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

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

import java.util.*;
import java.io.*;
import java.net.InetSocketAddress;
import pl.nalazek.komunikator.logika.logowanie.UzytkownikZdalny;

/** Interfejs obsługujący Model aplikacji*/
public interface IModel {

    /** Zarządza widzialnością użytkownika w sieci
     * @param wartosc Wartość "prawda" umożliwia nawiązanie połącznia z programem z zewnątrz. Wartość "fałsz" wyłącza widoczność w sieci.
     * @param localhost Parametr określający tryb pracy online.
     * */
    void ustawOnline(boolean wartosc, LocalHost localhost) throws IOException, ClassNotFoundException, Wyjatek;

    /** Dodaje uzytkownika zdalnego do listy uzytkownikow obecnie zalogowanego uzytkownika lokalnego.
     * @param nazwaUzytkownika Nazwa (alias) dodawanego uzytkownika.
     * @param adresSocket Gniazdo użytkownika zdalnego
     * */
    UzytkownikZdalny dodajUzytkownikaZdalnego(String nazwaUzytkownika, InetSocketAddress adresSocket) throws Wyjatek;

    /** Usuwa uzytkownika zdalnego z listy uzytkownikow obecnie zalogowanego uzytkownika lokalnego.
     * @param nazwaUzytkownika Nazwa (alias) dodawanego uzytkownika.
     * @param adres Adres IP lub ID z zewnętrznego serwera WWW/SQL.
     * */
    void usunUzytkownikaZdalnego(String nazwaUzytkownika, String adres) throws Wyjatek;

    /** Zwraca adres IP komputera na którym uruchomiony jest program
     * @return Adres IP w postaci ciągu znaków w formacie: xxx.xxx.xxx.xxx
     * */
    String mojeIP();

    /** Wysyła żądanie nawiązania rozmowy ze zdalnym użytkownikiem
     * @param uzytkownik Referencja do uzytkownika zdalnego
     * @return Referencja do rozmowy w przypadku zgody, referencja do rozmowy z id=0 w przypadku odmowy, "null" w przypadku braku odpowiedzi.*/
    RozmowaID rozpocznijRozmowe(UzytkownikZdalny uzytkownik) throws Wyjatek;

    /** Zwraca zbiór użytkowników zdalnych dla obecnie zalogowanego użytkownika lokalnego.
     * @return Lista użytkowników zdalnych w postaci zbioru LinkedHashSet<String[]>: [0]:nazwa użytkownika zdalnego, [1]:ip, [2]:nr użytkownika w systemie */
    LinkedHashSet<String[]> listaUZ();

    /** Wysyła żądanie odebrania pliku przez użytkownika zdalnego
     * @param uZdalny Adresat
     * @param id Referencja do rozmowy
     * @param sciezka_pliku Scieżka wysyłanego pliku
     * @return Obiekt reprezentujący stan wysyłania pliku */
    StanWyslaniaPliku wyslijPlik(UzytkownikZdalny uZdalny, RozmowaID id, String sciezka_pliku) throws Wyjatek;

    /** Wysyła wiadomość do użytkownika
     * @param id Referencja do rozmowy z której wysyłana jest wiadomość
     * @param wiadomosc Treść wiadomości
     * */
    void wyslijWiadomosc(RozmowaID id, String wiadomosc) throws IOException;

    /** Kończy rozmowę
     * @param id Referencja do rozmowy
     */
    void zakonczRozmowe(RozmowaID id) throws IOException;

}
```