Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №8**

**«ОСНОВЫ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПРОГРМАММИРОВАНИЯ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк 101-51-00

Орлов Валерий Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы:

Получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

1. Скриншот задания (с вариантом)

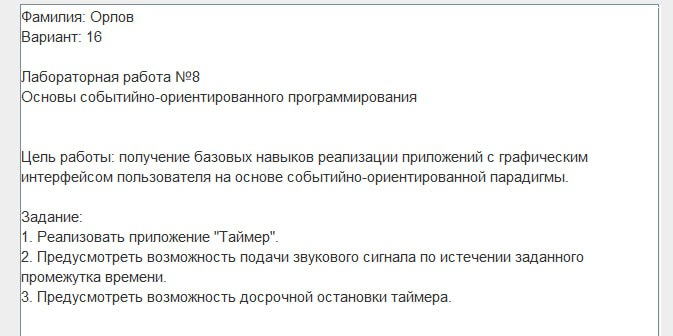


Рисунок 1 – Скриншот задания

1. Описание кода

Таймер — это специализированный тип часов, используемых для измерения определенных временных интервалов. Таймеры можно разделить на два основных типа. Таймер, который отсчитывает время вверх от нуля для измерения прошедшего времени, часто называют секундомером, в то время как устройство, которое отсчитывает время вниз от заданного интервала времени, чаще называют таймером.

1. Код программы
2. unit Unit1;
3. {$mode objfpc}{$H+}
4. interface
5. uses
6. Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,
7. Spin, ExtCtrls, MMSystem, DateUtils, LCLType;
8. type
9. CompInfo=record
10. index,top,left,width,height,fontsize:integer;
11. end;
12. complist=array of CompInfo;
13. { TMform }
14. TMform = class(TForm)
15. Start\_kino: TButton;
16. Stop\_Kino: TButton;
17. Chistka: TButton;
18. Minuts: TEdit;
19. Secunds: TEdit;
20. trimer: TLabel;
21. Minuts\_txt: TLabel;
22. Secunds\_txt: TLabel;
23. Timer: TTimer;
24. procedure Start\_kinoClick(Sender: TObject);
25. procedure Stop\_KinoClick(Sender: TObject);
26. procedure ChistkaClick(Sender: TObject);
27. procedure MinutsChange(Sender: TObject);
28. procedure MinutsKeyPress(Sender: TObject; var Key: char);
29. procedure SecundsChange(Sender: TObject);
30. procedure SecundsKeyPress(Sender: TObject; var Key: char);
31. procedure FormCreate(Sender: TObject);
32. procedure PicClick(Sender: TObject);
33. procedure TimerTimer(Sender: TObject);
34. procedure FormResize(Sender: TObject);
35. private
36. { private declarations }
37. TimerStarted: Boolean;
38. DefWidth,defHeight:integer;
39. clist:complist;
40. FStartTime: TDateTime;
41. FDuration: TDateTime;
42. FStopTime: TDateTime;
43. TargetTime: TDateTime;
44. RemainingTime: TDateTime;
45. FSoundFile: string;
46. public
47. { public declarations }
48. procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
49. end;
50. var
51. Mform: TMform;
52. implementation
53. uses math;
54. {$R \*.lfm}
55. { TMform }
56. procedure TMform.FormCreate(Sender: TObject);
57. var i:integer;
58. begin
59. defwidth:=width;
60. defheight:=height;
61. for i:= 0 to ComponentCount-1 do
62. if (Components[i].Classname ='TCheckBox')
63. or (Components[i].Classname ='TButton')
64. or (Components[i].Classname ='TEdit')
65. or (Components[i].Classname ='TComboBox')
66. or (Components[i].Classname ='TLabel') then begin
67. setlength(clist,Length(clist)+1);
68. clist[Length(clist)-1].top:=(Components[i] as tcontrol).top;
69. clist[Length(clist)-1].left:=(Components[i]as tcontrol).left;
70. clist[Length(clist)-1].width:=(Components[i] as tcontrol).width;
71. clist[Length(clist)-1].height:=(Components[i]as tcontrol).height;
72. clist[Length(clist)-1].fontsize:=(Components[i]as tcontrol).font.Size;
73. clist[Length(clist)-1].index:=i;
74. end;
75. Minuts.MaxLength := 2;
76. Secunds.MaxLength := 2;
77. TimerStarted := False;
78. KeyPreview := True;
79. FSoundFile := 'ajjfon\_-\_budilnik\_radar\_76666078.wav';
80. OnKeyDown := @FormKeyDown;
81. RemainingTime := EncodeTime(23, 59, 59, 0);
82. trimer.Caption := FormatDateTime('hh:nn:ss', RemainingTime);
83. end;
84. procedure TMform.PicClick(Sender: TObject);
85. begin
86. end;
87. procedure TMform.FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
88. begin
89. if Key = VK\_ESCAPE then
90. begin
91. Close;
92. end;
93. end;
94. procedure TMform.FormResize(Sender: TObject);
95. var i:integer;
96. begin
97. For i:=0 to length(clist)-1 do begin
98. (components[clist[i].index] as tcontrol).Top:=round(clist[i].top\*height/defheight);
99. (components[clist[i].index] as tcontrol).height:=round(clist[i].height\*height/defheight);
100. (components[clist[i].index] as tcontrol).left:=round(clist[i].left\*width/defwidth);
101. (components[clist[i].index] as tcontrol).width:=round(clist[i].width\*width/defwidth);
102. (components[clist[i].index] as tcontrol).font.Size:=round(clist[i].fontsize\*min(width/defwidth,height/defheight));
103. end;
104. end;
105. procedure TMform.Start\_kinoClick(Sender: TObject);
106. var
107. Minutes, Seconds: Integer;
108. begin
109. if (Minuts.Text = '') and (Secunds.Text = '') then
110. begin
111. ShowMessage('Пожалуйста, заполните хотя бы одно поле времени.');
112. end;
113. if TryStrToInt(Minuts.Text, Minutes) and TryStrToInt(Secunds.Text, Seconds) then
114. begin
115. FStartTime := Now;
116. FDuration := Minutes / 1440 + Seconds / 86400;
117. TargetTime := EncodeTime(0, Minutes, Seconds, 0);
118. RemainingTime := TargetTime;
119. Timer.Enabled := True;
120. TimerStarted := True;
121. end
122. else
123. begin
124. ShowMessage('Пожалуйста, введите корректное время.');
125. end;
126. end;
127. procedure TMform.Stop\_KinoClick(Sender: TObject);
128. begin
129. Timer.Enabled := False;
130. FStopTime := Now;
131. if FSoundFile <> '' then
132. begin
133. sndPlaySound(PChar(FSoundFile), SND\_ASYNC);
134. end;
135. ShowMessage('Таймер остановлен');
136. end;
137. procedure TMform.ChistkaClick(Sender: TObject);
138. begin
139. Minuts.Clear;
140. Secunds.Clear;
141. end;
142. procedure TMform.MinutsKeyPress(Sender: TObject; var Key: char);
143. begin
144. if not (Key in ['0'..'9', #46]) then
145. begin
146. Key := #0;
147. end;
148. end;
149. procedure TMform.SecundsKeyPress(Sender: TObject; var Key: char);
150. begin
151. if not (Key in ['0'..'9', #46]) then
152. begin
153. Key := #0;
154. end;
155. end;
156. procedure TMform.MinutsChange(Sender: TObject);
157. var f : integer;
158. begin
159. if Minuts.Text <> '' then
160. begin
161. f := strtoint(Minuts.text);
162. if f <= 0 then
163. f := 0;
164. if f >= 59 then
165. f := 59;
166. Minuts.text := inttostr(f);
167. end;
168. end;
169. procedure TMform.SecundsChange(Sender: TObject);
170. var i : integer;
171. begin
172. if Secunds.Text <> '' then
173. begin
174. i := strtoint(Secunds.text);
175. if i <= 0 then
176. i := 0;
177. if i >= 59 then
178. i := 59;
179. Secunds.text := inttostr(i);
180. end;
181. end;
182. procedure TMform.TimerTimer(Sender: TObject);
183. var
184. ElapsedTime: TDateTime;
185. begin
186. if not TimerStarted then
187. Exit;
188. ElapsedTime := Now - FStartTime;
189. if MilliSecondsBetween(Now, FStartTime) < MilliSecondsBetween(0, FDuration) then
190. begin
191. RemainingTime := IncSecond(RemainingTime, -1);
192. trimer.Caption := FormatDateTime('hh:nn:ss', RemainingTime);
193. end
194. else
195. begin
196. Timer.Enabled := False;
197. TimerStarted := False;
198. if FSoundFile <> '' then
199. begin
200. sndPlaySound(PChar(FSoundFile), SND\_ASYNC or SND\_NODEFAULT);
201. end;
202. ShowMessage('Время вышло!');
203. end;
204. end;
205. end.
206. Результат выполнение программы

