Программное управление элементами в ListBox

Добавление элементов

Для добавления нового элемента в эту коллекцию, а значит и в список, надо использовать метод **Add**, например:

```
listBox1.Items.Add("Новый элемент");
```

При использовании этого метода каждый добавляемый элемент добавляется в конец списка.

Можно добавить сразу несколько элементов, например, массив. Для этого используется метод **AddRange:**

```
string[] countries = { "Бразилия", "Аргентина", "Чили", "Уругвай", "Колумбия" };
listBox1.Items.AddRange(countries);
```

Вставка элементов

В отличие от простого добавления вставка производится по определенному индексу списка с помощью метода Insert:

```
listBox1.Items.Insert(1, "Парагвай");
```

В данном случае вставляем элемент на вторую позицию в списке, так как отсчет позиций начинается с нуля.

Удаление элементов

Для удаления элемента по его тексту используется метод Remove:

```
listBox1.Items.Remove("Чили");
```

Чтобы удалить элемент по его индексу в списке, используется метод RemoveAt:

```
listBox1.Items.RemoveAt(1);
```

Кроме того, можно очистить сразу весь список, применив метод Clear:

```
listBox1.Items.Clear();
```

Доступ к элементам списка

Используя индекс элемента, можно присвоить элемент в списке переменной. Например, получим первый элемент списка:

```
string firstElement = listBox1.Items[0];
```

Свойство **Count** возвращает количество элементов в списке:

```
int number = listBox1.Items.Count;
```

Тема 2.3.
Кнопки выбора RadioButton.

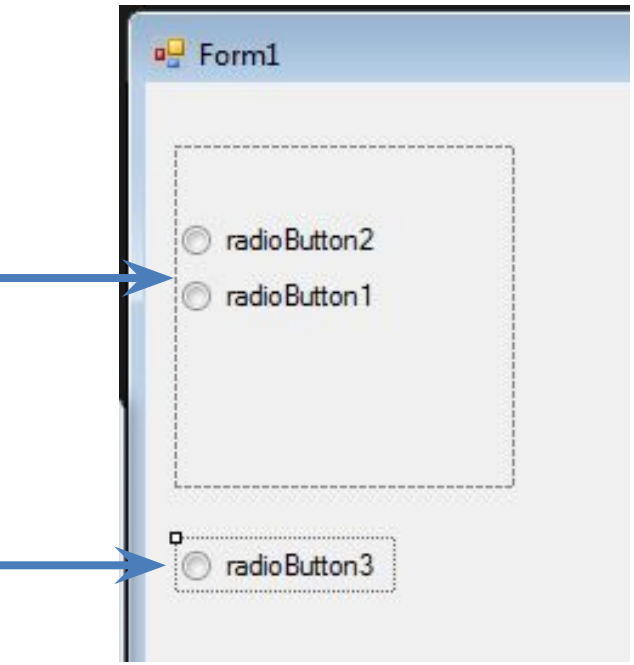
Вопросы:

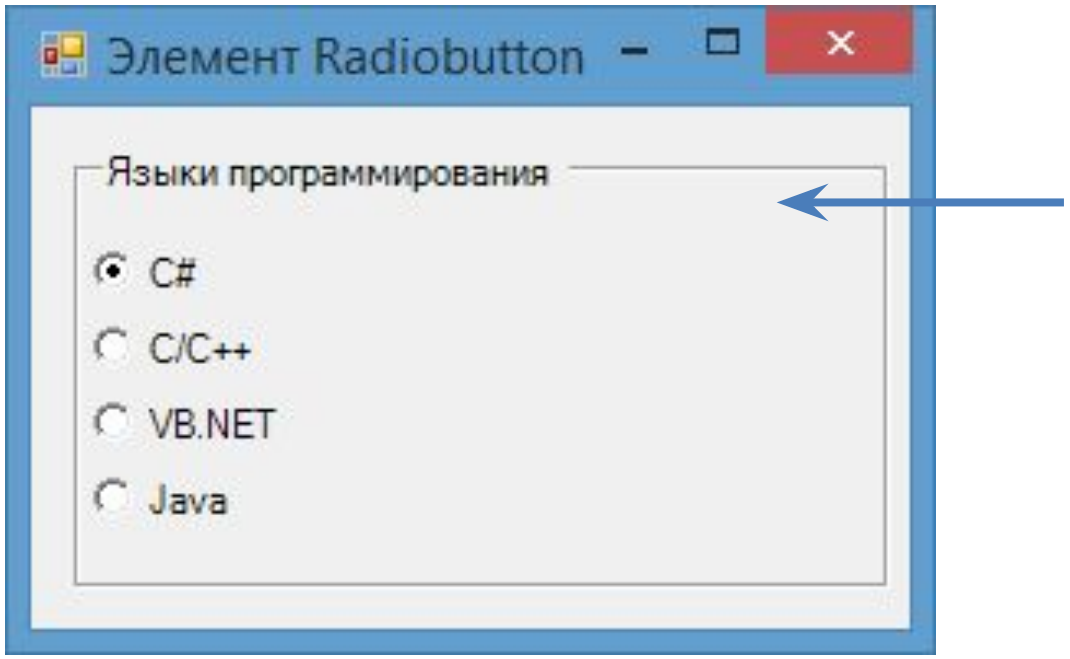
1. Элемент RadioButton.
2. Свойства, методы и события
3. Решение задач

