**ПРИМЕР РАЗРАБОТКИ ПРИЛОЖЕНИЙ WPF**

# 

# ПРИМЕР 1. Простейшее приложение

Разработаем простейший пример, содержащий две кнопки.

## Порядок выполнения

1. Создайте новый проект *Приложения WPF* на языке *Visual C#* с именем *WpfApp1* (рис. 1).

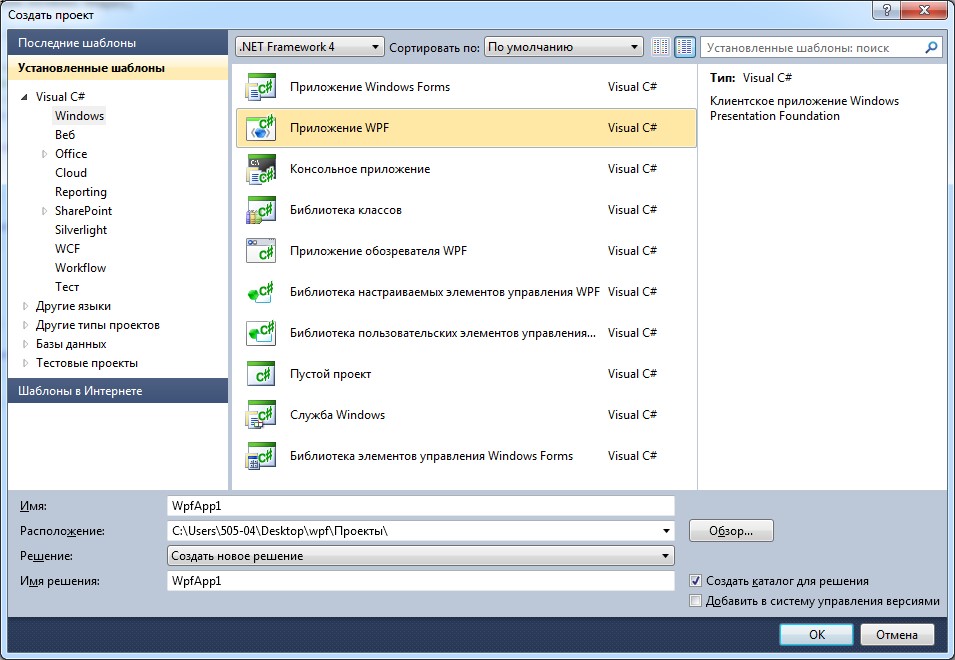


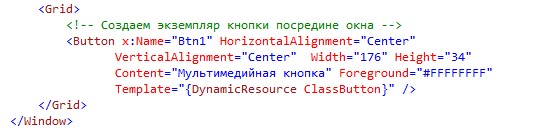
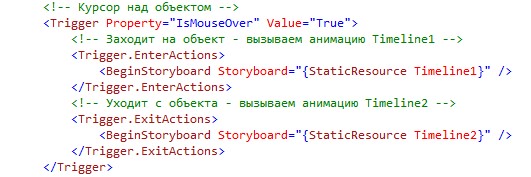
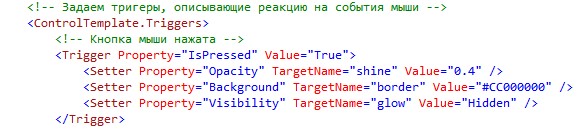
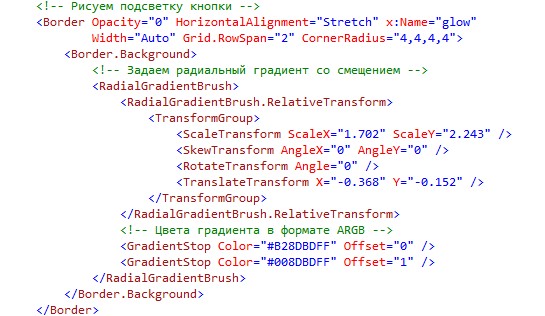
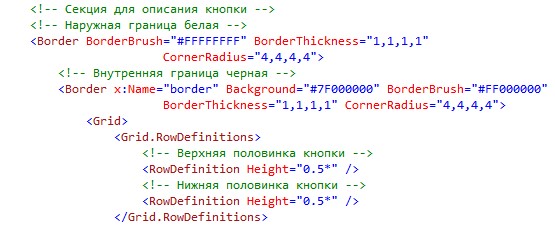
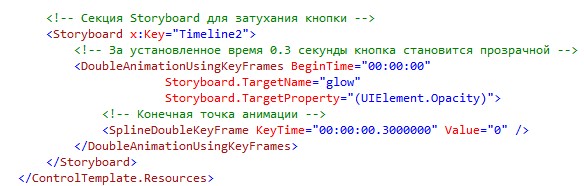
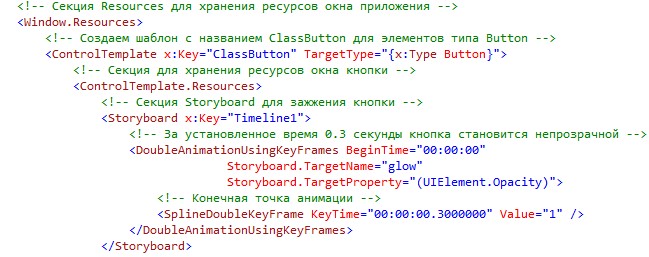
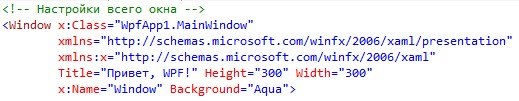
Рисунок 1 – Создание нового проекта WpfApp1

