

Лабораторная работа №30

Изучение особенностей элементов выбора в приложениях WPF

1 Цель работы

1.1 Изучить свойства и процесс обработки событий элементов выбора в приложениях WPF.

2 Литература

2.1 <https://metanit.com/sharp/wpf/> – гл.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

CheckBox – флажок. Особенности:

- при установке происходит событие Checked, при снятии – Unchecked.
- у событий Checked и Unchecked одинаковая сигнатура, т. е. можно использовать один обработчик на оба события.
- для проверки, что флажок установлен, используется IsChecked (может быть true, false, null) или флажок.IsChecked.Value (может быть true, false)
if (флажок.IsChecked == значение) { ... }
if (флажок.IsChecked.Value) { ... }

RadioButton – переключатель. Особенности:

- для объединения переключателей в группы используется свойство GroupName (между собой будут связаны переключатели, у которых значение GroupName одинаковое)
- Checked/Unchecked/IsChecked действуют так же, как у CheckBox.

5.1 Применение CheckBox для блокировки/разблокировки элементов управления

5.1.1 Создать WPF-приложение для подписки на рассылку.

5.1.2 Добавить на форму поля ввода для того, чтобы пользователь ввел логин и email и кнопку «Подписаться» (подписку реализовывать не нужно). Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести.

5.1.3 Добавить на форму CheckBox с подписью «Я согласен на получение рассылки». Если флажок отмечен, кнопка «Подписаться» должна быть доступной для нажатия, иначе – недоступной, но видимой. За доступность отвечает свойство Enabled.

5.2 Применение CheckBox для отображения/скрытия элементов управления

5.2.1 Создать WPF-приложение для авторизации пользователя.

5.2.2 Добавить на форму поля ввода для того, чтобы пользователь ввел логин и пароль и кнопку «Авторизоваться» (авторизацию реализовывать не нужно). Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести.

5.2.3 Добавить на форму CheckBox с подписью «Показать пароль». Если флажок отмечен, пароль должен отображаться, иначе – скрываться за звездочками. За видимость отвечает свойство Visibility:

- Visibility.Collapsed – элемент скрывается,
- Visibility.Visible – элемент отображается.

Для того, чтобы пароль можно было скрыть/показать нужно разместить рядом два поля ввода: PasswordBox (поле ввода с маской, пароль хранится в свойстве Password) и TextBox:

```
<PasswordBox Name="имя поля ввода пароля с маской" />
```

```
<TextBox Name="имя поля ввода пароля без маски" Visibility="Collapsed" />
```

Если флажок отмечен, скрывать PasswordBox, показывать TextBox. Тексту поля ввода TextBox присвоить значение пароля из PasswordBox.

Если флажок отмечен, скрывать TextBox, показывать PasswordBox. Паролю поля ввода PasswordBox присвоить значение текста из TextBox.

5.3 Применение RadioButton для смены размера текста

5.3.1 Создать WPF-приложение для ввода текста.

5.3.2 Добавить на форму поле ввода высотой 100 или более для того, чтобы пользователь мог ввести текстовые данные.

5.3.3 Добавить на форму набор переключателей с подписями 10, 20, 30 (подписи указываются в свойстве Content). Для всех переключателей создать **один обработчик** события Checked.

В обработчике привести sender к типу переключателя и получить числовое значение, приведя Content к числовому типу данных.

После получения числового значения присвоить его свойству FontSize поля ввода:
полеВвода.FontSize = значение;

5.4 Применение RadioButton для выравнивания текста

Добавить на форму из п.5.3 набор переключателей с подписями «По левому краю», «По центру», «По правому краю» (подписи указываются в свойстве Content). Для каждого переключателя создать свой обработчик события Checked, в котором в зависимости от выбранного переключателя изменять выравнивание текста на требуемое:

полеВвода.TextAlignment = TextAlignment.Значение;

5.5 Применение CheckBox для смены цвета фона

5.5.1 Создать WPF-приложение, на котором пользователь сможет настраивать цвет фона.

5.5.2 Добавить на форму три слайдера для того, чтобы пользователь мог указать число от 0 до 255 для цветовых составляющих (Red, Green, Blue). Снабдить слайдеры метками, чтобы пользователь понимал, что он должен указать.

5.5.3 Добавить на форму три CheckBox, каждый рядом со своим слайдером. Если флажок отмечен, учитывать цветовую составляющую, иначе вместо нее использовать 0.

Для всех CheckBox использовать один обработчик.

Для всех слайдеров использовать один обработчик.

Для указания цвета используется следующий код:

Background = new SolidColorBrush(Color.FromRgb(красный, зеленый, синий));

6Порядок выполнения работы

6.1 Выполнить все задания из п.5 в одном решении LabWork30. Каждый проект – приложение WPF.

6.2 Ответить на контрольные вопросы.

7Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8Контрольные вопросы

8.1 Что такое CheckBox и для чего он используется?

8.2 Что такое RadioButton и для чего он используется?

8.3 Как проверить, что флажок или переключатель выбран?

8.4 Какое событие срабатывает при выборе флажка или переключателя?

8.5 Какое событие срабатывает при снятии выбора флажка или переключателя?

8.6 Какие значения могут принимать флажки и переключатели?