

Министерство образования, науки и молодежной политики  
Краснодарского края  
Государственное бюджетное профессиональное образовательное учреждение  
Краснодарского края «Ейский полипрофильный колледж»

РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

## **ЛАБОРАТОРНАЯ РАБОТА №13**

**по теме:**

**Работа с элементами управления RadioButton и  
GroupBox**

**Выполнил:**


студент ЕПК, группа  
ФИО


**Проверил:**


преподаватель дисциплины  
«Разработка программных  
модулей»  
Фомин А. Т.


# 1 Теоретические сведения

## 1.1. Элементы управления «Кнопки».


Нажимаемой кнопкой (Button)  Button называется элемент управления, все взаимодействие пользователя с которым ограничивается одним действием — нажатием. Все, что вам необходимо сделать при работе с кнопкой, — это поместить ее в нужном месте формы и назначить ей соответствующий обработчик. Обработчик назначается для события Click.


Флажки (CheckBox)  CheckBox являются кнопками отложенного действия, т. е. их нажатие не должно запускать какое-либо немедленное действие. С их помощью пользователи вводят параметры, которые скажутся после, когда действие будет запущено иными элементами управления. Элемент CheckBox может иметь 3 состояния — помеченное, непомеченное и смешанное. Чаще всего этот элемент применяется для определения значений, которые могут иметь только два состояния.

**Радиокнопки (RadioButton)**  RadioButton по своим свойствам немного похожи на флажки. Их главное различие заключается в том, что группа флажков позволяет выбрать любую комбинацию параметров, радиокнопки же дают возможность выбрать только один параметр. Из этого различия проистекают и все остальные. Например, в группе не может быть меньше двух радиокнопок. Кроме того, у радиокнопок не может быть смешанного состояния.

**Блок группировки (GroupBox)**  GroupBox Блок группировки помогает визуально объединить несколько элементов управления в одну группу. Это бывает особенно полезно, когда надо придать вашему приложению более понятный пользовательский интерфейс. Например, объединить группу радиокнопок.

## 1.2. Элементы управления «Поля ввода» и «Списки».

**Поле ввода (TextBox)**  TextBox является основным элементом, предназначенным для ввода пользователем текстовых данных. Использовать TextBox можно в однострочном или многострочном режиме. Однако данный элемент управления имеет ограничение — до 64 кБайт текста. Если вам необходимо обрабатывать большие объемы информации, лучше использовать элемент RichTextBox.

**Расширенное поле ввода (RichTextBox)**  RichTextBox дает возможность пользователю вводить и обрабатывать большие объемы информации (более 64 кБайт). Кроме того, RichTextBox позволяет редактировать цвет текста, шрифт, добавлять изображения. RichTextBox включает все возможности текстового редактора Microsoft Word.

**Список (ListBox)**  ListBox представляет собой простейший вариант

пролистываемого списка. Он позволяет выбирать один или несколько элементов списка. Его дополнительное достоинство – возможность флажков рядом с каждым элементом списка. Пользователь имеет возможность отметить один или несколько элементов списка, выставив напротив его флажок.

**Выпадающий список (ComboBox)** удобен тем, что не занимает много пространства на форме. Постоянно на форме представлено только одно значение этого списка. При необходимости пользователь может раскрывать список и выбрать другое интересующее его значение. Кроме того, режим DropDown дает пользователю возможность вводить собственное значение при отсутствии необходимого значения в списке.

### 1.3. Элементы управления «Метки», «Индикаторы прогресса» и «Бегунки».

**Метка (Label)** предназначена для создания подписей к другим элементам управления или для вывода информационных сообщений прямо на поверхности формы.

**Метка-ссылка (LinkLabel)** представляет собой гиперссылку, которыми наполнен Интернет. Разработчики Visual Studio .NET представили этот элемент управления как разновидность метки (элемента управления Label). Однако элемент LinkLabel более похож на кнопку, чем на метку.