1. Ознакомьтесь с рабочей областью и заготовкой, созданной оболочкой.

Оболочка предоставляет возможность графического и дескрипторного способов разработки пользовательского интерфейса, которые являются равнозначными. На данном этапе нас интересуют файлы: дескрипторный *MainWindow.xaml* и кодовый *MainWindow.xaml.cs*, которые дополняют друг друга при описании одного и того же содержимого – класса *MainWindow* (по умолчанию) в пространстве имен *WpfApp1*.

Глядя на заготовку дескрипторного XAML-кода можно увидеть, что корнем приложения является контейнер *<Window>*, в который при развитии проекта будут включены дочерние элементы.

1. Для неопытных пользователей имеет смысл не вмешиваться в код XAML и создавать интерфейс формы по аналогии с обычными Windowsприложениями. Мы же, в демонстрационных целях, с помощью дескрипторного кода создадим простое изображение мультимедийной кнопки. Заполните вкладку XAML следующим кодом. Обратите внимание, что каждый дескриптор имеет свой тип и атрибуты, уточняющие его настройку. Кроме того, язык XAML чувствителен к регистру символов, поэтому следуйте указаниям подсказчика *IntalliSense*.



1. Запустите приложение. Посмотрите, что получается.
2. Выберите команду *Вид* – *Другие окна* – *Структура документа*.

Сравните представленную структуру с введенным ранее кодом XAML. Нажимая на различны элементы в структуре, обратите внимание, что соответствующий код XAML подсвечивается (рис. 2).

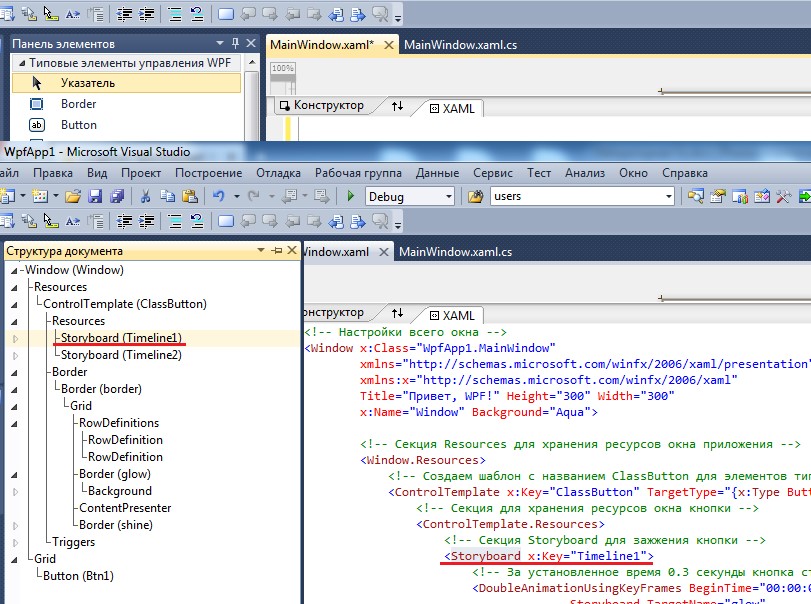
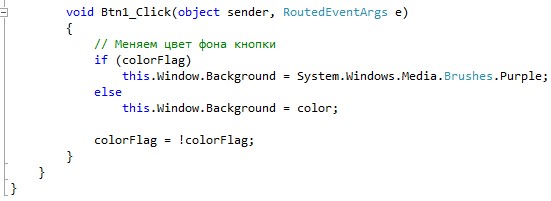
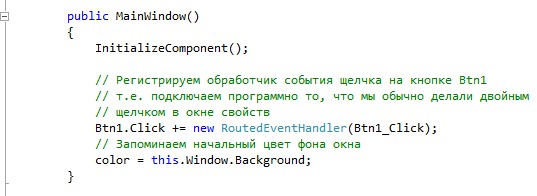
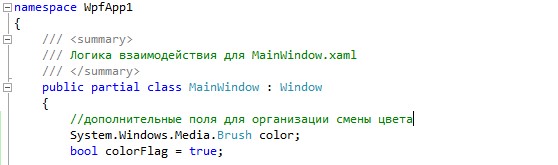
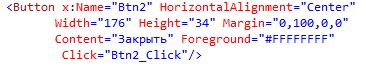


