

---

’

---

’ . . .

# VISUAL BASIC

”

”

8 09.03.04

2004



.....	4
<i>1</i> .....	5
2 .....	13
3 .....	21
4 .....	29
Visual Basic .....	36
, Visual Basic .....	37
.....	38



*I*

:

Visual Basic.

**1**

**1.1**

**Visual Basic**

Visual Basic

- ,

-

.

Visual Basic

-

:

1

( ).

2

, .

3

.

4

,

-

.

-

.

-

(Form\_Load),

(Click)

(

,

(DbClick)

-

,

-

,

).

-

Visual Basic,



←

;

← PictureBox,

;

← Label,

- Caption

;

← TextBox,

,

/

.

- Text;

← Frame,

;

← CommandButton,

;

← CheckBox,

;

← OptionButton,

;

← ComboBox,

;

← ListBox,

;

← HScrollBar,

;

← VScrollBar,

;

← Timer,

(

)

;

← DriveListBox,

-

;

← DirListBox,

(

);

← FileListBox,

;

← Shape,

:

,

,

;

← Line,

;

← Image,

;

← Data,

.



Private Sub Command1\_Click()

End Sub

Visual Basic

General (Declarations)



F5.

< > < >

< >: < >  
< >

.frm

.bas.

Visual Basic.

### 1.2

### Visual Basic

[LET] < ' > = < >

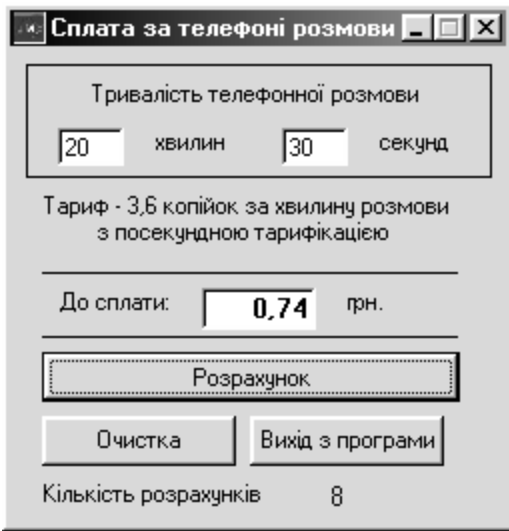
- 1) ^ - ;
- 2) ( ) - ;
- 3) \*, / - ;
- 4) \ - ( 17\2 8);
- 5) MOD - (17 MOD 2 1);
- 6) +, - - ( . 36).

REM

(:). : = 1 : = + 0.1.

### 1.3

### Visual Basic



1.2 –

Label, Line  
CommandButton.

( . 1.1)

TextBox,

: < >, Caption ( ).

( . 1.2):

TextBox

– TextBox  
– TextBox

Label1 – Label7–

Caption

Lbl –

Cmd1 – Cmd1

Cmd2 – Cmd2

Cmd3 – Cmd3

CommandButton Label

Caption.

Cmd1

“ ”

( )

Integer,

Visual Basic,

( . . 37).



```
String,
Val(),
< > - ' < > = Val(< > .text),
Single, Long Double;
< > - ' ;
text -
```

```
Dim i As Integer
Private Sub Form_Load()
    i = 0
    lbl1.Caption = i
    A.Text = ""
    B.Text = ""
    C.Text = ""
End Sub
Private Sub Cmd1_Click()
    Dim Sec, Minut, CC As Single
    Minut = Val(A.Text)
    Sec = Val(B.Text)
    CC = (Sec / 60 + Min) * 0.036
    C.Text = Round(CC, 2)
    i = i + 1
    lbl1.Caption = i
End Sub
Private Sub Cmd2_Click()
    A.Text = "" : B.Text = "" : C.Text = ""
End Sub
Private Sub Cmd3_Click()
    End
End Sub
```

## 2

```
1 Visual Basic?
2 Visual Basic.
3
4 Visual Basic.
5 ?
6
7 String.
8
```

## 3

( \* )  
 ,  
 :  
 ;  
 ;  
 .

