

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

## **РАБОТА СО СПИСКАМИ В VBA**

отчет о лабораторной работе №11

по дисциплине

*ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ*

Выполнила: студент гр. 230711

Павлова В.С.

Проверил: ассистент каф. ИБ

Курбаков М.Ю.

Тула, 2023 г.

## **ЦЕЛЬ РАБОТЫ**

**Цель:** научиться работать с различными видами списков для выбора значений переменной в контексте GUI.

## **ЗАДАНИЕ НА РАБОТУ**

В данной работе требуется создать проект с пользовательской формой, в котором присутствуют такие элементы управления, как списки, а также пользовательскую структуру данных Student для описания информации о студентах и заполнения ими вышеупомянутых списков.

## ТЕКСТ ПРОГРАММЫ

Для описания форм в качестве альтернативы VBA был использован язык С# и платформа WinForms. Содержимое кодового файла, описывающего разработанный проект для работы со списками, представлено в листингах 1-4 для каждой из форм №1-4 соответственно. Описание класса Student приведено в листинге 5.

### Листинг 1. Описание формы Form1.cs

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void form2ShowOnClick(object sender, EventArgs e)
    {
        Form2 frm2 = new Form2();
        frm2.Show();
        this.Hide();
    }
}
```

### Листинг 2. Описание формы Form2.cs

```
partial class Form2 : Form
{
    public List<Student> listOfStudents;
    public Form2()
    {
        listOfStudents = new List<Student>(){
            new Student("Булочкин", "Игорь", "Михайлович", "17.01.1999", 3.4f),
            new Student("Котиков", "Михаил", "Юрьевич", "19.08.1997", 4.9f),
            new Student("Бублик", "Александр", "Сергеевич", "30.09.2003", 4.3f),
            new Student("Яблочков", "Валерий", "Ярославович", "03.03.2003", 5f),
            new Student("Кексик", "Вероника", "Сергеевна", "12.05.2003", 4.8f),
        };
        InitializeComponent();
    }
    private void outputInExel(object sender, EventArgs e)
    {
        if (listOfStudents.Count > 0)
        {
            Microsoft.Office.Interop.Excel.Application ExcelApp = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel.Workbook ExcelWorkBook;
            Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;

            ExcelWorkBook =
ExcelApp.Workbooks.Add(System.Reflection.Missing.Value);

            ExcelWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

            ExcelApp.Cells[1, 1] = "№п/п";
            ExcelApp.Cells[1, 2] = "ФИО";
        }
    }
}
```

## Листинг 2. Описание формы Form2.cs (продолжение)

```
ExcelApp.Cells[1, 3] = "Дата рождения";
ExcelApp.Cells[1, 4] = "Средний балл";

for (int i = 2; i < listOfStudents.Count + 2; i++)
{
    ExcelApp.Cells[i, 1] = i-1 + ".\t";
}
for (int i = 0; i < listOfStudents.Count; i++)
{
    ExcelApp.Cells[i+2, 2] = listOfStudents[i].FullName;
    ExcelApp.Cells[i+2, 3] = listOfStudents[i].BirthDay;
    ExcelApp.Cells[i+2, 4] = listOfStudents[i].AverageScore;
}
ExcelApp.Visible = true;
ExcelApp.UserControl = true;
}

private void showStudentInfoOnClick(object sender, EventArgs e)
{
    if (studentBox.SelectedIndex != -1)
    {
        Form3 frm3 = new Form3();
        if (optionBox.SelectedIndex == 0)
        {
            frm3.option = Options.AllInfo;
        }
        else
        {
            frm3.option = Options.Marks;
        }
        frm3.currentStudent = listOfStudents[studentBox.SelectedIndex];
        frm3.Show();
    }
}

private void showAllStudentsOnClick(object sender, EventArgs e)
{
    Form4 frm4 = new Form4();
    frm4.studentsList = listOfStudents;
    this.Hide();
    frm4.Show();
}
}
```

## Листинг 3. Описание формы Form3.cs

```
public partial class Form3 : Form
{
    public Student currentStudent;
    public Options option;
    public Form3()
    {
        InitializeComponent();
    }
    private void Form3_Load(object sender, EventArgs e)
    {
        if (option == Options.AllInfo)
        {
            label5.Visible = false;

            textBox1.Text = currentStudent?.LastName;
            textBox2.Text = currentStudent?.FirstName;
        }
    }
}
```

### Листинг 3. Описание формы Form3.cs (продолжение)

