

Практическая работа №6

Создание пользовательских элементов управления

1Цель работы

1.1 Научиться создавать пользовательские элементы управления в приложении WPF.

2Литература

2.1 <https://metanit.com/sharp/wpf/12.2.php> Создание и использование шаблонов

2.2 <https://wpf-tutorial.com/ru/501/usercontrols-customcontrols/создание-и-использование-usercontrol/>

3Задание

Все созданные элементы управления должны быть протестированы в основном окне программы

3.1 Применение шаблона ControlTemplate для настройки вида элемента управления

3.1.1 Создать шаблон для переключателей RadioButton в ресурсах:

- Border со скругленными краями,
- ContentControl с выравниваем по центру по горизонтали и вертикали.

3.1.2 Настроить TemplateBinding для связывания Background и Content со значениями переключателя, к которому применен шаблон.

3.1.3 Добавить в ControlTemplate триггеры, срабатывающие при:

- выбранном элементе (IsChecked=true),
- недоступности переключателя (IsEnabled=false).

Во всех триггерах должен меняться цвет фона, в триггере IsEnabled – дополнительно меняется цвет текста на серый.

3.1.4 Применить созданные шаблоны для создания бокового меню из пяти пунктов. Название выбранного пункта отобразить в заголовке окна.

3.2 Создание пользовательского элемента управления для отображения текущего времени.

3.2.1 Добавить в приложение пользовательский элемент управления WPF. Разместить в контейнере пользовательского элемента управления метку

3.2.2 Добавить в код элемента управления переменную или поле DispatcherTimer:

DispatcherTimer timer = new(DispatcherPriority.Render);

3.2.3 Настроить и запустить таймер:

timer.Interval = TimeSpan.FromSeconds(1); // интервал срабатывания
timer.Start();

3.2.4 Создать обработчик Tick таймера для смены текста в метке на текущее время в формате чч:мм:сс. Проверить работу таймера в основном окне.

3.3 Создание событий в пользовательском элементе управления

3.3.1 Добавить в приложение пользовательский элемент управления WPF.

Разместить в контейнере две кнопки: «+» и «-», между ними – поле ввода (для увеличения/уменьшения количества как в корзинах интернет-магазинов).

3.3.2 Добавить в пользовательский элемент управления:

- свойство MaxValue для хранения максимально допустимого значения,

- открытое событие ValueChanged для делегата типа RoutedEventHandler (для того, чтобы на него можно было подписаться в основной программе),
 - обработчики нажатий на кнопки +/-, вызывающие ValueChanged.
- 3.3.3 В основном окне использовать созданный элемент управления. Указать у него MaxValue = 5.

3.4 Добавление свойств зависимости в пользовательский элемент управления

3.4.1 Добавить в пользовательский элемент управления из п.3.3 свойство зависимости Value для хранения количества.

3.4.2 Реализовать изменение значения свойства Value на 1 при нажатии на кнопки:

- если значение Value=0, делать кнопку «-» недоступной.
- если значение Value=MaxValue, делать кнопку «+» недоступной.

3.4.3 При изменении количества в пользовательском элементе управления изменять значение стоимости в метке (вычислять как цена * количество, цена – поле ввода в окне).

3.5 Создать библиотеку для хранения пользовательских элементов управления и подключить пользовательские элементы из библиотеки в новом приложении.

Для создания выбрать тип проекта «Библиотека пользовательских элементов управления WPF».

4Порядок выполнения работы

4.1 Выполнить все задания из п.3.

4.2 Ответить на контрольные вопросы.

5Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

6Контрольные вопросы

6.1 Для чего применяется ControlTemplate?

6.2 Где может быть описан ControlTemplate?

6.3 Для чего применяется UserControl?

6.4 Каков алгоритм создания пользовательского элемента управления?

6.5 Как программно создать обработчик события?

6.6 Для чего используется свойство зависимости?