## 4

1  
 $S = p(p - a)(p - b)(p - c), \quad p = (a + b + c)/2; a, b, c$   
 $(x_1, y_1), (x_2, y_2)$   
 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ .

2  
 $v = 1/L$   
 $L, L, T = 2\pi\sqrt{LC}$

3  
 $S = 1/2ah, a, h$

4  
 5  
 $( \dots 1 )$   
 $- a, b, c,$

6  
 $22 - 4y - 20z - 45 = 0$   
 $3x - y + 5z + 1 = 0,$   
 $\rho = \frac{|ax_0 + y_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}}$ .

7  
 $n = 1000$   
 $\lambda = 0,03$   
 $\tau = 3 \cdot 10^{-7}$   
 $W_1 = P_1 \tau;$   
 $k = c\tau/\lambda, c = 30.$

8  
 $ax^2 + bx + c = 0,$   
 $b, c ( a \neq 0 ), a = 2, b = -8,$   
 $c = -10$

9	$a, b.$	$a, b$	$2x/a + b - 12 = 0$	-
10				-
11	$t.$	$t$	$S = t^3 - \sqrt{t}.$	$($
12				$)$
13	$R_3$	$Z = (V_1 + V_2 + V_3)/3,$	$V_1, V_2, V_3 -$	$R_1, R_2,$
14				$V = 4/3 \pi R^3.$
15		$L = 0,04$		$= 10^{-6},$
16			$I = U \sqrt{\frac{C}{L}};$	$U = 100$
17	$n = 2$			$W = \frac{LI^2}{2}.$
18		$($	$a$	$0,5 \sqrt{2b^2 + 2c^2 - a^2}.$
19		$a, b, c$		
20			$W_1 = \frac{\Delta W}{n^2 - 1}$	$L = 0,5,$
				$\Delta W = 3$
				$I_1 = \sqrt{\frac{2W_1}{L}}.$
			$\frac{1}{b+c} \sqrt{bc(a+b+c)(b+c-a)}.$	$a, b, c$
		$Z = (R_1 + R_2 + R_3)/3,$	$R_1, R_2, R_3 -$	$R = \sqrt[3]{3V/4\pi}.$
		$V_1, V_2, V_3$		

21  $x$

$$\operatorname{sh}(x) \cdot \operatorname{tg}(x+1) - \operatorname{tg}^2(2 + \operatorname{sh}(x-1)), \quad \operatorname{sh}(x) = \frac{e^x - e^{-x}}{2}. \quad x$$

22  $a, b, c$

$$0,5 \sqrt{2b^2 + 2c^2 - a^2}.$$

23  $u/U_{\max}$   $n$

$$\frac{u}{U_{\max}} = \sqrt{\frac{n}{n+1}}; \quad \frac{t}{T} = \frac{\arccos \sqrt{\frac{n}{n+1}}}{2\pi}.$$

24  $b, V = h(S_1 + \sqrt{S_1 S_2} + S_2)/3; S_1, S_2 -$

$$a, b, h$$

25  $\cdot = / \cdot 100 \%$

$$2 \%$$

26  $p = 120 \left( \ln \frac{2\lambda}{\pi d} - 0.577 \right); \lambda = (3 + 0.1n).$

27  $a, R = \frac{a}{2 \sin(\pi/n)}, r = \frac{a}{\operatorname{tg}(\pi/n)}.$

28  $V = h(S_1 + \sqrt{S_1 S_2} + S_2)/3; S_1, S_2 -$

$$R, r, h$$

29  $(x, y).$

$$(\rho, \varphi), \quad \rho = \sqrt{x^2 + y^2}, \quad \operatorname{tg}(\varphi) = y/x.$$

30  $S = t^3 - 3t^2 + 2.$

2

:

Visual Basic.

1

, ( )

➤ : ;

➤ ;

➤ ); ,

1.1

Visual Basic

: If, Elseif, Select Case.

1.1.1

If / Then / Else:

If

IF < > THEN < 1> [ELSE < 2>]
------------------------------

1.1.2

If < > Then ... Else ... End If:

If < > Then
< 1, ( )>
[Else
< 2, ( )>]
End if

Else

Then End if

(False).

