```java
package pl.nalazek.komunikator.logika.logowanie;

import java.io.*;

import pl.nalazek.komunikator.Program;
import pl.nalazek.komunikator.logika.Sterowanie;

/**
 * @author Daniel Nalazek
 * Copyright (C) 2014 Daniel Nalazek
 */

/** Klasa odpowiadająca za komponent Logowanie */
public class Logowanie {
    private File plik;
    private static Uzytkownicy uzytkownicy;
    private static UzytkownikLokalny zalogowanyUzytkownikLokalny;
    private Sterowanie s;

    /** Konstuktor
     * @param s Referencja do Sterowania w programie
     */
    public Logowanie(Sterowanie s)
    {
        this.s = s;
        plik = new File("uzytkownicy.dat");
        if(plik.exists())
        {
            try
            {
                ObjectInputStream ois = new ObjectInputStream(new FileInputStream(plik));
                uzytkownicy = (Uzytkownicy) ois.readObject();
                ois.close();
                uzytkownicy.wczytajUzytkownikow();
            }
            catch(Exception e)
            {
                Program.komponentSterowanie().odbierzWyjatek(e.getMessage());
            }
        }
        else
        {
            uzytkownicy = new Uzytkownicy();
        }
    }

    /** Funkcja weryfikuje poprawność danych użytkownika lokalnego i dokonuje logowania
     * @param nazwa Nazwa użytkownika lokalnego
     * @param haslo Hasło użytkownika lokalnego
     * @return UżytkownikLokalny w przypadku poprawnej weryfikacji, w przeciwnym razie "null"
     */
    public UzytkownikLokalny zaloguj(String nazwa, String haslo)
    {
        zalogowanyUzytkownikLokalny = uzytkownicy.zwrocUzytkownikaLokalnego(haslo+"/"+nazwa);
        if(zalogowanyUzytkownikLokalny != null)
        {
            Program.komponentModel(s).zalogowano(zalogowanyUzytkownikLokalny,
zalogowanyUzytkownikLokalny.zwrocListeKontaktow());
            return zalogowanyUzytkownikLokalny;
        }
        else return null;
    }

    /** Funkcja wylogowywuje obecnie zalogowanego użytkownika lokalnego */
    public void wyloguj()
    {
        if(zalogowanyUzytkownikLokalny != null)
        {
            zalogowanyUzytkownikLokalny = null;
            Program.komponentModel(s).wylogowano();
            zapiszStan();
        }
        else zapiszStan();
    }

    /** Funkcja zapisuje dane logowania, w tym również kontakty na dysk */
    private void zapiszStan()
    {
```

```
78          uzytkownicy.zapiszUzytkownikow();
79          try
80          {
81              ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(plik));
82              oos.writeObject(uzytkownicy);
83              oos.close();
84          }
85          catch(Exception e)
86          {
87              Program.komponentSterowanie().odbierzWyjatek(e.getMessage());
88          }
89      }
90
91
92  }
93
```