Рисунок 2 – Структура документа

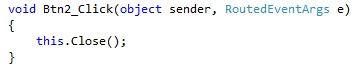
1. С помощью панели *Обозреватель решений* (*Solution Explorer*) откройте файл *MainWindow.xaml.cs* и дополните его следующим кодом:



1. Запустите приложение – теперь оно стало реагировать на щелчок мышью по кнопке и менять цвет клиентской области окна.
2. Добавим еще одну кнопку, которая будет закрывать форму. Добавьте между тегами *<Grid>* и *</Grid>* файла *MainWindow.xaml* следующий код, добавляющий обычную кнопку. Обратите внимание на атрибут *Margin* – добавляет пустое пространство вокруг элемента, а также на *Click* – сразу определяем обработчик соответствующего события кнопки:



1. Теперь откройте файл *MainWindow.xaml.cs* и допишите в него уже определенный выше обработчик для второй кнопки:



1. Готово. Запускайте, проверяйте.

# 

# ПРИМЕР 2. Приложение с несколькими окнами

## Постановка задачи

Разработаем приложение, которое будет иметь список персоналий на одной странице, и при выборе одного из них – открывать информацию о нем на другой.

## Порядок выполнения

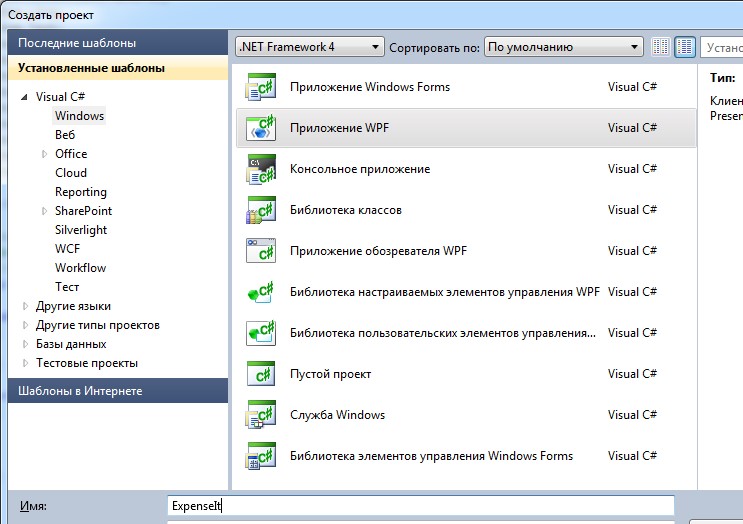
1. Создайте новый проект *Приложение WPF* на языке *Visual C#* с именем *ExpenseIt* (рис. 3). 

Рисунок 3 – Создание нового проекта ExpenseIt

1. Откройте файл *App.xaml*, который определяет приложение WPF и все его ресурсы. Посмотрите, как выглядит этот файл.



1. Откройте файл *MainWindow.xaml*, который представляет главное окно приложения, отображающий созданное содержимое страниц. Класс [*Window*](http://msdn.microsoft.com/ru-ru/library/system.windows.window(v=vs.110).aspx) определяет свойства окна, такие как заголовок, размер и значок, и обрабатывает события, такие как открытие и закрытие окна. Посмотрите, как выглядит этот файл.

