***Стиль*** ‑ это коллекция значений свойств, которые могут быть применены к элементу. В WPF стили играют ту же роль, которую CSS играет в HTML-разметке. Подобно CSS, стили WPF позволяют определять общий набор характеристик форматирования и применять их по всему приложению для обеспечения согласованности. Стили могут работать автоматически, предназначаться для элементов конкретного типа и каскадироваться через дерево элементов.

Любой стиль – это объект класса System.Windows.Style. Основным свойством стиля является коллекция Setters, в которой каждый элемент задаёт значение для некоторого свойства зависимостей (стили не работают с обычными свойствами – только со свойствами зависимостей). В разметке стиля дочерние элементы с именем <Setter> автоматически помещаются в коллекцию Setters. Свойство стиля TargetType позволяет указать конкретный тип, к которому применяется стиль. В этом случае префикс в установщиках свойств использовать необязательно:

<Style x:Key="baseStyle" TargetType="{x:Type Button}">

<Setter Property="FontSize" Value="22" />

<Setter Property="Background" Value="Purple" />

<Setter Property="Foreground" Value="White" />

</Style>

Определяя стиль, можно построить его на основе стиля-предка. Для этого следует воспользоваться свойством стиля BasedOn. Стили также могут содержать локальные логические ресурсы (коллекция Resources).

***Шаблоны данных* (*data templates*)** – механизм для настройки отображения объектов определённого типа. В WPF шаблон данных ‑ это объект класса System.Windows.DataTemplate. Основное свойство шаблона ‑ VisualTree. Оно содержит визуальный элемент, определяющий внешний вид шаблона. Часто этим визуальным элементом является контейнер компоновки. В разметке XAML для задания VisualTree достаточно поместить в DataTemplate дочерний элемент. При формировании VisualTree обычно используется привязка данных для извлечения информации из объекта, для которого применяется шаблон. Сам шаблон данных, как правило, размещают в ресурсах окна или приложения. У класса DataTemplate имеется свойство DataType, определяющее тип данных, к которому будет применяться шаблон.

В WPF списковые элементы управления поддерживают возможность выбора для объекта одного из нескольких шаблонов данных. Выбор шаблона выполняется программно, с помощью создания подкласса для DataTemplateSelector и переопределения метода SelectTemplate().

Затем можно объявить объект TaskTemplateSelector в качестве ресурса и назначить этот ресурс свойству ListBox.ItemTemplateSelector. Объект ListBox вызывает метод SelectTemplate() для каждого элемента в базовой коллекции.

<ListBox Name="lstTasks" Width="250"

HorizontalContentAlignment="Stretch"

ItemTemplateSelector="{StaticResource selector}"/>

При работе с иерархическими элементами управления (например, TreeView) вместо шаблона данных на основе DataTemplate следует использовать HiererchicalDataTemplate. У такого шаблона имеется свойство ItemsSource, которое нужно связать с дочерней коллекцией, и свойство ItemTemplate – дочерний шаблон данных (DataTemplate или HiererchicalDataTemplate).