

# **Тема 4. Работа с элементами управления. Списки, комбинированные поля, чекбоксы и радиокнопки.**

**Цель занятия:**

**Изучить работу с элементами управления, такими как списки, комбинированные поля, чекбоксы и радиокнопки.**

# **Учебные вопросы:**

- 1. Чекбоксы (CheckBox).**
- 2. Радиокнопки (RadioButton).**
- 3. Списки (ListBox).**
- 4. Комбинированные поля (ComboBox).**
- 5. Элементы управления для работы с датой и временем.**

# 1. Чекбоксы (CheckBox).

Назначение и использование:

- Выбор одного или нескольких вариантов.
- Использование для включения/выключения опций.

Основные свойства:

- Text: текст рядом с чекбоксом.
- Checked: состояние чекбокса (выбран/не выбран).
- ThreeState: поддержка третьего состояния (неопределенное).

Основные методы:

- `Check()`: установить состояние "выбран".
- `Uncheck()`: установить состояние "не выбран".

Основные события:

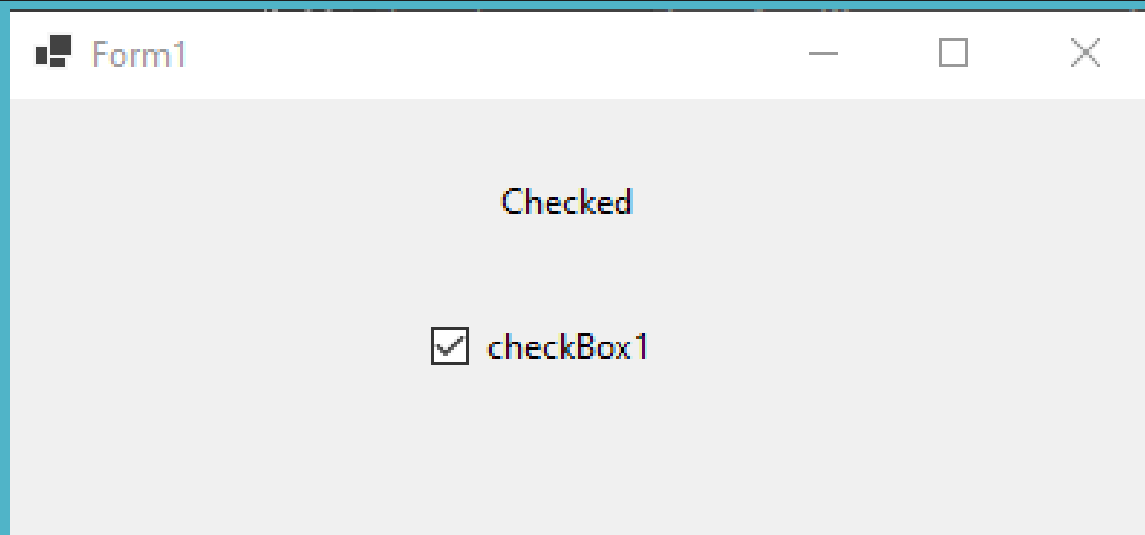
- `CheckedChanged`: изменение состояния чекбокса.

Пример использования:

- Создание группы чекбоксов.
- Обработка изменения состояния.

Пример обработки события `CheckedChanged` (изменение состояния чекбокса).

