|  |  |
| --- | --- |
| Реализация caps lock и shift  begin  if lngbutton = True  then  if capsbtn = False then  begin  keybd\_event(VK\_CAPITAL,0,0,0);  keybd\_event(VK\_CAPITAL,0,KEYEVENTF\_KEYUP,0);  bigletter\_en;  capsbtn := True;  end  else begin  keybd\_event(VK\_CAPITAL,0,0,0);  keybd\_event(VK\_CAPITAL,0,KEYEVENTF\_KEYUP,0);  littleletter\_en;  capsbtn:=False;  end  else if capsbtn = false then  begin  keybd\_event(VK\_CAPITAL,0,0,0);  keybd\_event(VK\_CAPITAL,0,KEYEVENTF\_KEYUP,0);  bigletter\_RU;  capsbtn := true;  end  else begin  keybd\_event(VK\_CAPITAL,0,0,0);  keybd\_event(VK\_CAPITAL,0,KEYEVENTF\_KEYUP,0);  littleletter\_ru;  capsbtn:=false;  end  end;  procedure TForm1.Button76Click(Sender: TObject);  begin  if k = False then begin  keybd\_event(VK\_SHIFT, 0, 0, 0);  k:=True;  end  else begin  shiftoff;  k:=False;  end;  end; | Перегружен код, реализацию отпускания шифта придется вносить в обработку нажатия каждой кнопки.  Решение:  Отслеживать нажатие служебных кнопок в отдельной функции:  Function TFormKey.GetUnicodeSymbol(k:word):widechar;   procedure VK\_ALL\_CLEAR;    begin  KS[VK\_CAPITAL]:=0;  KS[VK\_SHIFT]:=0;  KS[VK\_CONTROL]:=0;  KS[VK\_MENU]:=0;  end; |
| Реализация нажатия каждой кнопки:  procedure TForm1.Button66Click(Sender: TObject);  begin  keybd\_event(37,0,0,0);  keybd\_event(37,0,KEYEVENTF\_KEYUP,0);  end;  …..  …..  procedure TForm1.Button1Click(Sender: TObject);  begin  keybd\_event(VK\_ESCAPE,0,0,0);  keybd\_event(VK\_ESCAPE,0,KEYEVENTF\_KEYUP,0);  end; | Куски одинакового кода придется повторять для всех буквенных и цифровых клавиш.  Решение:  Отправлять код клавиши следующей процедурой  var VK, SC, C: cardinal;  Ch: Char;  modeKey:(symbol, nosymbol);      begin  // получаем скан код клавиш  SC := MapVirtualKey(k, MAPVK\_VK\_TO\_VSC);// получаем скан-код  // отделяем символьные клавиши от остальных  if ((SC>=2)and(SC<=13)) OR  ((SC>=16)and(SC<=27)) OR  ((SC>=30)and(SC<=41)) OR  ((SC>=44)and(SC<=53)) OR (SC=57)  then modeKey:=symbol  else modeKey:=nosymbol;  // обрабатываем клавиши  case modeKey of  // работаем с несимвольными  nosymbol: begin  case SC of // переключаем состояние на противоположное  {Caps} 58:begin  KS[VK\_CAPITAL] := KS[VK\_CAPITAL] xor $1; // младший бит  Caps\_Pressed:= not(Caps\_Pressed);  end;    {Shift} 42:begin  KS[VK\_SHIFT] := KS[VK\_SHIFT] xor $80; // старший бит  SHIFT\_Pressed:= not(SHIFT\_Pressed);  end;  {Ctrl} 29:begin  KS[VK\_CONTROL] := KS[VK\_CONTROL] xor $81; // ??  CONTROL\_Pressed:= not(CONTROL\_Pressed);  end;  {Alt} 56:begin  KS[VK\_MENU] := KS[VK\_MENU] xor $81;// ??  ALT\_Pressed:= not(Alt\_Pressed);  end;  end;  // записываем состояние клавиатуры  SetKeyboardState(KS);  end; // конец секции {nosymbol}    // РАБОТАЕМ с символьными  symbol : begin  VK := MapVirtualKey(SC, MAPVK\_VSC\_TO\_VK);// скан-код переводится в виртуальный код  C := MapVirtualKey(VK, MAPVK\_VK\_TO\_CHAR);// виртуальный код в код симола по ASCII  if (ToUnicode(VK,SC,KS,@Ch,Sizeof(Ch),0) = 1) then GetUnicodeSymbol:=Ch;  VK\_ALL\_CLEAR; // очистим <Shift> <Ctrl> <Alt>  end; // конец секции symbol  end; |
| Активация постоянного фокуса на окне.  SetWindowLong(Handle, GWL\_EXSTYLE,  GetWindowLong(Handle, GWL\_EXSTYLE) or WS\_EX\_TOPMOST or WS\_EX\_NOACTIVATE  ); | Мене громоздко выполняется через внутреннюю среду Delphi:  *Form.ShowModal* |
| Маллуев Сергей Юрьевич |