

```

1 package pl.nalazek.komunikator.logika;
2
3 import java.util.*;
4
5 /**
6  * @author Daniel Nalazek
7  * Copyright (C) 2014 Daniel Nalazek
8  */
9
10 /** Klasa obsługująca wątki w programie */
11 public class Watek extends Thread{
12
13     /** Referencja do modelu, dla obsługi pakietów */
14     protected static Model modelRef;
15     /** Lista przechowująca wszystkie aktywne wątki */
16     static ArrayList<Watek> listaWatkow = null;
17
18     /** Konstruktor */
19     public Watek()
20     {
21         if(listaWatkow == null) { listaWatkow = new ArrayList<Watek>(); }
22         listaWatkow.add(Watek.this);
23     }
24
25     /** Rozszerzona metoda, oddziedziczona z klasy Thread. Dodatkowo usuwa aktualny wątek z listy wątków */
26     public void interrupt()
27     {
28         super.interrupt();
29         listaWatkow.remove(Watek.this);
30     }
31
32     /** Zabija wątek */
33     public void zabijWatek()
34     {
35         interrupt();
36     }
37
38     /** Zabija wszystkie stworzone wątki */
39     public void zabijWszystkieWatki()
40     {
41         Iterator<Watek> itertr = listaWatkow.iterator();
42         while(itertr.hasNext())
43         {
44             itertr.next().interrupt();
45         }
46     }
47
48     /** Ustawia referencję do modelu programu */
49     static void ustawRefModelu(Model m)
50     {
51         modelRef = m;
52     }
53
54     /** Sprawdza czy dany wątek jeszcze istnieje */
55     public static boolean sprawdzCzyIstniejeWatek(Watek w)
56     {
57         if(listaWatkow.contains(w)) return true;
58         else return false;
59     }
60 }
61

```