1. Замените в файле *MainWindow.xaml* элемент [*Window*](http://msdn.microsoft.com/ru-ru/library/system.windows.window(v=vs.110).aspx) на [*NavigationWindow*.](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx) Приложение будет поддерживать переходы в стиле вебобозревателя. При этом класс [*NavigationWindow*](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx) наследует все свойства класса [*Window*.](http://msdn.microsoft.com/ru-ru/library/system.windows.window(v=vs.110).aspx)

В страничных приложениях содержимое страниц встраивается в специальный навигационный каркас, поддерживающий навигационные ссылки и журнал навигации. Корневым узлом в них является не класс *Window*, а производный от него класс *NavigationWindow*, который добавляет стандартный интерфейс навигации и всю необходимую инфраструктуру.

1. Измените следующие свойства элемента [*NavigationWindow*.](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx)
2. Задайте свойству [*Title*](http://msdn.microsoft.com/ru-ru/library/system.windows.window.title(v=vs.110).aspx) значение "*ExpenseIt*".
3. Задайте свойству [*Width*](http://msdn.microsoft.com/ru-ru/library/system.windows.frameworkelement.width(v=vs.110).aspx) значение 500 пикселей.
4. Задайте свойству [*Height*](http://msdn.microsoft.com/ru-ru/library/system.windows.frameworkelement.height(v=vs.110).aspx) значение 350 пикселей.
5. Удалите элементы [Grid](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) между тегами [*NavigationWindow*.](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx) В итоге XAML-код должен выглядеть следующим образом:



1. Откройте файл *MainWindow.xaml.cs*. Это файл программного кода, в котором содержится код обработки событий, объявленных в файле *MainWindow.xaml*. В этом файле содержится разделяемый класс окна, определенного в XAML-коде.

1. Измените класс *MainWindow* так, чтобы он наследовался от класса [*NavigationWindow*.](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx)



1. Запустите проект. Убедитесь, что в нем нет ошибок, что открывается пустое окно.

### 9. Добавьте в проект новую *Страницу (WPF)* с именем

*ExpenseItHome.xaml*. Эта страница отображается первой при запуске приложения; на ней будет выводится список, в котором можно выбрать человека и просмотреть отчет о его затратах (рис. 4).

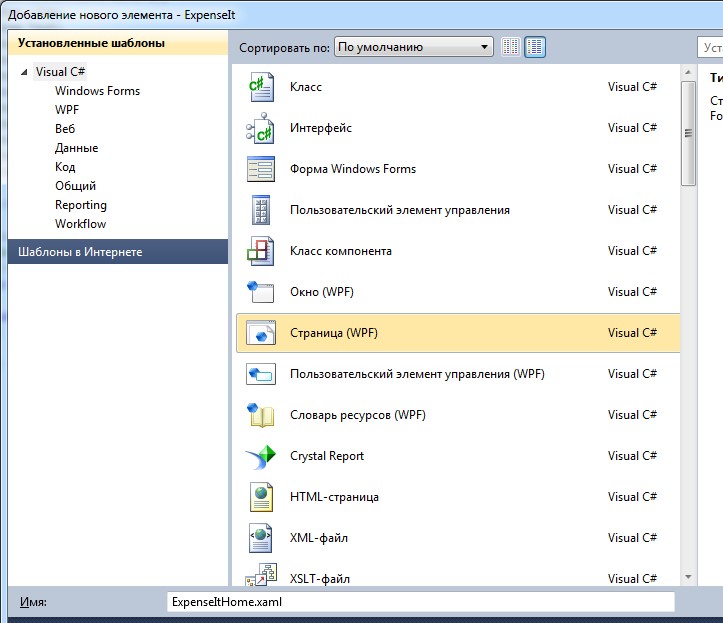


Рисунок 4 – Добавление новой страницы

### 10. Откройте файл *ExpenseItHome.xaml*. Задайте свойству [*Title*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.page.title(v=vs.110).aspx) значе-

ние "*ExpenseIt - Home*".



11. Откройте файл *MainWindow.xaml*. Задайте свойству [*Source*](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow.source(v=vs.110).aspx) объекта [*NavigationWindow*](http://msdn.microsoft.com/ru-ru/library/system.windows.navigation.navigationwindow(v=vs.110).aspx) значение "*ExpenseItHome.xaml*".



