.add(etykieta2);
        area = new JTextArea(16,20);
        panel.add(area);
        JButton przycisk = new JButton("Zapisz");
        przycisk.addActionListener(this);
        panel.add(przycisk);
        getContentPane().add(panel);
    }

    public void actionPerformed(ActionEvent e) {
        String nazwa = pole.getText();
        String text = area.getText();
        try{
            BufferedWriter bw = new BufferedWriter(new FileWriter(nazwa));
```

```
    bw.write(text,0,text.length());  
    bw.flush();  
    bw.close();  
}catch(Exception ek){};  
}  
}
```