

Лабораторная работа №7

Изучение способов компоновки элементов в приложениях WPF

1 Цель работы

1.1 Изучить элементы-контейнеры, применяющиеся в приложениях WPF для компоновки.

2 Литература

2.1 <https://metanit.com/sharp/wpf/> – гл.3.

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

При разработке WPF-приложений используется XAML – язык разметки, используемый для инициализации объектов в технологиях на платформе .NET.

Синтаксис языка XAML похож на HTML и XML.

В тэгах описываются элементы управления, в атрибутах – свойства элементов управления.

Window (окно) – аналог тэга body в HTML – внутри него размещаются все остальные элементы управления.

5.1 Применение контейнера Grid (сетки) для создания формы авторизации

5.1.1 Создать приложение WPF.

5.1.2 В открывшемся окне, отображающем внешний вид окна (Конструктор) и разметку (XAML) изменить содержимое элемента Grid:

- поставить курсор между тэгами **<Grid>** и **</Grid>**,
- во вкладке Свойства в разделе Макет нажать на «...» у **ColumnDefinitions** (определение столбцов) и добавить 2 столбца,
- во вкладке Свойства в разделе Макет нажать на «...» у **RowDefinitions** (определение строк) и добавить 3 строки.

5.1.3 Разместить на сетке элементы управления:

- перенести две метки с текстом Логин и Пароль в ячейки левого столбца сетки (подписи меток меняются в свойстве **Content**)
- перенести поля ввода для указания логина и пароля в ячейки правого столбца сетки
- перенести кнопку с подписью «Авторизоваться» в нижнюю строку (подпись кнопки меняется в свойстве **Content**).

5.1.4 Выровнять элементы управления по сетке:

- убрать из тэгов **Label**, **TextBox**, **Button** все свойства (пишутся как атрибуты в html и xml) кроме **Content** (подписи элементов, видимые пользователю)
- используя вкладку Свойства или разметку XAML, указать, в каких строках (**Grid.Row**) и столбцах (**Grid.Column**) находятся элементы (нумерация строк и столбцов с нуля)

- используя вкладку Свойства или разметку XAML, указать, что кнопка занимает 2 столбца (**Grid.ColumnSpan**)

5.1.5 Автоматическое вычисление ширины и высоты ячеек

- у все тэгов RowDefinition указать Height="Auto" для того, чтобы высота строк соответствовала высоте элементов управления в строках

- у левого столбца внутри тэга ColumnDefinition указать Width="Auto" для того, чтобы ширина столбца соответствовала ширине элементов управления в столбцах

5.2 Применение контейнера StackPanel (панели, в которой элементы управления отображаются друг за другом по горизонтали или вертикали) для отображения списка кнопок для перехода к другим окнам

5.2.1 Создать приложение WPF.

5.2.2 В открывшемся окне, отображающем внешний вид окна (Конструктор) и разметку (XAML) вместо элемента Grid написать StackPanel (закрывающийся тэг изменится автоматически).

5.2.3 Разместить в StackPanel кнопки со следующими подписями:

- Производители,
- Товары,
- Заказчики,
- Заказы.

5.2.4 У каждой кнопки указать минимальную ширину 200, а у StackPanel – выравнивание по горизонтали (HorizontalAlignment) и выравнивание по вертикали (VerticalAlignment) – по центру.

5.2.5 Сменить у StackPanel ориентацию (Orientation) на горизонтальную, выравнивание по горизонтали убрать, по вертикали – по верхнему краю. Сравнить результат с предыдущим (в п.5.2.4).

5.3 Применение контейнера GridSplitter (две панели с разделителем) для создания интерфейса, аналогичного проводнику.

5.3.1 Создать приложение WPF.

5.3.2 В открывшемся окне, отображающем внешний вид окна (Конструктор) и разметку (XAML) изменить содержимое элемента Grid:

- поставить курсор между тэгами <Grid> и </Grid>,
- добавить в сетку три столбца, ширина центрального столбца – Auto, ширина левого – 0.5*, ширина правого - * (т.е., ширина пропорциональна и у левого – 50% от ширины правого).

5.3.3 Разместить на сетке элементы управления:

- перенести в левый столбец элемент TreeView,
- перенести в правый столбец элемент ListView,
- перенести в центральный столбец элемент GridSplitter.

У GridSplitter указать следующие настройки:

- выравнивание по горизонтали: по центру,
- выравнивание по вертикали: растянуть (Stretch),
- ширина – 3.

5.4 Применение контейнера DockPanel (панели с привязкой к краям окна) для создания интерфейса, аналогичного проводнику.

5.4.1 Создать приложение WPF.

5.4.2 В открывшемся окне, отображающем внешний вид окна (Конструктор) и разметку (XAML) вместо элемента Grid написать DockPanel (закрывающийся тэг изменится автоматически).

5.4.3 Разместить в DockPanel метки со следующими подписями:

- Шапка,
- Меню,
- Контент,
- Подвал.

Цвет меток (Background) сделать различным.

Метка «Шапка» должна быть выровнена по верхнему краю, «Подвал» - по нижнему, «Меню» - в области между шапкой и подвалом с выравниванием по левому краю, «Контент» - вся оставшаяся область.

5.5 Применение контейнера WrapPanel для отображения меню.

Вместо метки «Шапка» в п.5.4 разместить в DockPanel WrapPanel с кнопками, совпадающими с кнопками из п.5.2.

6Порядок выполнения работы

6.1 Выполнить все задания из п.3 в одном решении LabWork7. 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 «компоновка» в WPF?

8.2 Какой класс является родительским для всех элементов-контейнеров в WPF?

8.3 Как выравнивать элементы внутри контейнера по высоте и по ширине?

8.4 В чем особенность компоновки с использованием следующих элементов-контейнеров: Grid и Canvas?

8.5 Чем отличается компоновка с использованием StackPanel, DockPanel, WrapPanel?

8.6 В каких единицах измерения могут задаваться размеры элементов в приложениях WPF?