### 12. Добавьте в проект новую *Страницу (WPF)* с именем

*ExpenseReportPage.xaml*. На этой странице будет представлен отчет о расходах для человека, выбранного на странице *ExpenseItHome.xaml*.

1. Откройте файл *ExpenseReportPage.xaml*. Задайте свойству [*Title*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.page.title(v=vs.110).aspx) значение "*ExpenseIt - View Expense*".

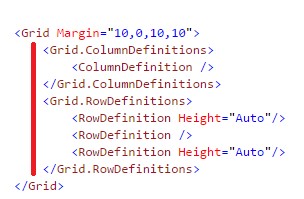


1. Добавьте в проект изображение с именем *watermark.png*. Можно создать собственное изображение или скопировать файл из образца кода (*Добавить – Существующий элемент*).
2. Запустите приложение. Убедитесь, что все в порядке, что в коде нет ошибок.

1. Далее создадим макет страницы *ExpenseItHome*. Откройте файл *ExpenseItHome.xaml*.

1. Задайте свойству [*Margin*](http://msdn.microsoft.com/ru-ru/library/system.windows.frameworkelement.margin(v=vs.110).aspx) элемента [*Grid*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) значение "*10,0,10,10*", указывающее ширину полей слева, сверху, справа и снизу соответственно.

1. Добавьте следующий XAML-код между тегами [*Grid*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) чтобы создать определения строк и столбцов. Обратите внимание, что свойству [*Height*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.rowdefinition.height(v=vs.110).aspx) двух строк задано значение [*Auto*,](http://msdn.microsoft.com/ru-ru/library/system.windows.gridlength.auto(v=vs.110).aspx) что означает, что размер строк будет зависеть от их содержимого.



1. Далее будем добавлять на форму элементы управления *ListBox*,

*Label* и *Button* так, как показано на рисунке. Откройте файл

*ExpenseItHome.xaml* (рис. 5).

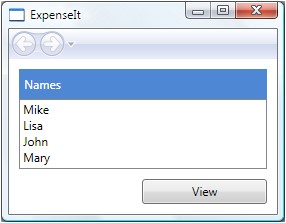
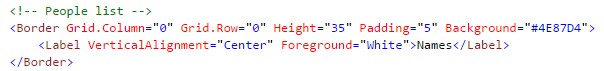


Рисунок 5 – Вид формы

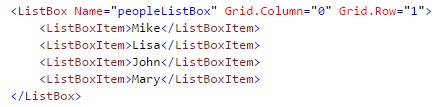
1. Добавьте следующий XAML-код между тегами [*Grid*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) для вставки

метки *Label*.



1. Далее добавьте следующий XAML-код для вставки списка

*ListBox*.



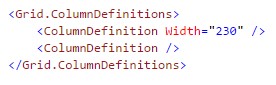
1. Наконец, следом добавьте следующий XAML-код для вставки

кнопки *Button*.

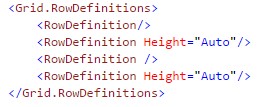


1. Запустите приложение. Проверьте, что получается.

1. Далее на страницу *ExpenseItHome.xaml* добавим изображение и заголовок страницы. Добавьте столбец в свойство [*ColumnDefinitions*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid.columndefinitions(v=vs.110).aspx) с фиксированной длиной [*Width*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.columndefinition.width(v=vs.110).aspx) *230* пикселей (вставить это нужно между тегами *<Grid>* и *</Grid>*, например – после кода, которым мы вставляли кнопку):

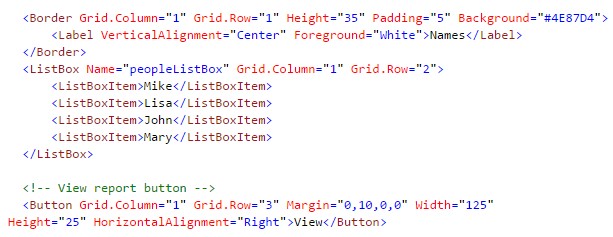


1. Далее добавьте строку в свойство [*RowDefinitions*:](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid.rowdefinitions(v=vs.110).aspx)



1. Теперь задайте свойству [*Grid.Column*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid.column(v=vs.110).aspx) вставленных ранее элементов *Label*, *ListBox* и *Button* значение *1* – таким образом Вы переместите их во второй столбец.

