1.4. ТЕОРИЯ И МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ В ШКОЛЕ И ВУЗЕ

УДК 778.14

TECHNOLOGY OF DESIGNING AND REALIZATION OF COMPUTER TEST FOR CHECKING THE KNOWLEDGE OF STUDENTS IN VISUAL BASIC PROGRAMMING ENVIRONMENT

M.J. Zaripova, Z.S. Djumaev, J. Zaripov

In the article is presented the technology of designing the computer test in Visual Basic programming environment. It also deals with the particularities of computer test in educational process. : Summarizing, we should note that using of such test checking allows the teacher: Conducting checking by minimum taking time; conclude on the level of mastering the material; find out what questions are mastered better; get objective evaluation of the knowledge, since the pattern is impartial and does not depend on mood of the teacher. Moreover, composed test may be integrated in electronic textbook as subsystems of continuous checking the knowledge.

Key words: Test, Visual Basic, New, Existing, Resent

One of the important components of educational process and progressive methods of teaching is a checking, realized by tests. Tests as measuring tool are used in most of countries of the world. Their design and use are founded on powerful theory and is proved by numerous researches [1-8]. Checking by test ensures the feedback of the teacher with trained. The most important task of the test is a reception of objective mark of knowledge.

In comparison with traditional forms of checking the computer testing has a number of advantages:

quick reception of results;

objectivity of evaluation of knowledge;

tests are more interesting in comparison with traditional forms that influences on improving the cognitive activity of learners and produces at them positive motivation and etc.

Computer tests may be used in taking any type of checking: intermediate, current, final. Designing computer tests, draws interest of teachers of different discipline, including the teachers of informatics science. However, for many of them designing computer tests on taught discipline calls a significant difficulty because of incompetency in the field of independent writing of tests by programming.

The suggested by authors technology designing computer test may interest any teacher engaging in designing tests, as well as people incompetent in this field.

The current article introduces the technology of designing computer test in Visual Basic programming environment. The given technology may be used as a manual on composing computer test by teachers of different disciplines.

The reason of choosing precisely this programming language for designing test may be several reasons.

Firstly, Visual Basic advantageously differs from the other languages by its simplicity and clarity that users enough quickly master the language that allows producing them their own Windows-application with professional interface.

Secondly, Visual Basic is alive, dynamically developing language. It does get old, "marches in step" with development of computers and computer technology. Every following version increases the capacity of previous one [9].

Thirdly, modern information technologies are based on principle of the object-oriented programming so it is inevitable transition to study the language founded on these principles of programming.

Thus, we’ll consider the sequence of action required for designing test.

Project: Test.

Problem statement: Design project (testing programme), in which is counted the number of correct answer. The project must consist of two forms:

Form 1 – input: It requests username, but afterwards typed small greeting on the form.

Form 2 – output: the form of the test, in it for each question it is necessary to choose one correct answer from four offered and result of the test.

Action order.

1. Start Visual Basic. After its start on the monitor will be displayed dialogue windows *New Project*. The window allows the user a possibility to choose further action and contains three tabs:

New – beginning of designing new project;

Existing – selection of applications from existing projects;

Resent – list of most recent projects.

On default there will be open tab **New** and selected the type of programming project *Standart.EXE.*

2. Click on the button <Open>. As a result you enter environment of Visual Basic – an integrated environment of designing project. When designing new project in Visual Basic is opened new form. In window characteristic *Properties* initially presented characteristics forms.

3. Change the value of characteristic form by means of *window characteristic Properties*:

in characteristic *Caption* delete the value Form1 and type the phrase Greeting. Already in process of typing the headline form will be changed. So, characteristic *Caption* defines the type of the headline of the form;

change the background color of the form – a characteristic *BackColor*. Having clicked on characteristic *BackColor*, you'll see the button list. Open the list, tab **Palette** and in appeared palette of colour select the colour. You'll see that the colour of the form has changed.