1.1.3

If /Then /ElseIf /EndIf:

```

If < 1 > Then
< , 1 >
ElseIf < 2 > Then
< , , 2 >
[ElseIf < 3 > Then
< , , 3 >]
[Else
< , >]
End if
    
```

1.1.4.

Select Case:

```

Select Case < >
Case < 1 >
< 1 >
Case < 2 >
< 2 >
[.....
Case Else
< N >]
End Select
    
```

Select Case

Case.

To,

: 10 To 20;

Is,

: Is > 200.

1.2

OptionButton

< ' \_

: Value:  
>.Value = True ( False)

CheckBox

< ' \_

: >.Value = 0 (1 2)  
; 1 - ; 2 - ( -

0 -

ComboBox

( , ): ,

<List>

1)

2)

```

AddItem:
< ' _ComboBox>.AddItem [< >]
Dan(5)
    
```

```

cbo 1.Clear
For i=1 To 5
cbo 1.AddItem Dan(i)
Next i
    
```

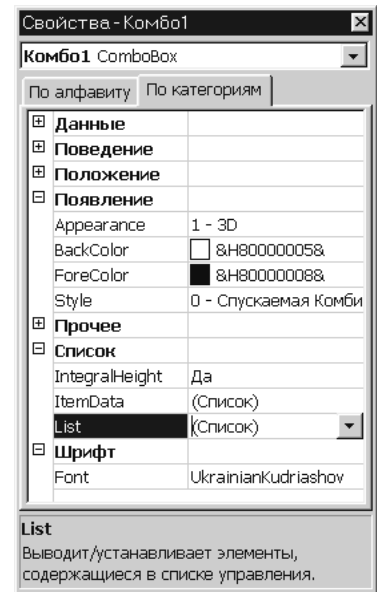
3)

```

cboN.Clear
For i=1 To 5
    cbo 1.List(i-1) = Dan(i)
Next i
Sorted
True False.
    
```

<Ctrl>+<Enter>.

<List>.



### 1.3

Visual Basic

For, While Do.

#### 1.3.1

#### For...Next

<pre> For &lt; &gt; = &lt; 1&gt; To &lt; 2&gt; [Step &lt; &gt;] Next [ ]         </pre>
---

1, . Next

2 , ,

[Step < >]

Next.

1).

Next

For...Next (

For...Next

).

1.3.2.

For Each...Next

```

For Each <          > In <          >
    <          >
Next <          >
    
```

For Each...Next

Next.

Variant  
For Each...Next,

1.3.3.

While...Wend

```

While <          >
    <          >
Wend
    
```

True.

< > - , True False.  
Null, False.  
< > - ,  
True, While -

False.

While...Wend

1 100

```

Dim Sum, i As Integer
Sum = 0
i = 2
While i <= 100
    sum = sum + i
    i = i + 2
Wend
    
```

1.3.4

Do...Loop

, 2)

: While Until.

: 1)



1	2	3	4
<b>Do While</b> < > < > <b>Loop</b>	<b>Do Until</b> < > < > <b>Loop</b>	<b>Do</b> < > <b>Loop While</b> < >	<b>Do</b> < > <b>Loop Until</b> < >

1. While ( ).

(True).  
(False),

2. Until ( ).

( ).

3 4. ,

,  
,

### 1.4

#### 1.4.1

#### Goto:

**GOTO** < >

aaa: x=x+1

...  
**GOTO** aaa

#### 1.4.2

#### Exit :

**EXIT** < >

**Exit For**  
**If...Then).**

Next.

### 1.5

ComboBox

**Rnd.**

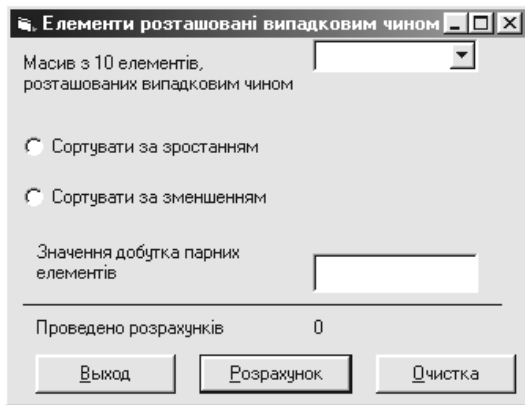
100,

**Int,**

0 1,

0 100.