**Бегунок (TrackBar)**. Типичным примером применения элемента TrackBar является регулятор уровня громкости в панели Windows. TrackBar может использоваться в различных режимах: в горизонтальном или вертикальном положении, с включенными черточками или без.

**Индикатор прогресса (ProgressBar)** чаще всего используют для отображения степени завершенности той или иной задачи.

**Регулятор численных значений (NumericUpDown)** позволяет без помощи клавиатуры вводить численные значения в поле для ввода. Данный элемент управления имеет три возможности для ввода данных: щелчок мышкой на указатели вверх-вниз, использование кнопок вверх-вниз на клавиатуре или ввод данных в поле ввода.

## 2 Описание порядка выполнения работы

### *Постановка задачи*

Составить программу определения объема несжатого аудио-файла по данным: количеству каналов, глубине кодирования, частоте дискретизации и времени звучания.

1. Добавьте на форму три элемента GroupBox (Секция «Контейнеры»). Измените в свойствах элементов свойство «Text». Для первого элемента — значение «f» (частота дискретизации), для второго — значение «nu» (глубина кодирования) и для третьего — значение «n» (число каналов).
2. Добавьте в первый и второй контейнеры GroupBox по три элемента RadioButton. В третий контейнер два элемента RadioButton. После помещения элементов RadioButton в контейнер GroupBox они становятся связанными друг с другом. Найдите в свойствах элементов RadioButton свойство Checked. Установите значение этого свойства True для элементов RadioButton с текстом «11», «8» и «1».
3. Добавьте на форму 3 элемента Label, один элемент TextBox и одну кнопку Button. Измените свойство «Text» этих элементов, как показано на Рис. 1
4. Кликните дважды на элементе Button для перехода в редактор к функции обработчику событий нажатия кнопки button1\_Click(). Код обработчика приведен в листинге 1.

Листинг 1.

```
1. private void button1_Click(object sender, EventArgs e)
2. {
3.     double t = double.Parse(textBox1.Text);
4.
5.     double nu, n, f;
6.     if (radioButton1.Checked == true) {
7.         f = 11.0;
8.     } else if (radioButton2.Checked == true) {
9.         f = 22.05;
10.    } else {
11.        f = 44.1;
12.    }
13.    if (radioButton4.Checked == true) {
```

```

14.         nu = 8.0;
15.     } else if (radioButton5.Checked == true) {
16.         nu = 16.0;
17.     } else {
18.         nu = 32.0;
19.     }
20.     if (radioButton7.Checked == true) {
21.         n = 1.0;
22.     } else {
23.         n = 2.0;
24.     }
25.     double i = f * 1000 * nu * t * n * 60 / 8 / 1024 / 1024;
26.     label3.Text = i.ToString("f2") + " MiB";
27. }

```

The screenshot shows a Windows application window titled "Form1". Inside the window, there are three groups of radio buttons for selecting values for variables *f*, *nu*, and *n*.

- For variable *f*, the options are 11, 22,05, and 44,1. The option 11 is selected.
- For variable *nu*, the options are 8, 16, and 32. The option 8 is selected.
- For variable *n*, the options are 1 and 2. The option 1 is selected.

Below these groups, there is a text input field for the variable *t*, followed by the unit "min". At the bottom of the window, there is a button labeled "Вычислить" (Calculate).

**Рис. 1**

5. Оформите приложение произвольным образом.

### **3. Индивидуальные задания**

1. Разработать программу «конвертер физических величин», производящую перевод из одной единицы измерения в другую. Варианты: единицы массы, расстояния, скорости, площади, объема, температуры.
2. Использовать в программе обработчик исключительных ситуаций для перехвата ошибочного ввода данных в поле TextBox.
3. Оформить отчет по выполненной работе. Отчет должен включать: листинги Form1.cs и Form1.Designer.cs, скриншоты различных состояний окна приложения, вывод, содержащий рекомендации по улучшению функционала программы.