4. In the centre of the form place a label. In order to this:

in window tools ToolBox select (Label) A if in Main window of the project there is no *Window tools Toolbox*, add it (select the commands menu **View**, **Toolbox** or click on button ⚒ on the panel of tools). The Window tools – a main operation tool in visual design of the forms of application. The window contains the pictographs of control elements. With their help there designed objects on the form of your project and you construct graphical part of the programme – a graphical user's interface (Graphical User Interface – GUI);

the characteristic *Caption* install value *Welcome*;

the characteristic *Alignment* (leveling) install value 2 – Center (along the centre);

by means of characteristic *Font* at own discretion change the font, outline and size of the font.

5. Dispose on the form the command button. In order to do this:

in *Window tool ToolBox* click on the button (CommandButton) □;

draw in window of the form rectangle with the help of mouse, it'll become a button;

do so that on button was written *Click here* (the characteristic Caption);

with the help of characteristic *Font* at own discretion change the font, outline and size of the font.

6. Install value of the characteristic *Name* for the form and button.

for the form *frmGreeting*;

for the button *cmdGreeting*.

7. Start the program to completion (select the commands menu **Run, Start**, or press the key [F5], or click on the button ▶ on panels of tool). The window of the form will take an operating view. It may be treated as with ordinary window Windows: fold, unfold, restore in sizes, drag over for caption, close.

8. Click on the button <Click here> – nothing will happen.

9. Abort completion of the program (select the commands menu Run, End or click on the button ■ on the panel tool or on the button ☒ in the window of Application).

10. Programme the action of the button i.e. report what the computer must do at clicking on the button.

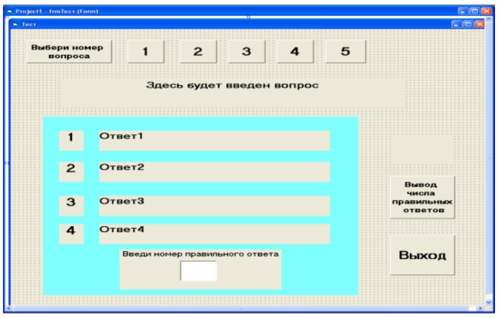
11. For designing relevant procedures twice click on the button <Click here>. The *Window programme code* will be opened with preparation procedure. In the second line *Window programme code* there are two windows list: List of objects and List of events. In the left is specified *object*, which is bound with procedure. In the right window is specified event, which must be made in order to accomplish the given procedure.

12. Type the programme code of the project.

```
Private Sub cmdGreeting_Click()
Dim YuorName As String, t As String
```

```
YuorName = InputBox ("Enter your name", "Name")
cmdGreeting.Visible = False
t = 0 + 64
MsgBox "Hello" + YuorName + "!  Pleased to greet You in our project!", t, "Hello!!!"
K = MsgBox(YuorName + "!  You are agree to pass testing?", vbYesNo + vbQuestion, "Testing!!!")
If K = 6 Then
Print "Pressed the button 'Yes' "
Else
Print "Pressed the button 'No' "
End If
t = 0 + 64
MsgBox YuorName + "! Wish success!", t, "Good luck!!!"
With frmGreeting
.BackColor = QBColor(8)
.FontName = "Georgia"
.FontSize = 14
.FontItalic = True
.FontBold = True
.ForeColor = QBColor(6)
End With
frmGreeting.Hide
frmTest.Show
End Sub
```

13. Add in the project the second form. The form of the test. (Form2). Dispose the objects on the form in accordance with pic 1.