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked)
        label1.Text = "Checked";
    else
        label1.Text = "Unchecked";
}
```



The screenshot shows a Windows Form titled "Form1". Inside the form, there is a label displaying the text "Checked". Below the label, there is a checkbox labeled "checkBox1" which is currently checked, indicated by a small square with a checkmark inside it.

## 2. Радиокнопки (RadioButton).

Назначение и использование:

- Выбор одного варианта из группы.
- Использование для взаимоисключающих опций.

Основные свойства:

- Text: текст рядом с радиокнопкой.
- Checked: состояние радиокнопки (выбрана/не выбрана).
- AutoCheck: автоматическое снятие выбора с других радиокнопок в группе.

Основные методы:

- `Check()`: установить состояние "выбрана".
- `Uncheck()`: установить состояние "не выбрана".

Основные события:

- `CheckedChanged`: изменение состояния радиокнопки.

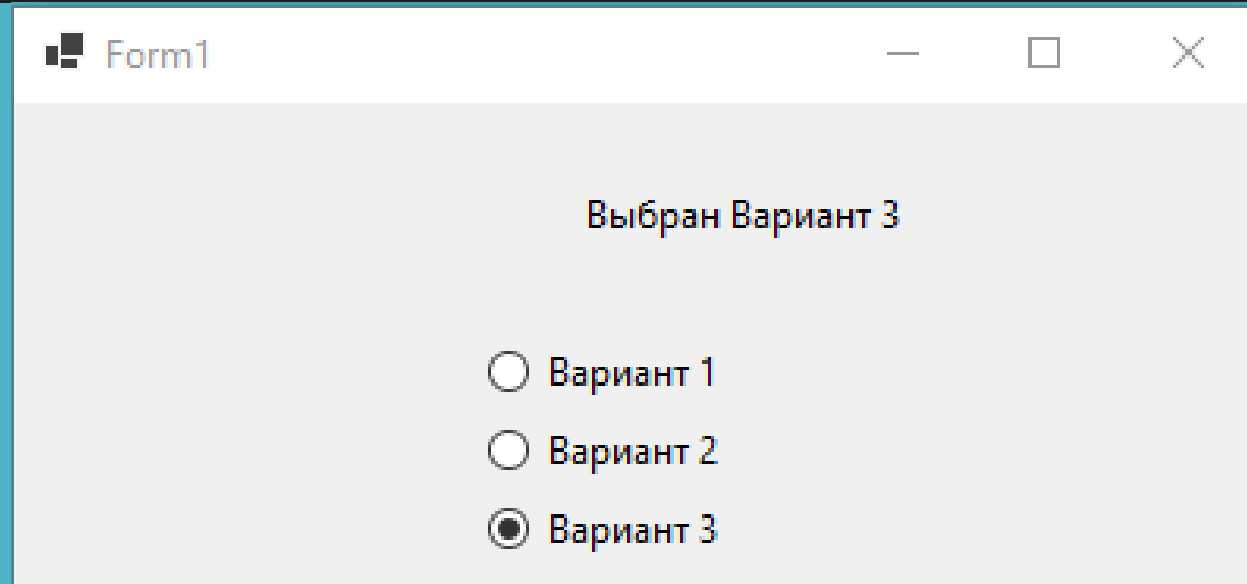
Пример использования:

- Создание группы радиокнопок.
- Обработка выбора радиокнопки.



# Пример обработки события CheckedChanged (изменение состояния радиокнопки).