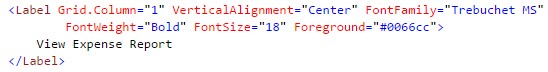
1. Аналогично переместите каждый из указанных в предыдущем пункте элементов управления на одну строку вниз – для этого увеличьте значение свойства [*Grid.Row*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid.row(v=vs.110).aspx) на *1*.



1. Добавьте классу [*Grid*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) свойство [*Background*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.panel.background(v=vs.110).aspx) задав в качестве его значения файл *watermark.png*. Для этого вставьте между тегами *<Grid>* и *</Grid>* следующий код:



1. В заголовке страницы перед элементом [*Border*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.border(v=vs.110).aspx) добавьте элемент [*Label*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.label(v=vs.110).aspx) с содержимым "*View Expense Report*":



### 30. Запустите приложение. Проверьте, что все работает без ошибок

(рис. 6).

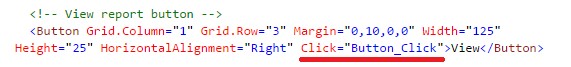


Рисунок 6 – Вид приложения

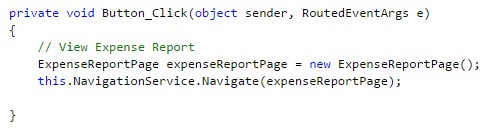
1. На следующем этапе добавим код для обработки событий, а именно: при нажатии на кнопку будем переходить к файлу

*ExpenseReportPage.xaml*. Откройте файл *ExpenseItHome.xaml*.

1. Добавьте обработчик событий [*Click*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.primitives.buttonbase.click(v=vs.110).aspx) в элемент [*Button*.](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.button(v=vs.110).aspx)

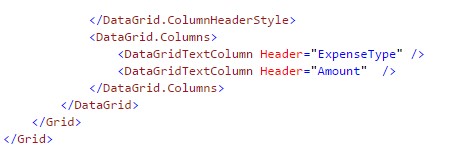
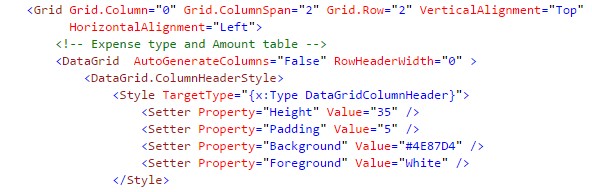
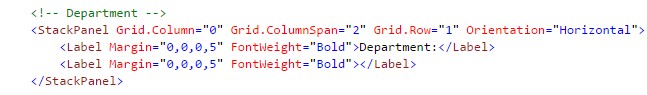
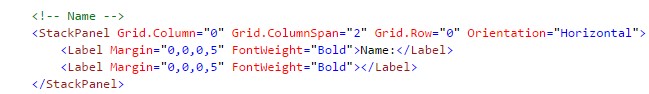
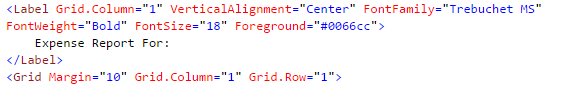
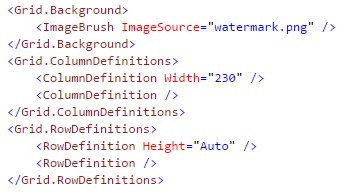


1. Откройте файл *ExpenseItHome.xaml.cs* и добавьте в него следующую реализацию указанного обработчика события [*Click*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.primitives.buttonbase.click(v=vs.110).aspx) обеспечивающий переход окна к файлу *ExpenseReportPage.xaml*:



1. Теперь займемся страницей *ExpenseReportPage.xaml*, где будет отображаться отчет о расходах для человека, выбранного на странице *ExpenseItHome.xaml*. Откройте файл *ExpenseReportPage.xaml*.