Project1 - frmTest (Form)
Тест
Выбери номер вопроса
1
2
3
4
5
Здесь будет введен вопрос
1
Ответ1
2
Ответ2
3
Ответ3
4
Ответ4
Введи номер правильного ответа
Вывод числа правильных ответов
Выход

Pic. 1. The form of the test

14. Install the value of the characteristic object in accordance with table 1.

Table 1

The characteristic object

Object	Characteristic	Characteristic value
(Form2)	Name	frmTest
	Caption	Test
Command button 1	Name	Command1
	Caption	Select question number
	Style	1 – Graphical
	Picture	Fit relevant picture (icon)
Command button 2	Name	Command2
	Caption	1
Command button 3	Name	Command3
	Caption	2
Command button 4	Name	Command4
	Caption	3
Command button 5	Name	Command5
	Caption	4
Command button 6	Name	Command6
	Caption	5
	Style	1 – Graphical (for all buttons)
Command button 7	Name	Command7
	Caption	Display the number of correct answers
	Style	1 – Graphical
Command button 8	Name	Command8
	Caption	Exit
	Style	1 – Graphical
Frame	Name	Frame1
	Caption	Frame1
	BorderStyle	0 – None
Label 1	Name	Label1
	Caption	Enter question here
	BorderStyle	0 – None
Label 2	Name	Label2
	Caption	1

Object	Characteristic	Characteristic value
Label 3	Name	Label3
	Caption	2
Label 4	Name	Label4
	Caption	3
Label 5	Name	Label5
	Caption	4
	Alignment	2 – Center (for all labels)
Label 6	Name	Label6
	Caption	Answer1
Label 7	Name	Label7
	Caption	Answer2
Label 8	Name	Label8
	Caption	Answer3
Label 9	Name	Label9
	Caption	Answer4
	Alignment	0 – Left Justify (for all labels)
Label 10	Name	Label10
	Caption	Enter correct answer here
	Alignment	2 – Center
Label 11	Name	Label11
	Caption	Delete
	Alignment	2 – Center
	Font	Font size – 24, bold
Text box 1	Name	Text1
	Text	Delete
	Alignment	2 – Center
	Font	Font size – 24, bold
	Visible	True
Text box 2	Name	Text2
Text box 3	Name	Text3
Text box 4	Name	Text4
Text box 5	Name	Text5
	Text	Delete
	Alignment	2 – Center
	Font	Font size – 24, bold
	Visible	False

Select the sizes and colour of the font yourself, using characteristic corresponding to component.

15. Type the programming code in accordance with table 2. In the beginning of the programming code add the operator *Option Explicit* to avoid the troubles in case of wrong writing the name variable.

Here If Val(Text1.Text) = 1 Then K = K + 1 means if the entered number corresponds to the number of the correct answer(1), that variable K increases to 1.

Table 2

The programming code

Programming code start	**Command button 2** **Code buttons**	**Command button 3** **Code buttons**

Option Explicit
Public K As Integer
	Private Sub Command2_Click()
	Label1.Caption = "Question 1"
	Label6.Caption = "2"
	Label7.Caption = "3"
	Label8.Caption = "4"
	Label9.Caption = "5"
	Command2.Visible = False
	Text1.Text = ""
	Text2.Visible = False
	Text3.Visible = False
	Text4.Visible = False
	Text5.Visible = False
	End Sub
		Private Sub Command3_Click()
		Label1.Caption = "Question 2"
		Label6.Caption = "Answer1"
		Label7.Caption = "Answer2"
		Label8.Caption = "Answer3"
		Label9.Caption = "Answer4"
		Command3.Visible = False
		Text1.Text = ""
		Text1.Visible = False
		Text2.Visible = True
		End Sub

Command button 1 **Code buttons**		
Private Sub Command1_Click() K = 0 End Sub		
Command button 7 **Code buttons**		
Private Sub Command7_Click() Label11.Caption = K End Sub		

Command button 4 **Code buttons**	**Command button 5** **Code buttons**	**Command button 6** **Code buttons**