```
private void radioButton_CheckedChanged(object sender, EventArgs e)
{
    // Проверяем, какая радиокнопка выбрана
    if (radioButton1.Checked)
        label1.Text = "Выбран Вариант 1";
    else if (radioButton2.Checked)
        label1.Text = "Выбран Вариант 2";
    else
        label1.Text = "Выбран Вариант 3";
}
```



Form1

Выбран Вариант 3

☐ Вариант 1

☐ Вариант 2

☒ Вариант 3

# 3. Списки (ListBox).

**Назначение и использование:**

- Отображение списка элементов.
- Поддержка множественного выбора.

**Основные свойства:**

- Items: коллекция элементов.
- SelectedItem: выбранный элемент.
- SelectedIndex: индекс выбранного элемента.
- SelectionMode: режим выбора (один или несколько элементов).
- MultiColumn: отображение в несколько колонок.

## **Основные методы:**

- `Items.Add()`: добавление элемента.
- `Items.Remove()`: удаление элемента.
- `Items.Clear()`: очистка списка.

## **Основные события:**

- `SelectedIndexChanged`: изменение выбранного элемента.
- `SelectedValueChanged`: изменение значения выбранного элемента.

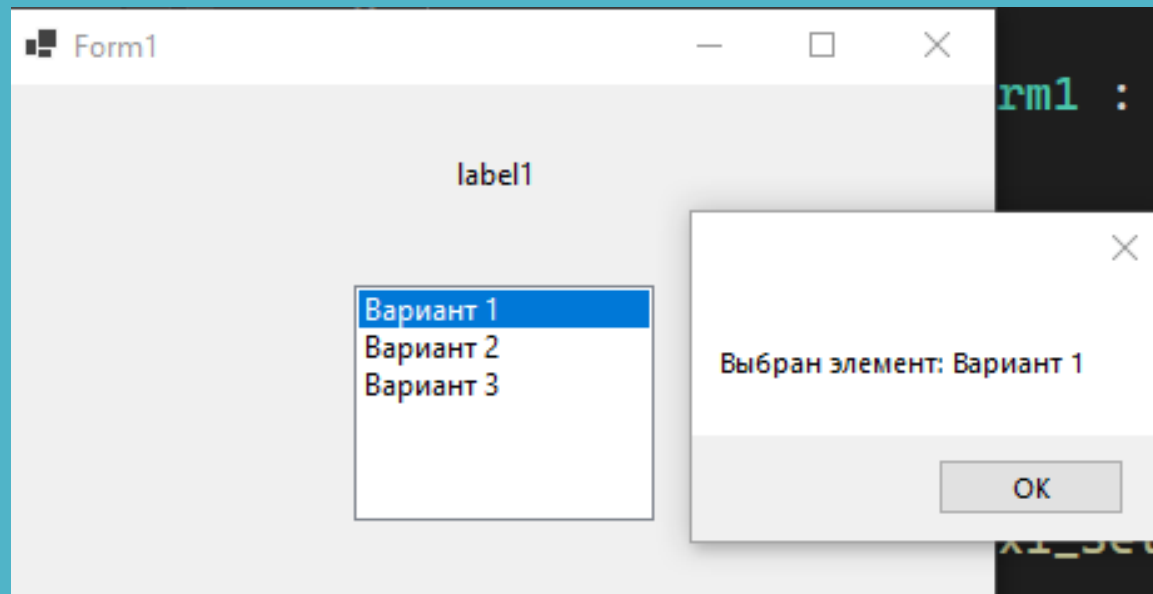
## **Пример использования:**

- Добавление элементов в список.
- Обработка выбора элемента.

# Пример обработки события SelectedIndexChanged (изменение выбранного элемента).

```
Ссылка: 1
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    // Проверяем, выбран ли элемент
    if (listBox1.SelectedItem != null)
    {
        // Получаем выбранный элемент
        string selectedItem = listBox1.SelectedItem.ToString();