”  
”  
’ lbl\_i ( . 2.1).



2.1 –

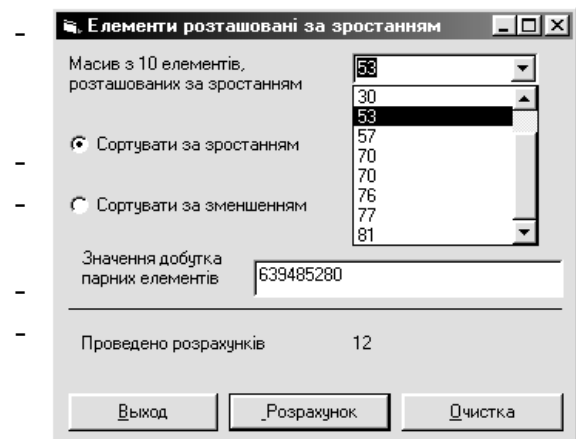
```
Dim i, j, n, k As Integer
Private Sub Form_Load()
    n = 10
    For i = 1 To n
        .List(i - 1) = Int (Rnd * (100))
    Next i
    k = 0
    lbl_i.Caption = k
    A = ""
    = ""
    = ""
End Sub
```

Option1,

ComboBox

( . 2.2).

```
Private Sub Option1_Click()
    Dim z As Integer
    Lbl5.Caption = " " & n & " "
    ,
    Formal.Caption = " " , "
    For i = 0 To n - 2
        For j = i + 1 To n - 1
            If Val( .List(i)) > Val( .List(j)) Then
                z = .List(i)
                .List(i) = .List(j)
                .List(j) = z
            End If
        Next j
    Next i
End Sub
```



2.2 –

**2.**

1 ?  
 2  
 3 ? -  
 4 ?  
 5 .  
 6 .

**3**

( \*.exe)  
 ,  
 .  
 ComboBox AddItem List ( . 2.1, 2.2).  
 .

( . 2.1) TextBox.  
 :  
 ■ ;  
 ■ TextBox ComboBox;  
 ■ .

2.1 –

1	15	
2	10	
3	8	
4	12	
5	14	
6	18	8 ,
7	11	
8	14	
9	16	,
10	19	, 6

## 2.1

11	17	
12	9	7 ,
13	15	,
14	10	,
15	8	
16	12	,
17	20	
18	18	, -
19	11	,
20	9	, -
21	16	
22	19	
23	17	,
24	8	,
25	7	, -
26	18	,
27	11	,
28	10	,
29	16	
30	9	
31	17	, -

: Visual Basic

1

1.1

Visual Basic

MouseDown ( ), MouseUp ( ) MouseMove ( )

MouseDown –

MouseUp –

;

MouseMove –

Button –

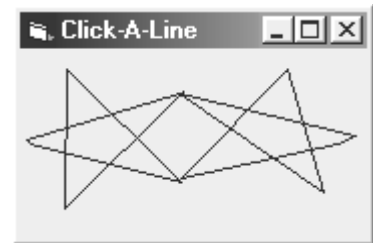
Shift –

Shift, Ctrl Alt;

1

Line

MouseDown



3.1 –

1

( , ):

Private Sub Form\_MouseDown (Button As Integer, Shift As Integer,X As Single, Y As Single)

Line -(X, Y)

End Sub



3.2 –

2

2.

Line

MouseMove

MouseMove

Private Sub Form\_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

Line -(X, Y)

End Sub



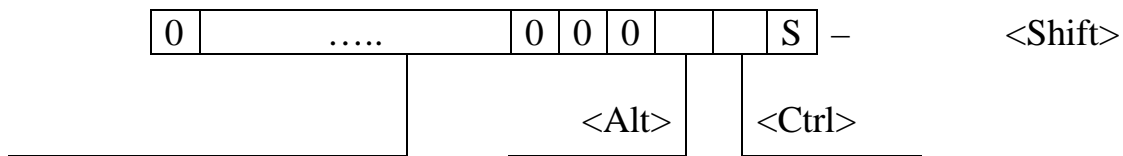
3 (011).

