

```

1 package pl.nalazek.komunikator;
2
3 import pl.nalazek.komunikator.gui.Gui;
4 import pl.nalazek.komunikator.logika.Model;
5 import pl.nalazek.komunikator.logika.Sterowanie;
6 import pl.nalazek.komunikator.logika.Watek;
7 import pl.nalazek.komunikator.logika.WatekSluchajacy;
8
9 /**
10  * @author Daniel Nalazek
11  * Copyright (C) 2014 Daniel Nalazek
12  */
13
14 /** Klasa typu Singleton, przechowująca referencję trzech podstawowych komponentów programu (MVC) */
15 public class Program {
16
17     private static Model model ;
18     private static Gui gui;
19     private static Sterowanie sterowanie;
20     private static Program referencja;
21
22     protected Program()
23     {}
24
25     /** Konstruktor klasy
26      * @param gui Referencja do interfejsu graficznego w postaci klasy Gui
27      * @param model Referencja do modelu programu w postaci klasy Model
28      * @param sterowanie Referencja do sterowania programu w postaci klasy Sterowanie
29      * @return Referencja do programu w postaci klasy program
30      */
31     public static Program utworz(Gui gui, Model model, Sterowanie sterowanie)
32     {
33         if(referencja == null)
34         {
35             referencja = new Program();
36         }
37         Program.model = model;
38         Program.gui = gui;
39         Program.sterowanie = sterowanie;
40         return referencja;
41     }
42
43     /** Metoda umożliwiająca dostęp do metod zawartych w modelu programu,
44      * dostępna tylko dla unikalnego egzemplarza klasy Sterowanie, przekazanego wcześniej w postaci referencji
45      * w konstruktorze klasy Program.
46      * @param s Referencja do sterowania programu
47      * @return Referencja do modelu programu lub "null" w przypadku niepoprawnej weryfikacji
48      */
49     public static Model komponentModel(Sterowanie s)
50     {
51         if(s!=null && s == sterowanie)
52             return model;
53         else return null;
54     }
55
56     /** Metoda umożliwiająca dostęp do metod zawartych w interfejsie graficznym programu,
57      * dostępna tylko dla unikalnego egzemplarza klasy Sterowanie, przekazanego wcześniej
58      * w postaci referencji w konstruktorze klasy Program.
59      * @param s Referencja do sterowania programu
60      * @return Referencja do interfejsu graficznego programu lub "null" w przypadku niepoprawnej weryfikacji
61      */
62     public static Gui komponentGui(Sterowanie s)
63     {
64         if(s!=null && s == sterowanie)
65             return gui;
66         else return null;
67     }
68
69     /** Metoda umożliwiająca dostęp do metod zawartych w interfejsie graficznym programu,
70      * dostępna tylko dla wątków słuchających w programie
71      * @param s Referencja do klasy WatekSluchajacy
72      * @return Referencja do interfejsu graficznego programu lub "null" w przypadku niepoprawnej weryfikacji
73      */
74     public static Gui komponentGui(WatekSluchajacy s)
75     {
76         if(s!=null && (Watek.sprawdzyCzyIstniejeWatek(s)))
77             return gui;
78         else return null;

```

```
79     }
80
81     /** Metoda umożliwiająca dostęp do metod zawartych w sterowaniu, dostępna dla Modelu i Gui
82      * @return Referencja do sterowania
83      */
84     public static Sterowanie komponentSterowanie()
85     {
86         return sterowanie;
87     }
88
89     /** Metoda uruchamiająca sterowanie programu
90      * Musi zostać uruchomiona po wywołaniu konstruktora */
91     public void uruchom()
92     {
93         sterowanie.sterowanieStart();
94     }
95
96
97 }
98
```