```
        textBox3.Text = currentStudent?.MiddleName;
        textBox4.Text = currentStudent?.BirthDay;
        textBox5.Text = currentStudent?.AverageScore.ToString();
    }
    if (option == Options.Marks)
    {
        label1.Visible = false;
        label2.Visible = false;
        label3.Visible = false;
        label4.Visible = false;
        textBox1.Visible = false;
        textBox2.Visible = false;
        textBox3.Visible = false;
        textBox4.Visible = false;

        textBox5.Location = new Point (textBox5.Location.X, 101);
        labelAS.Location = new Point(labelAS.Location.X, 101);
        textBox5.Text = currentStudent?.AverageScore.ToString();
    }
}
```

### Листинг 4. Описание формы Form4.cs

```
public partial class Form4 : Form
{
    public List<Student> studentsList = new List<Student>();
    public Form4()
    {
        InitializeComponent();
    }

    private void Form4_Load(object sender, EventArgs e)
    {
        dataGridView1.RowCount = studentsList.Count;
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Cells[0].Value = i + 1;
            dataGridView1.Rows[i].Cells[1].Value = studentsList[i].FullName;
            dataGridView1.Rows[i].Cells[2].Value = studentsList[i].BirthDay;
            dataGridView1.Rows[i].Cells[3].Value =
studentsList[i].AverageScore;
        }
    }
}
```

### Листинг 5. Имплементация класса Student

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StudentsInfo
{
    public class Student
    {
        private string _firstName;
        public string FirstName { get { return _firstName; } }

        private string _lastName;
        public string LastName { get { return _lastName; } }
    }
}
```

## Листинг 5. Имплементация класса Student (продолжение)

```
private string _middleName;
public string MiddleName { get { return _middleName; } }

private string _birthday;
public string BirthDay { get { return _birthday; } }

private string _fullName;
public string FullName { get { return _fullName; } }

private float _averageScore;
public float AverageScore { get { return _averageScore; } }

public Student(string lastName, string firstName, string middleName,
string birthday, float score)
{
    this._firstName = firstName;
    this._lastName = lastName;
    this._middleName = middleName;
    this._birthday = birthday;
    this._fullName = _lastName + " " + _firstName + " " + _middleName;
    this._averageScore = score;
}
}
```

## ДЕМОНСТРАЦИОННЫЙ ПРИМЕР

При запуске приложения отображается маленькое первичное окно с кнопкой для перехода к рабочей области со списками (рисунок 1):

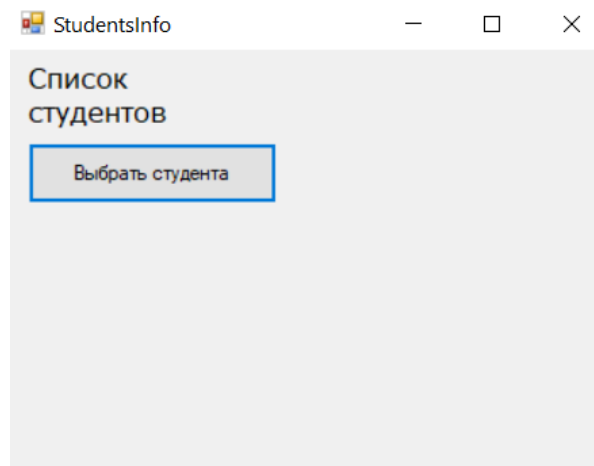


Рисунок 1 – Первичная форма

После нажатия на кнопку «Выбрать студента» появляется следующее окно (рисунок 2). В нём можно выбрать конкретного студента, а также тип сведений, которые о нём нужно получить.