7 (111).

**MouseMove**

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then Print " "
    If Button = 2 Then Print " "
    If Button = 3 Then Print " "
End Sub
```

Shift, <Shift>, <Ctrl> - 2, <Shift>, <Alt>, <Ctrl>  
 <Shift>, <Ctrl> <Alt>.  
 Shift 1; <Ctrl> - 2,  
 <Alt> - 4. <Shift> <Alt>,  
 Shift 5 (101). Shift <Shift>, <Ctrl>  
 <Alt>.



```
Private Sub Form_MouseDown(button As Integer, shift As Integer, x As Single, y As Single)
    shiftTest = shift And 7
    Select Case shiftTest
        Case 1 'vbShiftMask
            Print "You're pressed the shift key."
        Case 2 'vbCtrlMask
            Print "You're pressed the Ctrl key."
        Case 3
            Print "You're pressed both shift and Ctrl key."
        Case 4 'vbAltMask
            Print "You're pressed the Alt key."
        Case 5
            Print "You're pressed both shift and Alt key."
        Case 6
            Print "You're pressed both Ctrl and Alt key."
        Case 7
            Print "You're pressing the all three key."
    End Select
End Sub
```

## 1.2

### 1.2.1

### Line

**Line** [Step] (X1, Y1) - [Step] (X2, Y2), [color], [ ] [F]

X1 Y1, - X2, Y2

**Line** (X1, Y1) - (X2, Y2)

**Step** **Line**

### ForeColor

**color,** **Line** **Circle** (

, **Line** (100, 10) - (100, 410), RGB(255, 0, 0) ‘  
**Line** (200, 10) - (200, 410), RGB(255, 255, 255) ‘  
 Clr# = RGB( 0, 255,0)  
**Line** (300, 10) - (300, 410), Clr# ‘

**DrawWidth,** - **DrawStyle**

( . 3.1).

**DrawStyle** 1...4

0.

### 3.1 –

### DrawStyle

vbSolid	0	( )
vbDash	1	
vbDot	2	
vbDashDot	3	-
vbDashDotDot	4	- -
vbInvisible	5	
vbInsideSolid	6	

Form1:

```
Private Sub Form1_Click()
    Hstep = Form1.ScaleHeight / 6
```

1/6

```
For i% = 1 to 6
```

```
    Form1.DrawStyle = i% - 1
```

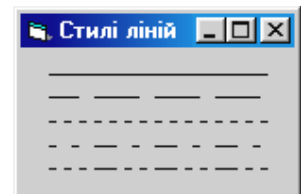
```
    Form1.Line (0.1 * Form1.ScaleWidth, Hstep * i%) -(0.9 * Form1.ScaleWidth, Hstep * i%)
```