Private Sub Command4_Click()
Label1.Caption = "Question 3"
Label6.Caption = "Answer1"
Label7.Caption = "Answer2"
Label8.Caption = "Answer3"
Label9.Caption = "Answer4"
Command4.Visible = False
Text1.Text = ""
Text3.Visible = True
End Sub
	Private Sub Command5_Click()
	Label1.Caption = "Question 4"
	Label6.Caption = "Answer1"
	Label7.Caption = "Answer2"
	Label8.Caption = "Answer3"
	Label9.Caption = "Answer4"
	Command5.Visible = False
	Text1.Text = ""
	Text4.Visible = True
	End Sub
		Private Sub Command6_Click()
		Label1.Caption = "Question 5"
		Label6.Caption = "Answer1"
		Label7.Caption = "Answer2"
		Label8.Caption = "Answer3"
		Label9.Caption = "Answer4"
		Command6.Visible = False
		Text1.Text = ""
		Text5.Visible = True
		End Sub

Command button 8 **Code buttons**	**Text box 1** **Code (TextBox)**	**Text box 2** **Code (TextBox)**
Private Sub Command8_Click() End End Sub	Private Sub Text1_Change() If Val(Text1.Text) = 1 Then K = K + 1 End If End Sub	Private Sub Text2_Change() If Val(Text2.Text) = 2 Then K = K + 1 End If End Sub
Text box 3 **Code (TextBox)**	**Text box 4** **Code (TextBox)**	**Text box 5** **Code (TextBox)**
Private Sub Text3_Change() If Val(Text3.Text) = 3 Then K = K + 1 End If End Sub	Private Sub Text4_Change() If Val(Text4.Text) = 4 Then K = K + 1 End If End Sub	Private Sub Text5_Change() If Val(Text5.Text) = 1 Then K = K + 1 End If End Sub

16. Start the program to completion. Save the given project in the folder *Project_1*. The file name of the project – Test.vbp

17. In order to operate Windows-applica-tion regardless of environment of the designing, it is necessary to produce the translation of the project into performable exe-file. For this: accomplish the commands menu File, Make FileName.exe. (FileName – this is your project name i.e. *Test.exe*). On default file will be created in the same folder, where the project is saved.

18. Close the environment of the programs design Visual Basic. Start the file Test.exe to completion and make sure that it operates autonomously.

Results: Summarizing, we should note that using of such test checking allows the teacher:

1. Conducting checking by minimum taking time.

2. Conclude on the level of mastering the material.

3. Find out what questions are mastered better.

4. Get objective evaluation of the knowledge, since the pattern is impartial and does not depend on mood of the teacher.

Moreover, composed test may be integrated in electronic textbook as subsystems of continuous checking the knowledge.

References

1. Kudaev M.R. Korrektiruyushchiy kontrol' v uchebnom protsesse. Problemy i metody postroeniya i realizatsii ego sistemy. Maykop: Kachestvo, 1997.
2. Lerner I.Ya. Protsess obucheniya i ego zakonomernosti. M.: Znanie, 1980.
3. Talyzina N. F. Teoreticheskie problemy programmirovannogo obucheniya. M.: Znanie, 1968.
4. Brusilovsky P., Miller P. Web-based testing for distance education // WebNet'99. ngs of AACE World Conference of the WWW and Internet.Honolulu, HI, 1999. P. 149-154.
5. Avanesov V.S., Getmanenko G.E. Metodicheskie printsipy kompozitsii testovykh zadaniy: Ucheb. posobie dlya prepodavateley gumanitarnykh distsiplin. M., 1996.
6. Zaytseva L.V., Prokof'eva N.O. Problemy komp'yuternogo kontrolya znaniy // Proceedings. IEEE International Conference on Advanced Learning Technologies (ICALT 2002). 9-12 September 2002. Kazan, Tatrstan, Russia, 2002. P. 102-106.
7. Sinitsa E.M., Burtsev M.S. Opisanie uchebnykh resursov: metadannye, standarty, profili // Obrazovatel'nye tekhnologii i obshchestvo. 2006. T. 9. №1. S. 365-396.
8. Bashmakov A.I., Bashmakov I.A. Razrabotka komp'yuternykh uchebnikov i obuchayushchikh sistem. M.: Filin, 2003.
9. Guseva O.L. Praktikum po Visual Basic. M.: Finansy i statistika, 2007.