Ресурсы (стили, шаблоны, кисти и др.) могут располагаться не только в секции ресурсов элемента (например <Window.Resources> …). Так же ресурсы можно выносить в отдельные XAML-файлы: словари ресурсов. Чтобы добавить в проект словарь ресурсов, нужно выбрать в меню проекта: «добавить» - «Создать элемент» - Словарь ресурсов (WPF). В проект добавится XAML-файл, в который можно включать любые ресурсы. Чтобы использовать ресурсы из словаря в окне, нужно добавить следующий код:

<Window.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Theme.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Window.Resources>

Теперь внутри окна можно использовать любые ресурсы из данного словаря.

Если помимо ресурсов из словаря мы хотим добавить в окно собственные ресурсы, их нужно внести в секцию ResourceDictionary:

<Window.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Theme.xaml"/>

</ResourceDictionary.MergedDictionaries>

<Style TargetType="...">

...

</Style>

</ResourceDictionary>

</Window.Resources>

## Динамические ресурсы

Механизм динамических ресурсов позволяет менять тему приложения на ходу.

Предположим, требуется сделать две темы приложения с разным дизайном кнопок. Добавляем в проект два словаря ресурсов (например Theme1.xaml и Theme2.xaml). В оба словаря добавляем стили с одинаковыми ключами:

<Style TargetType="Button" x:Key="ButtonStyle">

...

Для кнопок, у которых мы хотим менять внешний вид, в качестве значения атрибута Style задаём не статический, а динамический ресурс:

<Button Content="Некая кнопка" Style="{DynamicResource ButtonStyle}"/>

Теперь можно загрузить другой словарь ресурсов, тогда стиль кнопки изменится в соответствии со значением ресурса ButtonStyle. Сделать это можно из CS-кода:

var dict = new ResourceDictionary();

// Загружаем словарь ресурсов

string uriStr = "/DynamicRes;component/Theme1.xaml";

dict.Source = new Uri(uriStr, UriKind.RelativeOrAbsolute);

// Очищаем предыдущий списк словарей

\_themeTab.Resources.MergedDictionaries.Clear();

// Добавляем загруженный словарь

\_themeTab.Resources.MergedDictionaries.Add(dict);

После выполнения данного кода все элементы, использующие динамические ресурсы с ключами из данного словаря будут обновлены.

## Строковые ресурсы

В качестве ресурсов могут использоваться обычные строки. Поскольку тип string определён в пространстве имён System, нужно добавить его в корневой элемент XAML:

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:system="clr-namespace:System;assembly=mscorlib">

Теперь можно использовать строковые ресурсы:

<system:String x:Key="Yellow">Yellow</system:String>

<system:String x:Key="Red">Red</system:String>

Использовать строковые ресурсы можно так же, как и любые другие. Если значения строк менять не требуется, подойдёт статический ресурс:

<TextBlock Foreground="Red" Text="{StaticResource Red}"/>

<TextBlock Foreground="#C0A000" Text="{StaticResource Yellow}"/>

Если же нужно менять строковые значения (например, для поддержки языков), то требуются динамические ресурсы:

<TextBlock Foreground="Red" Text="{DynamicResource Red}"/>

<TextBlock Foreground="#C0A000" Text="{DynamicResource Yellow}"/>

## Задание

1. В работе №8 были написаны стили и шаблоны для круглых кнопок. Нужно вынести данные ресурсы в словарь ресурсов. Создать ещё один словарь с такими же ключами ресурсов, но другим внешним видом кнопок (прямоугольная форма, цвета – на своё усмотрение). Сделать функцию выбора темы для кнопок.
2. Доработать графический редактор из прошлых работ: вынести все строки в словарь, добавить поддержку ещё одного языка, помимо русского, сделать функционал выбора языка).