МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФГБОУ ВО «БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра математических методов и цифровых технологий

**Разработка приложений для мобильных устройств**

Отчёт по второй лабораторной работе. Вариант 14.

|  |  |  |
| --- | --- | --- |
| студента бакалавриата группы | БИ-17-1 | Точилова Е.А. Фамилия И.О. |
| проверил | к.т.н., доцент кафедры математических методов и цифровых технологий  ученое звание, должность | Родионов А.В.  Фамилия И.О |

Иркутск, 2020

# Задание 1. Основные элементы компоновки XAML

1. Изучить и описать назначение, свойства и события следующих XAML элементов и классов (на примере UWP):

− Паттерн MVVM

− Binding

− Bind

− ObservableCollection

− ListBox

− ComboBox и AutoSuggestBox

− ListView

− GridView

− FlipView

1. Провести параллели с элементами XAML из главного подпроекта, исследовать их сходство и различия.
2. Изучите создание элементов как с использованием XAML разметки, так и генерации их из кода приложения.
3. Результат выполнения задания оформить в виде отчета в Word.

# Задание 2. Индивидуальный проект

1. Напишите приложение (возможно, многостраничное) для работы с данными в соотв. с вариантом (по номеру в списке группы). Предусмотреть все необходимы для работы базовые функции

(добавление/удаление/редактирование записей и пр.) 2. Обязательно продемонстрируйте работу всех элементов из задания 1. При необходимости придумайте/дополните варианты для демонстрации работы всех элементов.

2. Ответьте на вопросы преподавателя, при необходимости выполните дополнительные задания, которые даст преподаватель.

# Паттерн MVVM

Паттерн MVVM (Model-View-ViewModel) позволяет отделить логику приложения от визуальной части (представления). Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения.

Данный паттерн был представлен [Джоном Госсманом](https://blogs.msdn.microsoft.com/johngossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/) (John Gossman) в 2005 году как модификация шаблона Presentation Model и был первоначально нацелен на разработку приложений в WPF. И хотя сейчас данный паттерн вышел за пределы WPF и применяется в самых различных технологиях, в том числе при разработке под Android, iOS, тем не менее WPF является довольно показательной технологией, которая раскрывает возможности данного паттерна.

MVVM состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View).

# Binding

Для определения привязки используется элемент Binding, который представляет объект класса Windows.UI.Xaml.Data.Binding. В общем случае привязка выглядит следующим образом:

|  |
| --- |
| {Binding ElementName=Имя\_объекта-источника, Path=Свойство\_объекта-источника} |

Параметры ElementName и Path представляют свойства класса Binding. Всего у данного класса можно выделить следующие свойства:

* ElementName: имя элемента, к которому создается привязка
* Path: свойство элемента, к которому создается привязка
* Mode: режим или направление привязки
* Source: объект-источник привязки, который необязательно является элементом управления UWP
* TargetNullValue: устанавливает значение по умолчанию, если привязанное свойство источника привязки имеет значение null

# Bind

Элемент Bind в некотором отношении аналогичен Binding, только позволяет выполнить привязку к определенным свойствам страницы. Bind имеет ряд свойств, которые также определены в Binding: Path, Mode, TargetNullValue. Но в отличие от Binding x:Bind по умолчанию работает по режиму привязки OneTime, то есть привязка осуществляется только один раз. Хотя через свойство Mode можно изменить режим привязки.

Но главное отличие между Bind и Binding состоит в том, что выражение x:Bind выполняет код, который генерируется во время компиляции, а Binding обрабатывает привязки во время выполнения. В результате привязки Bind имеет большую производительность.

В качестве источника привязки элемент Bind использует страницу или элемент управления и устанавливает привязку к их свойствам и методам. Например, пусть в коде страницы MainPage имеется свойство Phone, которое ссылается на объект Phone

# ObservableCollection

Как правило, привязка в элементах управления данными осуществляется не к стандартным спискам, а к объектам класса ObservableCollection. Почему используется именно этот класс, а не список типа List? Потому что ObservableCollection уже реализует интерфейс INotifyPropertyChanged и поэтому может уведомлять элементы, которые применяют привязку, в результате чего обновляется не только сам объект ObservableCollection, но и привязанные к нему элементы интерфейса.

# ListBox

Элемент ListBox представляет собой обычный список. Содержит коллекцию элементов ListBoxItem, которые являются элементами управления содержимым. Также ListBox может содержать любые другие элементы.

# ComboBox

ComboBox представляет раскрывающийся список. Он применяется, чтобы предоставить пользователю возможность выбрать одно из значений в списке. ComboBox больше подходит для выбора элементов, которые представляют обычные строки. Хотя более сложные данные также можно добавлять в ComboBox.

Если элементов в списке меньше пяти, то оптимальнее использовать для выбора набор элементов RadioButton или CheckBox.

# AutoSuggestBox

AutoSuggestBox представляет сложный элемент, который содержит текстовое поле и подключаемый к нему список. При вводе в текстовое поле могут автоматически подставляться похожие результаты из некоторого предопределенного списка.

Чтобы задействовать функциональные возможности данного элемента управления, надо обработать его событие TextChanged. В обработчике события необходимо лимитировать список предопределенных вариантов таким образом, чтобы они соответствовали введенному в поле тексту.

# ListView

ListView отображает информацию на множестве строк и столбцов. Он унаследован от класса ListBox, поэтому может вести себя простой список.

# GridView

GridView также, как и ListView, создает отображение данных в виде строк и столбцов. Единственное отличие GridView от ListView состоит в том, что GridView использует в качестве панели для расположения элементов класс ItemsWrapGrid и поэтому по умолчанию может создавать дополнительные колонки для отображения данных.

# FlipView

FlipView представляет элемент, который содержит несколько объектов (например, изображений), которые мы можем пролистывать. На сенсорных устройствах пролистывание осуществляется с помощью касания пальца. На устройствах, которые не поддерживют сенсорных ввод, для перелистывания используются кнопки навигации, либо клавиши со стрелками. FlipView лучше всего подходит для просмотра небольшого количества объектов (примерно до 25 объектов), например, фотографий в фотоальбоме или карточек с информацией о товаре.

FlipView наследуется от класса ItemsControl и поэтому может содержать коллекцию элементов любого типа. Поэтому мы можем либо определить все ее элементы через свойство Items, либо привязать список объектов к свойству ItemsSource элемента FlipView.

**Задание лабораторной работы**

Вариант: 14

Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит: Ф.И.О. абитуриента, оценки. Определить средний балл по университету и вывести список абитуриентов, средний балл которых выше среднего балла по университету. Первыми в списке должны идти студенты, сдавшие все экзамены на 5.

**Скриншоты лабораторной работы**