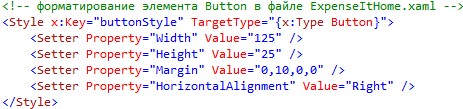
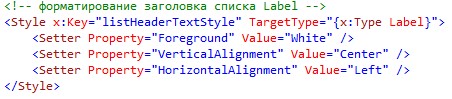
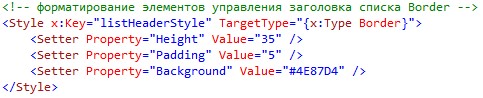
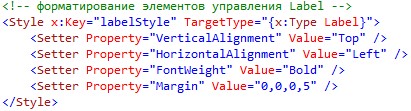
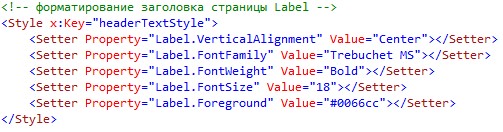
1. Внимательно ознакомьтесь со следующим XAML-кодом и вставьте его между тегами [*Grid*.](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) С помощью данного кода на страницу будут добавлены элементы управления, создан пользовательский интерфейс, установлен фон и цвета заливки для различных элементов Пользовательского интерфейса:



1. Запустите приложение. Нажмите на кнопку *View*, убедитесь, что теперь все работает как нужно.

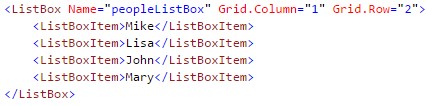
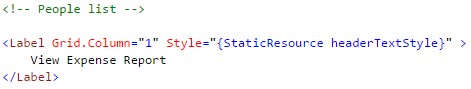
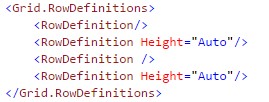
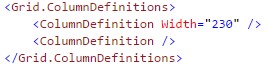
1. Т.к. пользовательский интерфейс часто подразумевает одинаковый внешний вид для всех элементов одного типа, то будем использовать стили. Повторное использование стилей помогает упростить создание XAML-кода и управление им. Заменим стилями атрибуты, установленные ранее для каждого элемента. Откройте файл *App.xaml*.

1. Внимательно ознакомьтесь со следующим XAML-кодом, задающим стили оформления для различных элементов, и вставьте его между тегами [*Application.Resources*.](http://msdn.microsoft.com/ru-ru/library/system.windows.application.resources(v=vs.110).aspx)

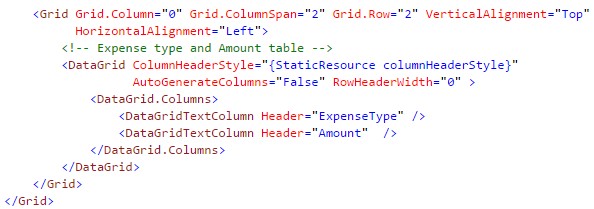
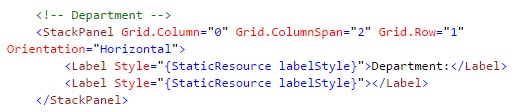
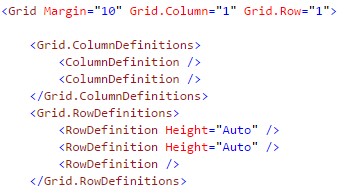
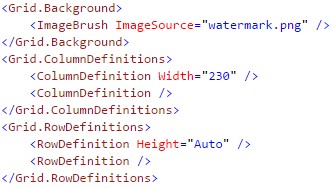


1. Откройте файл *ExpenseItHome.xaml*. Отредактируйте содержимое между элементами [*Grid*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) записанное нами ранее, таким образом, чтобы он принял следующим вид (свойства, определяющие внешний вид элементов управления, удаляются, а вместо них применяются стили):





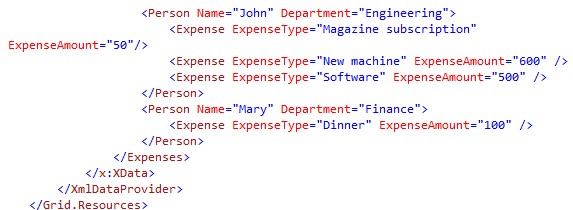
1. Откройте файл *ExpenseReportPage.xaml*. Отредактируйте содержимое между элементами [*Grid*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) записанное нами ранее, таким образом, чтобы он принял следующим вид (добавляем стили в элементы [*Label*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.label(v=vs.110).aspx) и [*Border*)](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.border(v=vs.110).aspx):



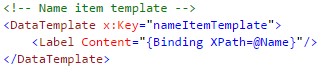
1. Запустите приложение. Посмотрите, что получается.

1. Далее создадим XML-данные, привязанные к различным элементам управления. Откройте файл *ExpenseItHome.xaml*.