```
        Step 0.1 0.9
```

1/10

```
Next
```

```
End Sub
```





100 300, (100, 400),

**Line** (100, 400) - (200, 400) ’  
**Line** (200, 400) - (200, 700) ’  
**Line** (200, 700) - (100, 700) ’  
**Line** (100, 700) - (100, 400) ’

Y  
X

**Line** (x,y) - **Step** (100, 0)  
**Line** - **Step**(0,300)  
**Line** - **Step**(-100,0)  
**Line** - **Step**(0,-300)

, Visual Basic

**Line** (Box – ).

( **Step** )

**Line** (100,400) - (200,700), ,

**color**

Visual Basic

**F** (Fill – )

: ( ) **BF** ( ) (

**F** ).

**FillColor.**

**FillStyle.**

**FillStyle** ( . 3.2).

3.2 –

**FillStyle**

vbFSSolid	0	
vbFSTransparent	1	( )
vbHoriwntallLine	2	( )
vbVerticalLine	3	
vbUpwardDiagonal	4	
vbDownwardDiagonal	5	
vbCross	6	
vbDiagonalCross	7	

## 1.2.2

## Circle

Circle:

**Circle [Step] (X, Y), radius, [color], [start], [end], [aspect]**

**Y** – , a **radius** – .  
**R** : **X, Y,** -

**Circle (X, Y), R**

**Circle**(Form1.ScaleWidth / 2, Form1.ScaleHeight / 2),Form1.ScaleHeight / 3

**Line,**  
**Step.**

. **Aspect**

**Circle** –

**aspect**

**aspect**

1,

**Circle** (x, y), r, 255, , , 1.5

i# = RGB(0, 155, 0)

r% = 0

D2 = False

**While** r% < 1000

**Circle** (x, y), r%, i#, , , 0.5

    i# = i# + 10

    r% = r% + 10

**Wend**

**Circle**

**start**

**end**

:

360

, 2π

D

2πD/360.

2

1  
2  
3  
4  
5

?

**Button.**  
**Shift.**  
**Line.**  
**Circle.**

3

-

**Shift**

**Button.**

3.3

**Shift**

**Line**

**Circle.**

3.3 –

		<Shift>	<Ctrl>	<Alt>		<Shift>	<Ctrl>	<Alt>
1	2				5			
2	3				1			
3	4				2			
4	5				1			
5	6				2			
6	7				2			
7	8				3			
8	7				5			
9	6				1			
10	5				2			
11	4				1			
12	3				2			

## 3.3

		<Shift>	<Ctrl>	<Alt>		<Shift>	<Ctrl>	<Alt>
13	2				5			
14	1				3			
15	2				5			
16	3				1			
17	4				2			
18	5				1			
19	6				2			
20	7				5			
21	8				3			
22	9				1			
23	8				2			
24	7				1			
25	6				2			
26	5				2			
27	4				3			
28	3				5			
29	2				5			
30	1				3			
31	2				5			

: Visual Basic -

**1**

**1.1**

(keyboard handler).

( )

Visual Basic

KeyPress KeyDown KeyUp	, ASCII-
------------------------------	----------

**KeyDown** **KeyPress,** **KeyUp,**

**KeyPress** **KeyDown,**

**KeyUp.**

**KeyPress**

ASCII:

<Enter>, < b> <Backspace>.

**KeyPress**

**KeyPress,**

ASCII.

**KeyDown** **KeyUp,**

```

    ( KeyUp). KeyPress ( KeyDown)
    -
    -
    -
    . KeyPress
    -
    ASCII.
    KeyDown KeyUp
    .
    KeyPress , :
    ➤ <Shift>, <Ctrl> <Alt>;
    ➤
    ( , )
    :
    ;
    ➤ <PageUp> <PageDown>;
    ➤
    ;
    ➤
    KeyPress );
    ➤ (<Delet >, <BackSpace> );
    ➤ <F1>--<F12>,
    .
    KeyPress KeyUp KeyDown,
    KeyDown KeyUp
    :
    1) KeyCode . < >
    < >
    KeyCode. , "1" "1"
    , "1" "!" ASCII-
    .
    2) KeyCode,
    Shift <Shift>, <Ctrl> <Alt>.
    .
    Shift
    .
    1
    , Text1. <Home>,
    :
    Private Sub Text1_KeyDown (KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyHome Then MsgBox "You pressed the HOME key."
    End Sub
    vbKeyHome
    .
    <F1>, vbKeyF1, <F2> - vbkeyF2
    .
    ,
  
```

```

        <F5>
        .
        Shift
        ,
        Shift
        ,
        <Shift>, <Ctrl>
        KeyDown
        :
        <Alt>.
        KeyUp,
        .
        2
        KeyUp,
        <F5>, <F6>, <F7>, <F8>
        Text1.
        <F5>
        Date,
        -
        <F6> -
        Time.
        <F7>
        -
        <F8>
        ,
        .

```

```

Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
Const DateKey = vbKeyF5, TimeKey = vbKeyF6
Const SpecKey1 = vbKeyF7, SpecKey2 = vbKeyF8
Select Case KeyCode:
Case DateKey: Text1.SelText = Date
Case TimeKey: Text1.SelText = Time
Case SpecKey1: Text1.SelText = "Vishual Basic 6.0"
Case SpecKey2: Text1.SelText = " <F8>"
End Select
End Sub

```