        // Выводим сообщение с выбранным элементом
        MessageBox.Show("Выбран элемент: " + selectedItem);
    }
}
```



## 4. Комбинированные поля (ComboBox).

**Назначение и использование:**

- Выпадающий список для выбора одного элемента.

**Основные свойства:**

- Items: коллекция элементов.
- SelectedItem: выбранный элемент.
- SelectedIndex: индекс выбранного элемента.
- DropDownStyle: стиль списка (простой, выпадающий, выпадающий с редактированием).
- Text: текст в поле ввода.

## **Основные методы:**

- `Items.Add()`: добавление элемента.
- `Items.Remove()`: удаление элемента.
- `Items.Clear()`: очистка списка.

## **Основные события:**

- `SelectedIndexChanged`: изменение выбранного элемента.
- `TextChanged`: изменение текста в поле ввода (если `ComboBox` редактируемый).

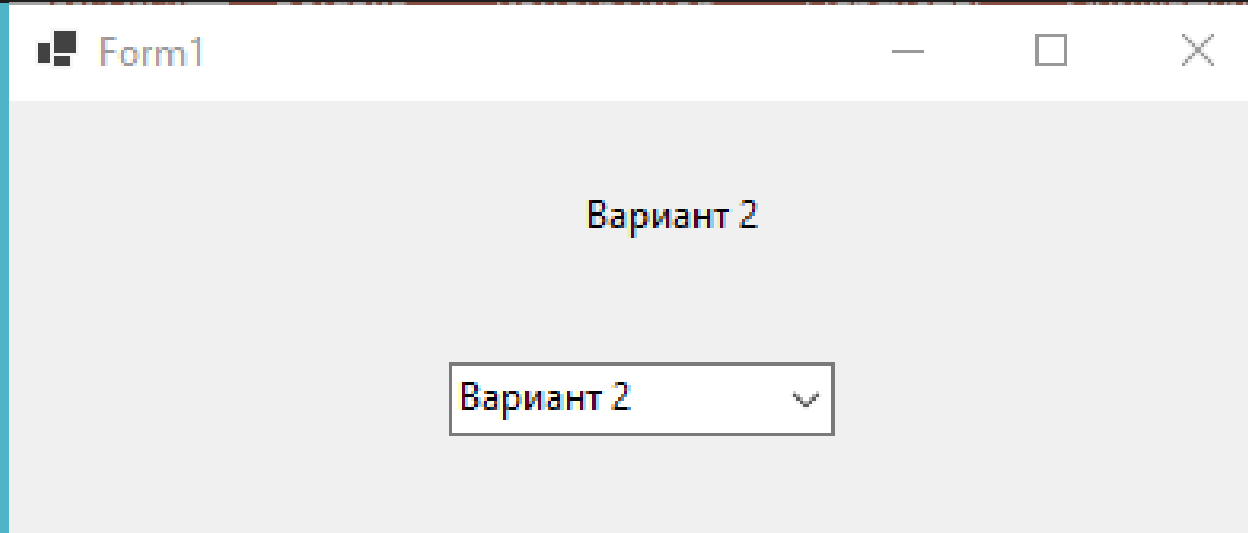
## **Пример использования:**

- Добавление элементов в `ComboBox`.
- Обработка выбора элемента.

# Пример обработки события SelectedIndexChanged (изменение выбранного элемента).

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    // Проверяем, выбран ли элемент
    if (comboBox1.SelectedItem != null)
    {
        // Получаем выбранный элемент
        string selectedItem = comboBox1.SelectedItem.ToString();

        // Выводим сообщение с выбранным элементом
        label1.Text = selectedItem;
    }
}
```



The screenshot shows a Windows application window titled "Form1". Inside the window, there is a label with the text "Вариант 2". Below the label is a dropdown menu (ComboBox) that also displays "Вариант 2" and has a downward arrow on its right side, indicating it is the selected item.

# 5. Элементы управления для работы с датой и временем.

## **DateTimePicker**

### **Назначение:**

DateTimePicker позволяет пользователю выбирать дату и/или время из выпадающего календаря или вводить их вручную.

Поддерживает различные форматы отображения даты и времени.



## Основные свойства:

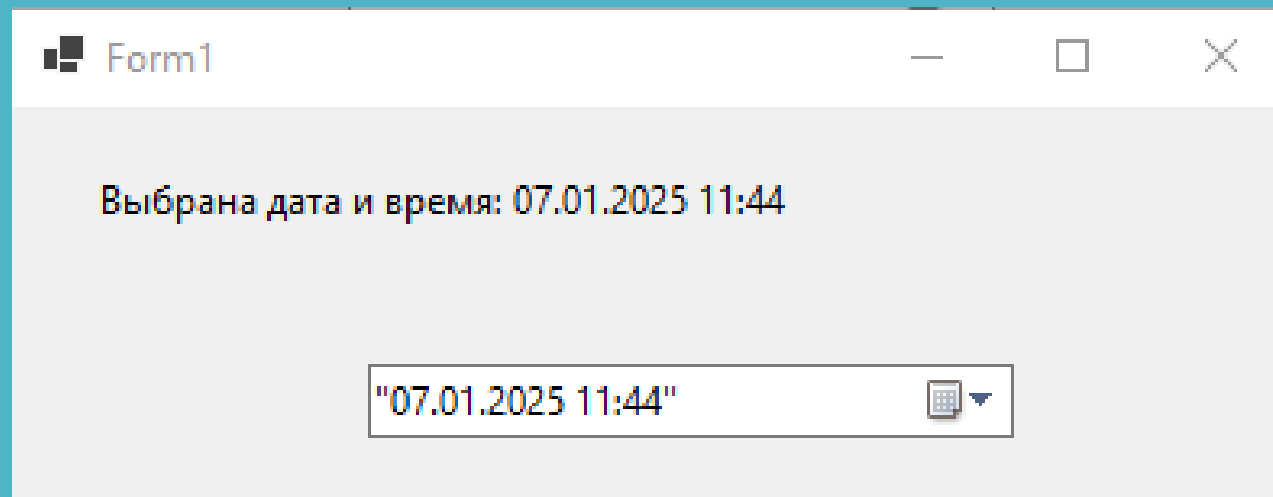
- Value: текущая выбранная дата и время.
- Format: формат отображения даты и времени (например, Short, Long, Time, Custom).
- CustomFormat: пользовательский формат (например, dd.MM.yyyy HH:mm).
- ShowUpDown: отображение стрелок для изменения значений (вместо выпадающего календаря).
- MinDate и MaxDate: минимальная и максимальная допустимая дата.