Radiobutton

Переключатели располагаются группами, и включение одного переключателя означает отключение всех остальных.
Чтобы установить у переключателя включенное состояние, надо присвоить его свойству **Checked** значение **true.**

Для создания группы переключателей, из которых можно бы было выбирать, надо поместить несколько переключателей в какой-нибудь контейнер, например, в элементы **GroupBox** или **Panel**. Переключатели. находящиеся в разных контейнерах, будут относиться к разным группам:





Свойство Checked Позволяет узнать состояние, внешний вид кнопки. Если кнопка выбрана, то значения свойства **Checked** равно **True**; если не выбрано то **False**;

Свойство CheckAllign позволяет изменять положение кнопки в поле компонента, может быть прижата к границе в верху или внизу, размещена в центре, либо к краям, слева и с права.

Свойство TextAllign чем то похож на CheckAllign позволяет изменять положение текста в поле компонента, может быть прижата к границе в верху или внизу, размещена в центре, либо к краям, слева и с права.позволяет изменять положение текста в поле компонента, может быть прижата к границе в верху или внизу, размещена в центре, либо к краям, слева и с права.

Свойство Appearance позволяет изменить вид переключателя, переключатель может выглядеть как обычно и как кнопка.

Свойство FlatStyle позволяет изменить стиль кнопки, стандартной, плоской и всплывающей.

События:

Click	Это событие происходит при нажатии кнопки RadioButton.
CheckedChanged	Это событие происходит, когда изменяется значение Проверяемого свойства.
AppearanceChanged	Это событие происходит при изменении значения свойства внешнего вида.
DoubleClick	Это событие происходит, когда пользователь дважды щелкает элемент управления RadioButton.
Leave	Это событие происходит, когда фокус ввода покидает элемент управления RadioButton.
MouseClick	Это событие происходит, когда элемент управления RadioButton щелкает мышью.
MouseDoubleClick	Это событие происходит, когда пользователь дважды щелкает элемент управления RadioButton с помощью мыши.
MouseHover	Это событие происходит, когда указатель мыши находится на элементе управления RadioButton.
MouseLeave	Это событие происходит, когда указатель мыши покидает элемент управления RadioButton.

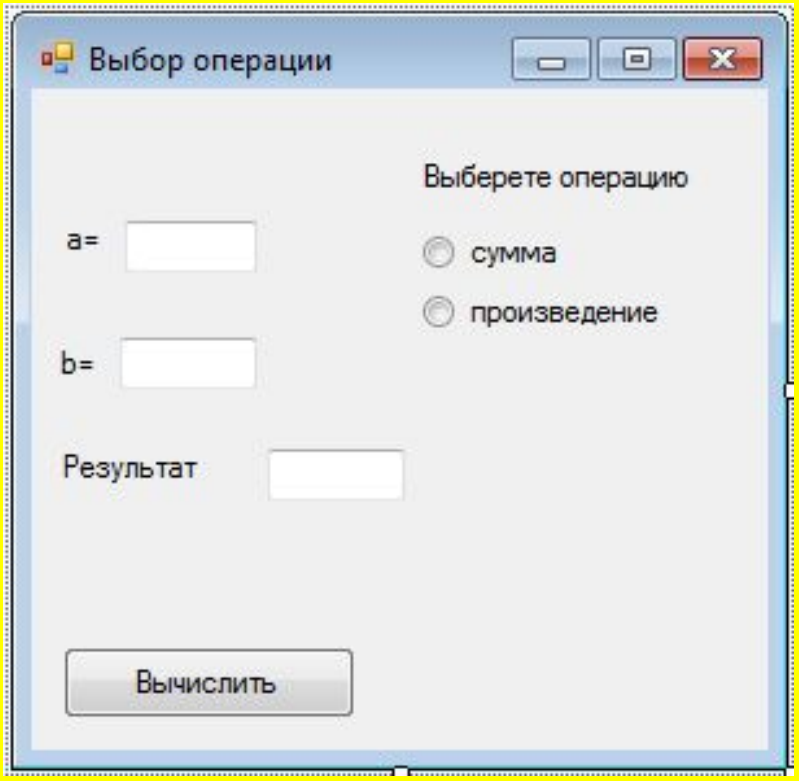
Похожим образом мы можем перехватывать переключение переключателей в группе, обрабатывая событие CheckedChanged. Связав каждый переключатель группы с одним обработчиком данного события, мы сможем получить тот переключатель, который в данный момент выбран:

Пример:

```
private void radioButton_CheckedChanged(object sender, EventArgs e)
{
   // приводим отправителя к элементу типа RadioButton
   RadioButton radioButton = (RadioButton)sender;
   if (radioButton.Checked)
   {
      MessageBox.Show("Вы выбрали " + radioButton.Text);
   }
}
```

Задание: Повторить пример

Пример: Произвести вычисления, после выбора операции:



```
private void button1_Click(object sender, EventArgs e)
{
double a = Convert.ToDouble(textBox1.Text);
double b = Convert.ToDouble(textBox2.Text);
double c = 0;
if (radioButton1.Checked == true)
c = a + b;
if (radioButton2.Checked == true)
c = a * b;
textBox3.Text = c.ToString();
}
```