```

3
A:
Shift
,
-

```

```

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = vbKeyA And Shift = 1 Then MsgBox "You pressed the A key."
End Sub

```

## 1.2

```

        KeyDown    KeyUp
        ,
        ,
        KeyPreview
True.
        KeyPress,
KeyUp    KeyDown
        ,
        .
        KeyPreview    True
        <
        >
Form_Load:
Private Sub Form_Load
Form1.Keypreview = True
End Sub

```

```

True, <Enter> Default,
lick
Cancel, <Esc> True
Click
KeyAscii KeyCode
0.
<Tab>
TabStop False.

```

### 1.3

#### Visual Basic

```

)
Interval -
0 64 767 ( 0 64,8 ).
Enabled -
TRUE,
( ). False
Timer
Interval
4
Times1,
:

```

```

Private Sub Times1_Timer()
L b 11.BorderStyle = 1
L b 11.Caption = Time
End Sub

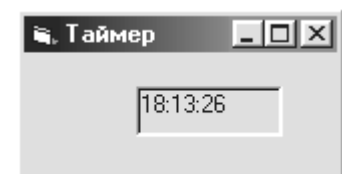
```

5

```

Dim T1 As Double
Private Sub Form_Load()
T1 = Time
LblTimer.Caption = 100
End Sub

```





```

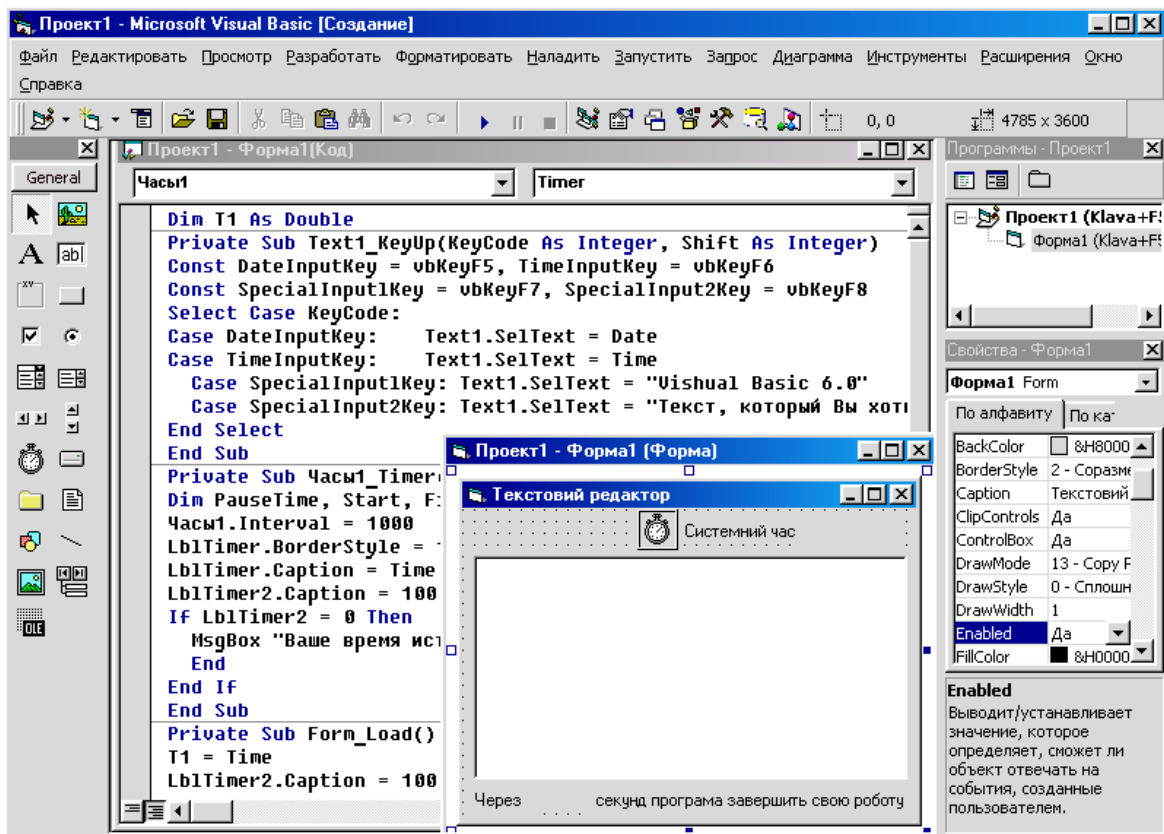
Private Sub 1_Timer()
Times1.Interval = 1000 ' - 1
LblTimer.Caption = 100-((Second(Time)+Minute(Time)*60)-_
(Second(T1)+Minute(T1)*60))