## Основные события:

- ValueChanged: происходит при изменении выбранной даты или времени.
- CloseUp: происходит при закрытии выпадающего календаря.

# Пример обработки события DateSelected (выбор даты/времени)

```
Семейство 1
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    DateTime selectedDateTime = dateTimePicker1.Value;
    label1.Text = "Выбрана дата и время: " + selectedDateTime.ToString("dd.MM.yyyy HH:mm");
}
```



The screenshot shows a Windows Form titled "Form1". Inside the form, a label displays the text "Выбрана дата и время: 07.01.2025 11:44". Below the label, there is a date and time picker control. The text box of the picker shows "07.01.2025 11:44", and to its right is a small icon of a calendar with a downward arrow, indicating a dropdown menu.

# MonthCalendar

## Назначение:

- MonthCalendar позволяет пользователю выбирать дату или диапазон дат из календаря.
- Подходит для выбора одной даты или диапазона дат (например, для бронирования).

## Основные свойства:

- `SelectionStart`: начальная дата выбранного диапазона.
- `SelectionEnd`: конечная дата выбранного диапазона.
- `SelectionRange`: выбранный диапазон дат.
- `MaxSelectionCount`: максимальное количество дней, которые можно выбрать.
- `MinDate` и `MaxDate`: минимальная и максимальная допустимая дата.

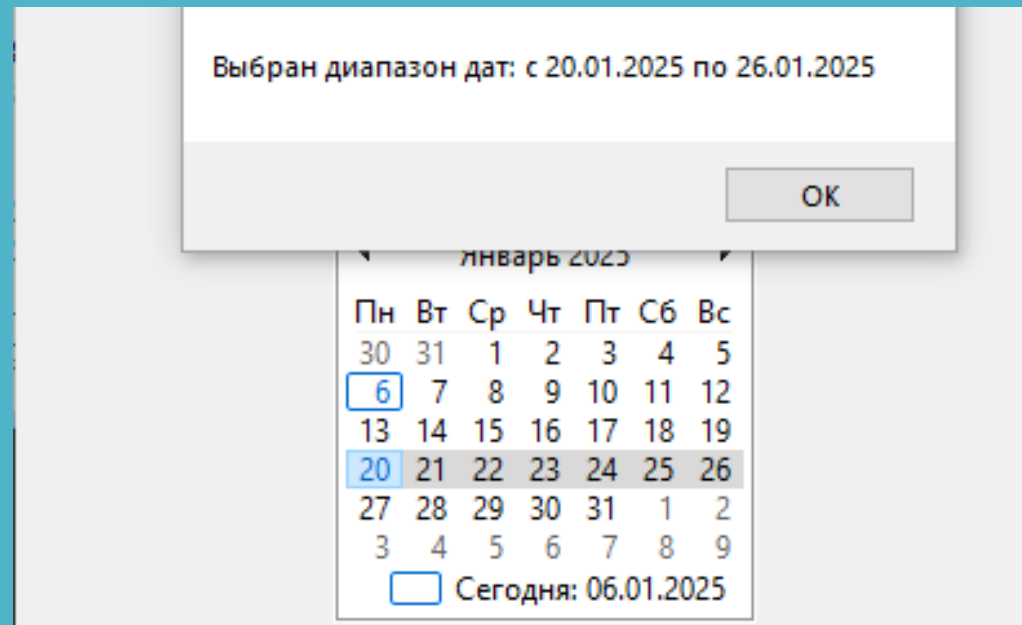
## Основные события:

