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

import java.io.*;
import java.net.Socket;
import pl.nalazek.komunikator.Program;
import pl.nalazek.komunikator.logika.logowanie.UzytkownikZdalny;

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

/** Watek tworzący połączenie do wysylania na określonym porcie */
public class WatekWysylajacy extends Watek {

    private Socket gniazdoWysylajace = null;
    private int port;
    private ObjectOutputStream strumienWych;
    private Rozmowa rozmowa;
    private UzytkownikZdalny uZdalny;
    private Pakiet pakiet;
    private String adresIp;
    /** Semafor obsługujący dostęp do strumienia wysyłającego. Umożliwia użycie na nim metod wait() i notify(). */
    Object semafor;

    /** Konstruktor wątku wysyłającego dla użytkownika zdalnego
     * @param rozmowa Referencja do rozmowy, z której będzie obsługiwany ten wątek
     * @param uZdalny Referencja do użytkownika zdalnego, który ma zostać podłączony strumień wychodzący w tym wątku
     * @param port Port użytkownika zdalnego do którego ma zostać podłączony strumień wychodzący w tym wątku
     */
    public WatekWysylajacy(Rozmowa rozmowa, UzytkownikZdalny uZdalny, int port)
    {
        semafor = new Object();
        this.port = port;
        this.uZdalny = uZdalny;
        this.rozmowa = rozmowa;
    }

    /** Konstruktor wątku wysyłającego dla pojedynczego pakietu
     * @param pakiet Pakiet do wysłania
     * @param adresIp Adres IP komputera docelowego
     * @param port Port komputera docelowego
     */
    public WatekWysylajacy(Pakiet pakiet, String adresIp, int port)
    {
        semafor = new Object();
        this.port = port;
        this.adresIp = adresIp;
        this.pakiet = pakiet;
    }

    /**  Główna metoda wątku */
    public void run()
    {
        try{

            if(uZdalny != null)
            {
                String ip = (uZdalny.zwrocIp().equals("localhost"))
                        || (uZdalny.zwrocIp().equals("127.0.0.1")) ?
                        null : uZdalny.zwrocIp();
                gniazdoWysylajace = new Socket(ip , port);
            }
            else if(adresIp != null)
                gniazdoWysylajace = new Socket(adresIp=="localhost" ? null : adresIp, port);
            else
                interrupt();
            BufferedOutputStream bos = new BufferedOutputStream(gniazdoWysylajace.getOutputStream());
            strumienWych = new ObjectOutputStream(bos);
            if(uZdalny != null)
            {
                synchronized(semafor) {

                    while(!interrupted())
                    {

                        semafor.wait();

```

```
79                          Pakiet a =  rozmowa.obslugaKolejki(uZdalny,null);
80                          strumienWych.writeObject(a);
81                          strumienWych.flush();
82                          strumienWych.reset();
83                      }
84                  }
85              }
86              else if(adresIp != null)
87              {
88                  strumienWych.writeObject(pakiet);
89                  strumienWych.flush();
90              }
91              gniazdoWysylajace.close();
92              strumienWych.close();
93          }
94          catch(IOException w)
95          {
96              if(w.getMessage().contains("refused")) ;
97              else Program.komponentSterowanie().odbierzWyjatek("W wątku: " + this.getName() + " wystąpił wyjątek.\n" +
    w.getMessage() + ";;" + w.getCause());
98          }
99          catch(InterruptedException w)
100         {
101         }
102         catch(Throwable w)
103         {
104             w.getMessage();
105         }
106         finally
107         {
108             try
109             {
110                 if(gniazdoWysylajace != null) gniazdoWysylajace.close() ;
111                 if(strumienWych != null) strumienWych.close();
112             }
113             catch (IOException i)
114             {
115                 i.getMessage();
116                 }
117         }
118
119     }
120 }
121
```