If LblTimer = 0 Then
MsgBox " !"
End
End If
End Sub

```

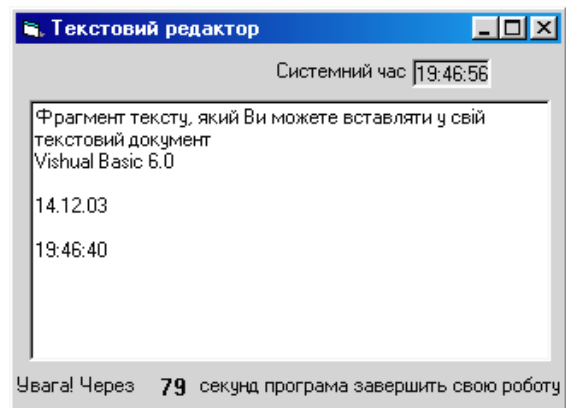
. 4.1  
2, 4 5,

. 4.2.



4.1 –

- 2
- 1
  - 2
  - 3
  - 4
- ?  
KeyCode.  
Shift.



4.2 –



## 4.1

1	2	3	4	5
21	, - ,	F9	Microsoft Excel	<Alt>+O
22		F10		<Shift>+P
23		F11	!	<Ctrl>+R
24		F12		<Alt>+S
25		F1		<Shift>+T
26		F2		<Ctrl>+U
27		F3	!	<Alt>+V
28		F4		<Shift>+W
29		F5	Windows XP	<Ctrl>+X
30		F6		<Alt>+Y
31		F7		<Shift>+Z

## Visual Basic

	Abs( )	
	Sqr( )	
	Exp( )	
	Log( )	
	Cos( )	.
	Sin( )	.
	Tan( )	.
	Atn( )	.
	Rnd	0 1
	Str( )	
	Val(str)	
	Asc(s)	ASCII-
	Chr(int)	, ASCII-
	Lcase(str)	, - ,
	Ucase(str)	, - ,
	Len(str)	
	Right(str,int)	int str
	Left(str,int)	int str
	Cbool( )	Boolean
	Cbyte( )	Byte
	Ccur( )	Currency
	Cdate( )	Date
	CDbl( )	Double
	Cint( )	Integer
	CLng( )	Long
	CSng( )	Single
	CStr( )	String
	Cvar( )	Variant

**Visual BASIC**

Byte ( )	1	0 255	
Boolean ( )	2	True False	
Integer ( )	2	-32 768 32 767	%
Long ( )	4	-2 147 483 648 2 147 483 647	&
Single ( )	4	-3,40282E38 -1,4013E-45 1,4013E-45 3,40282E38	!
Double ( )	8	-1,797E308 -4,94E-324 4,94-324 1,797E308	#
Currency ( )	8	-922 337 203 685 477,5808 922 337 203 685 477,5807	@
Date ( )	8	1 100 . 31 9999 .	
Object ( ' )	4	- ,	
String ( )	10 +	0 2 .	\$
Variant ( - )	16	-	Double
( - )	22 + -	(String)	

- 1 . Visual Basic 6. : 2 .-
- 2 .: V, 2000. Visual Basic 6. - : -
- 3 - " , 2000. - 624 .
- 4 . Microsoft Visual Basic, Scripting Editing . -
- 5 : , 1997. - 448 .
- 6 Visual Basic 6: . .
- 7 - .- .- .: , 1999. - 832 .
- 8 : , : . / . . . - .: , 2001. -
- 9 696 .
- 10 **Visual Basic 6:** . .- : - - , 2003. - 992 .



”

”

## VISUAL BASIC

. . .

— , . . .

. . .

,

. . .