- `DateChanged`: происходит при изменении выбранной даты.
- `DateSelected`: происходит при выборе даты пользователем.

# Пример обработки события DateSelected (выбор диапазона дат)

```
Ссылка: 1
private void monthCalendar1_DateSelected(object sender, DateRangeEventArgs e)
{
    DateTime startDate = monthCalendar1.SelectionStart;
    DateTime endDate = monthCalendar1.SelectionEnd;

    if (startDate == endDate)
        MessageBox.Show("Выбрана дата: " + startDate.ToString("dd.MM.yyyy"));
    else
        MessageBox.Show("Выбран диапазон дат: с " + startDate.ToString("dd.MM.yyyy")
            + " по " + endDate.ToString("dd.MM.yyyy"));
}
```



# Список литературы:

1. Жаксыбаева Н.Н. Основы объектно-ориентированного программирования: язык C#. Часть 2./ Учебное пособие предназначено для учащихся технического и профессионального образования, Алматы, 2010,
2. [Видеокурс C#.](#)

# **Материалы лекций:**

<https://github.com/ShViktor72/Education>

# **Обратная связь:**

[colledge20education23@gmail.com](mailto:colledge20education23@gmail.com)

# Домашнее задание:

## Задание 1. Работа с CheckBox и RadioButton

Создайте Windows Forms-приложение с формой, на которой:

- Разместите три CheckBox, имитирующих выбор дополнительных услуг (например, "Доставка", "Подарочная упаковка", "СМС-уведомление").
- Разместите группу из трех RadioButton, позволяющих выбрать способ оплаты ("Наличные", "Карта", "Перевод").
- Добавьте Label, который будет отображать выбранные опции при нажатии на кнопку "Показать выбор".

Что нужно реализовать:

- При изменении состояния CheckBox обновлять список выбранных услуг.
- Гарантировать, что в группе RadioButton можно выбрать только один вариант.
- Выводить информацию о выборе в Label при нажатии на кнопку.



## Задание 2. Работа с ListBox и ComboBox

Создайте форму, на которой:

- Разместите ListBox для отображения списка товаров.
- Добавьте текстовое поле (TextBox) и кнопку "Добавить", чтобы пользователь мог добавлять товары в ListBox.
- Добавьте ComboBox с предустановленными категориями товаров (например, "Электроника", "Одежда", "Продукты").

Что нужно реализовать:

- При нажатии кнопки "Добавить" товар должен добавляться в ListBox.
- При выборе категории в ComboBox список товаров должен фильтроваться по выбранной категории.

### Задание 3. Работа с DateTimePicker и MonthCalendar

Создайте приложение для бронирования встречи, в котором:

- Разместите DateTimePicker для выбора даты и времени встречи.
- Разместите MonthCalendar, позволяющий выбрать диапазон дат.
- Добавьте Label, отображающий выбранную дату и время.
- Добавьте кнопку "Подтвердить", при нажатии на которую выбранные параметры будут выводиться в MessageBox.

Что нужно реализовать:

- Позволить выбрать дату и время встречи через DateTimePicker.
- Позволить выбрать диапазон дат через MonthCalendar.
- При нажатии на кнопку "Подтвердить" отображать MessageBox с выбранными значениями.

## Задание 4.

Создайте небольшую анкету с элементами управления:

- Имя (TextBox).
- Дата рождения (DateTimePicker).
- Пол (RadioButton).
- Увлечения (несколько CheckBox).
- Кнопку "Отправить", которая будет отображать введенные данные в MessageBox.