

```
1 using System.Diagnostics;
2 using Shutdown;
3
4 namespace Shutdown_8
5 {
6     public partial class Form1 : Form
7     {
8         private System.Timers.Timer countdownTimer;
9         private int countdownSeconds;
10
11         public Form1()
12         {
13             InitializeComponent();
14         }
15
16         private void formatRadioButton_CheckedChanged(object sender, ↗
17             EventArgs e)
18         {
19             minutesTextBox.Enabled = false;
20             hourTextBox.Enabled = false;
21             timeTextBox.Enabled = true;
22         }
23
24         private void selectedHourRadioButton_CheckedChanged(object ↗
25             sender, EventArgs e)
26         {
27             timeTextBox.Enabled = false;
28             minutesTextBox.Enabled = true;
29             hourTextBox.Enabled = true;
30         }
31
32         private void startButton_Click(object sender, EventArgs e)
33         {
34             try
35             {
36                 InitializeCountdownTimer();
37
38                 if (selectedHourRadioButton.Checked)
39                 {
40                     int targetHour = int.Parse(hourTextBox.Text);
41                     int targetMinute = int.Parse(minutesTextBox.Text);
42
43                     DateTime now = DateTime.Now;
44                     DateTime targetTime = new DateTime(now.Year, ↗
45                         now.Month, now.Day, targetHour, targetMinute, 0);
46
47                     if (targetTime <= now)
48                     {
49                         targetTime = targetTime.AddDays(1);
50                     }
51                 }
52             }
53             catch { }
54         }
55     }
56 }
```

```
51         countdownSeconds = (int)(targetTime -  
            now).TotalSeconds;  
52  
53  
54     }  
55     else if (formatRadioButton.Checked)  
56     {  
57         countdownSeconds = ParseTimeInSeconds();  
58     }  
59  
60     countdownTimer.Start();  
61 }  
62  
63 catch (FormatException)  
64 {  
65     MessageBox.Show("Wprowadź prawidłowo sformatowany  
        czas.", "Błąd", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning);  
66 }  
67 }  
68  
69 private void stopButton_Click(object sender, EventArgs e)  
70 {  
71     countdownTimer.Stop();  
72     UpdateStatusLabel("Odliczanie zatrzymane.");  
73 }  
74 private void CountdownTimer_Tick(object? sender, EventArgs e)  
75 {  
76     string mode = "";  
77  
78     if (shutdownRadioButton.Checked)  
79     {  
80         mode = "Zamknięcie systemu ";  
81     }  
82     else if (hibernateRadioButton.Checked)  
83     {  
84         mode = "Hibernacja systemu";  
85     }  
86     else if (netflixRadioButton.Checked)  
87     {  
88         mode = "Zamknięcie aplikacji Netflix";  
89     }  
90  
91     if (countdownSeconds > 0)  
92     {  
93         UpdateStatusLabel($"{mode} nastąpi za: \n  
            {countdownSeconds} sekund");  
94         countdownSeconds--;  
95     }  
96     else  
97     {  
98         if (shutdownRadioButton.Checked)  
99         {
```

```
100         ShutdownComputer();
101         UpdateStatusLabel("Akcja wykonana. \n Komputer zostanie wyłączony");
102     }
103     else if (hibernateRadioButton.Checked)
104     {
105         HibernateComputer();
106         UpdateStatusLabel("Akcja wykonana. \n Komputer zostanie zahibernowany");
107     }
108     else if (netflixRadioButton.Checked)
109     {
110         ShutdownNetflix();
111         UpdateStatusLabel("Akcja wykonana. \n Aplikacja Netflix zostanie zamknięta.");
112     }
113
114     else
115     {
116         UpdateStatusLabel("Żadna akcja nie została wybrana.");
117     }
118
119     countdownTimer.Stop();
120 }
121 }
122 private void InitializeCountdownTimer()
123 {
124     if (countdownTimer == null)
125     {
126         countdownTimer = new System.Timers.Timer(1000);
127         countdownTimer.Elapsed += CountdownTimer_Tick;
128     }
129 }
130
131 private bool IsValidTimeFormat(string time)
132 {
133     return !string.IsNullOrEmpty(time);
134 }
135
136 private int ParseTimeInSeconds()
137 {
138     if (formatRadioButton.Checked)
139     {
140         string timeText = timeTextBox.Text;
141
142         // Sprawdź, czy tekst zawiera tylko jedną cyfrę i dodaj "0" z przodu
143         if (timeText.Length == 1)
144         {
145             timeText = "00:00:0" + timeText;
146         }
147         else if (timeText.Length == 2)
```

```
148         {
149             timeText = "00:00:" + timeText;
150         }
151
152         if (TimeSpan.TryParse(timeText, out TimeSpan timeSpan))
153         {
154             return (int)timeSpan.TotalSeconds;
155         }
156     }
157     else
158     {
159         int minutes = int.Parse(minutesTextBox.Text);
160         int hours = int.Parse(hourTextBox.Text);
161         return (hours * 60 + minutes) * 60;
162     }
163
164     return 0;
165 }
166
167 private void ShutdownComputer()
168 {
169     Process.Start("shutdown", "/s /f /t 1");
170 }
171
172 private void HibernateComputer()
173 {
174     Process.Start("shutdown", "/h");
175 }
176
177 private void ShutdownNetflix()
178 {
179     Process.Start("taskkill", "/IM ApplicationFrameHost.exe");
180 }
181
182 private void UpdateStatusLabel(string message)
183 {
184     if (statusLabel.InvokeRequired)
185     {
186         // Jeśli wywołanie pochodzi z innego wątku, użyj Invoke, aby przenieść się na wątek UI.
187         statusLabel.Invoke(new Action(() => statusLabel.Text = message));
188     }
189     else
190     {
191         // Jeśli jesteśmy już na wątku UI, możemy bezpośrednio aktualizować kontrolkę.
192         statusLabel.Text = message;
193     }
194 }
195 }
196 }
197
```