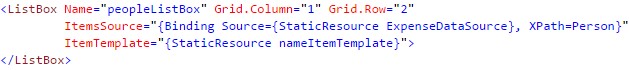
1. Добавьте следующий XAML-код в начало блока Grid для создания объекта [*XmlDataProvider*,](http://msdn.microsoft.com/ru-ru/library/system.windows.data.xmldataprovider(v=vs.110).aspx) содержащего данные по каждому человеку. Данные создаются в качестве ресурса объекта [*Grid*.](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) Обычно такие ресурсы загружаются в виде файла, но в данном случае данные просто добавлены в код.



1. Также в ресурс [*Grid*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) после тега *</XmlDataProvider>* но до *</Grid.Resources>*, добавьте следующий элемент [*DataTemplate*,](http://msdn.microsoft.com/ru-ru/library/system.windows.datatemplate(v=vs.110).aspx) определяющий способ отображения данных в списке [*ListBox*:](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.listbox(v=vs.110).aspx)

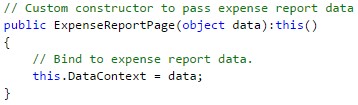


1. Наконец, отредактируйте существующий класс [*ListBox*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.listbox(v=vs.110).aspx) (элемент *peopleListBox*) следующим образом (обеспечиваем привязку свойства [ItemsSource](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.itemscontrol.itemssource(v=vs.110).aspx) элемента [*ListBox*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.listbox(v=vs.110).aspx) к источнику данных и применяем шаблон источника данных как элемент [*ItemTemplate*)](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.itemscontrol.itemtemplate(v=vs.110).aspx):

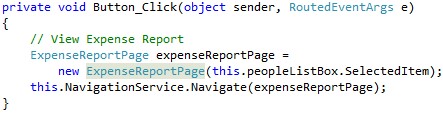


1. Запустите, проверьте.

1. Далее запишем код, извлекающий текущий пункт, выбранный в списке людей на странице *ExpenseItHome.xaml*, и передающий его ссылку конструктору элемента *ExpenseReportPage* во время создания объекта. Откройте файл *ExpenseReportPage.xaml.cs* и добавьте в него конструктор, принимающий объект, чтобы можно было передать данные отчета о затратах выбранного человека:

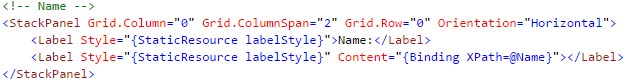


1. Откройте файл *ExpenseItHome.xaml.cs* и измените обработчик событий [*Click*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.primitives.buttonbase.click(v=vs.110).aspx) так, чтобы он вызывал новый конструктор и передавал ему данные отчета о затратах выбранного человека:

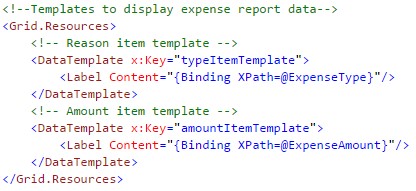


1. Далее займемся обновлением пользовательского интерфейса каждого элемента в списках с привязкой данных с помощью шаблонов данных. Откройте файл *ExpenseReportPage.xaml*.

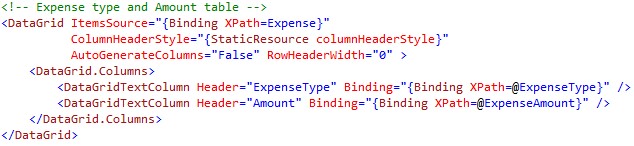
1. Привяжите содержимое "*Name*" и "*Department*" элементов [*Label*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.label(v=vs.110).aspx) к соответствующему свойству источника данных, добавив к существующему коду атрибут *Content*:



1. После открывающего элемента [*Grid*](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.grid(v=vs.110).aspx) добавьте следующие шаблоны данных, определяющие способ отображения отчета о затратах:



1. Примените шаблоны к столбцам [*DataGrid*,](http://msdn.microsoft.com/ru-ru/library/system.windows.controls.datagrid(v=vs.110).aspx) в которых отображаются данные отчета о затратах, дополнив его код до следующего вида:



1. Готово! Запустите, проверьте. Теперь, при нажатии на кнопку *View*, откроются данные выбранного человека (рис. 7).



Рисунок 7 – Вид готового приложения