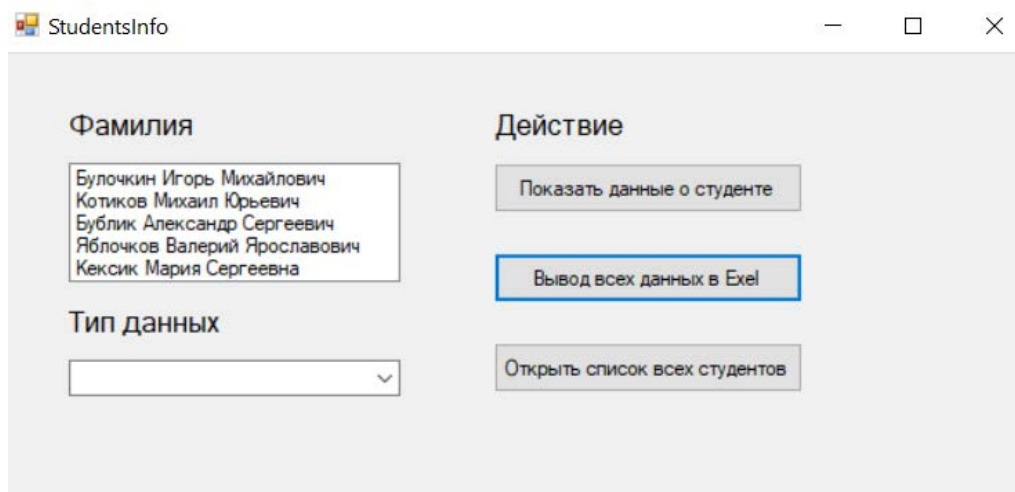


Рисунок 2 – Окно с опциями выбора

При нажатии на кнопку «Показать данные о студенте» в соответствии с выбранной опцией (общая информация или успеваемость) появляется одно из двух окон. На рисунке 3 показано, как выглядит общая информация:

 The screenshot shows the 'StudentsInfo' window displaying student details. It contains five rows of labels and text boxes: 'Фамилия' (Surname) with 'Котиков', 'Имя' (Name) with 'Михаил', 'Отчество' (Patronymic) with 'Юрьевич', 'Дата рождения' (Date of birth) with '19.08.1997', and 'Средний балл' (Average grade) with '4.9'.

Рисунок 3 – Вывод всей информации о студенте

Рисунок 4 иллюстрирует, что выводится при выборе опции «Успеваемость».

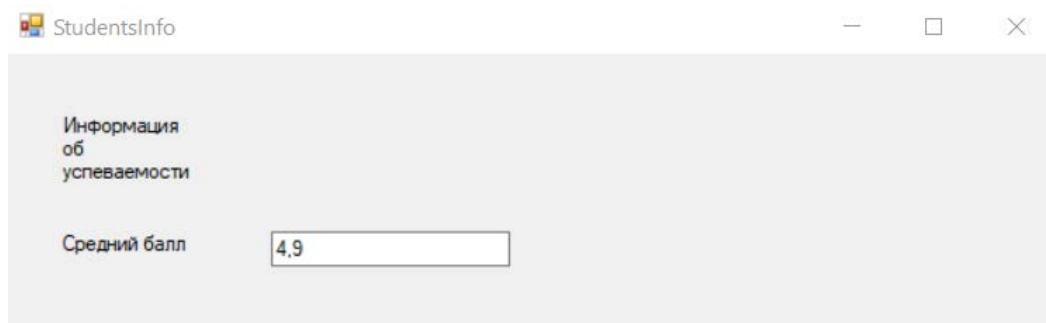


Рисунок 4 – Вывод информации об успеваемости

При нажатии на кнопку «Вывод всех данных в Excel» отображается следующее (рисунок 5):

	A	B	C	D	E
1	№п	ФИО	Дата рождения	Средний балл	
2	1.	Булочкин Игорь Михайлович	17.01.1999	3.4	
3	2.	Котиков Михаил Юрьевич	19.08.1997	4.9	
4	3.	Бублик Александр Сергеевич	30.09.2003	4.3	
5	4.	Яблочков Валерий Ярославович	03.03.2003	5	
6	5.	Кексик Вероника Сергеевна	12.05.2003	4.8	
7					
8					
9					

Рисунок 5 – Вывод всей информации в Excel

Наконец, при нажатии на кнопку «Открыть список всех студентов» выводится следующая таблица DataGridView (рисунок 6):



StudentsInfo

	№ п/п	ФИО	Дата рождения	Средний балл
▶	1	Булочкин Игорь Мих...	17.01.1999	3,4
	2	Котиков Михаил Юрь...	19.08.1997	4,9
	3	Бублик Александр С...	30.09.2003	4,3
	4	Яблочков Валерий Я...	03.03.2003	5
	5	Кексик Вероника Се...	12.05.2003	4,8

Рисунок 6 – Форма со списком всех студентов

## ВЫВОД

В ходе выполнения данной работы я научилась работать с различными списками при создании приложений с GUI.