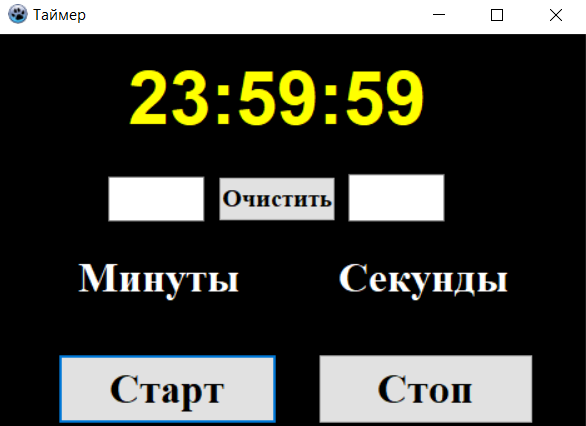


Рисунок 2 – результат выполнения программы 1

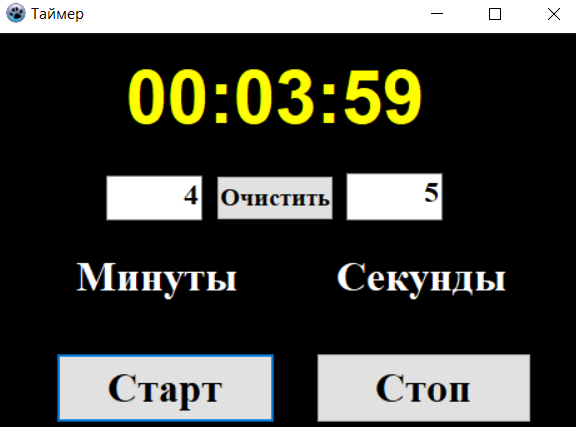


Рисунок 3 – результат выполнения 2

1. Вывод

В результате выполнения Контрольной работы №8, были созданы: код программы и результаты выполнения программы. При выполнении данной работы были получены дополнительные навыки работы с Lazarus и его компонентами. В таймере введены ограничения в каждое поле, для того чтобы реализовать корректную работу программы.