МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 2 по курсу «Программирование мобильных систем»**

«Использование элементов пользовательского интерфейса. Использование ListView»

ВЫПОЛНИЛ студент группы 16-ИТ-3

Яблонский А.С.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Ознакомиться с различными типами элементов интерфейса. Научиться использовать элементы пользовательского интерфейса в разрабатываемых приложениях. Ознакомиться с использованием адаптеров в Android. Научиться использовать элемент ListView в разрабатываемых приложениях.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. Дайте определение понятиям «Представление» и «Группа Представлений».

**Представления** (View) являются базовым классом для всех визуальных элементов UI (элементов управления (Control) и виджетов (Widget)). Все эти элементы, в том числе и разметка (Layout), являются расширениями класса View.

**Группы Представлений** (ViewGroup) – потомки класса View; могут содержать в себе несколько дочерних Представлений. Расширение класса ViewGroup используется для создания сложных Представлений, состоящих из взаимосвязанных компонентов. Класс ViewGroup также является базовым для различных Разметок (Layout).

1. Перечислите и охарактеризуйте основные Представления в ОС Android.

•**TextView**. Стандартный элемент, предназначенный для вывода текста. Поддерживает многострочное отображение, форматирование и автоматический перенос.

•**EditText**. Редактируемое поле для ввода текста. Поддерживает многострочный ввод, перенос слов на новую строку и текст подсказки.

•**ListView**. Группа представлений (ViewGroup), которая формирует вертикальный список элементов, отображая их в виде строк внутри списка. Простейший объект ListView использует TextView для вывода на экран значений toString(), принадлежащих элементом массива.

•**Spinner**. Составной элемент, отображающий TextView в сочетании с соответствующим ListView, которое позволяет выбрать элемент списка для отображения в текстовой строке. Сама строка состоит из объекта TextView и кнопки, при нажатии на которую всплывает диалог выбора. Внешне этот элемент напоминает тэг &lt;SELECT&gt; в HTML.

•**Button**. Стандартная кнопка, которую можно нажимать.

•**CheckBox**. Кнопка, имеющая два состояния. Представлена в виде отмеченного или неотмеченного флажка («галки»).

•**RadioButton**. «Радиокнопка», позволяет выбрать только один из нескольких вариантов.

•**ViewFlipper**. Группа представлений (ViewGroup), позволяющая определить набор элементов и горизонтальную строку, в которой может выводиться только одно Представление (View). При этом переходы между отображающимися элементами осуществляются с помощью анимации.

1. Перечислите и охарактеризуйте основные виды Разметки в ОС Android.

•**FrameLayout**. Самая простая разметка, прикрепляет каждое новое дочернее Представление к левому верхнему углу экрана, накладывая новый элемент на предыдущий, заслоняя его.

•**LinearLayout**. Помещает дочерние Представления в горизонтальный или вертикальный ряд. Вертикальная разметка представляет собой колонку, а горизонтальная – строку с элементами. Данная разметка позволяет задавать не только размеры, но и «относительный вес» дочерних элементов, благодаря чему можно гибко контролировать их размещение на экране.

•**RelativeLayout**. Наиболее гибкий среди стандартных видов разметки. Позволяет указывать позиции дочерних Представлений относительно границ свободного пространства и других Представлений.

•**TableLayout**. Позволяет размещать дочерние Представления внутри ячеек «сетки», состоящей из строк и столбцов. Размеры ячеек могут оставаться постоянными или автоматически растягиваться при необходимости.

•**Gallery**. Представляет элементы в виде прокручиваемого горизонтального списка (обычно графические элементы).

1. В каком виде реализуется разметка экрана Android приложения?

Наиболее предпочтительный способ реализации разметки экрана – использование внешних ресурсов: XML-файлов, описывающих размещение элементов на экране и их параметры.

5. Возможно ли использование нескольких видов разметки на одной Активности?

6. Дайте определение понятию «Адаптер» в Android приложении.

Адаптеры в Android являются связующими классами между данными приложения и Представлениями. Адаптер отвечает за создание дочерних Представлений, отображающими каждый элемент внутри родительского виджета, а также обеспечивает доступ к исходным данным, используемым приложением. Представления, использующие привязку к Адаптеру, должны быть потомками абстрактного класса AdapterView.

7. Перечислите и охарактеризуйте основные виды Адаптеров.

**ArrayAdapter** использует механизм обобщенных типов (generics) языка Java для привязки родительского класса AdapterView к массиву объектов указанного типа. По умолчанию ArrayAdapter использует метод toString() для каждого элемента в массиве, чтобы создать и заполнить текстовыми данными виджеты TextView.

**SimpleCursorAdapter** привязывает указанное в разметке Представление к столбцам Курсора, ставшего результатом запроса к СУБД или Контент-Провайдеру. Для его использования требуется описать разметку в формате XML, а затем привязать каждый столбец к Представлениям из этой разметки. Адаптер создаст Представления для каждой записи из Курсора и наполнит их данными из соответствующих столбцов.

**SimpleAdapter** позволяет привязать ListView к списку ArrayList, содержащему объекты типа Map (ассоциативные массивы, содержащие пары «ключ-значение»). Для каждого такого объекта при отображении используется один элемент из ListView. Как и для SimpleCursorAdapter, для отображения применяется XML-разметка, к элементам которой привязываются члены каждого объекта типа Map.

8. Какой стандартный адаптер используется для работы с результатом запроса СУБД? **SimpleCursorAdapter**

9. Охарактеризуйте Представление ListView.

Группа представлений (ViewGroup), которая формирует вертикальный список элементов, отображая их в виде строк внутри списка. Простейший объект ListView использует TextView для вывода на экран значений toString(), принадлежащих элементом массива.

10. В каком виде храниться внешний ресурс, содержащий массив данных?

В основном ресурсы хранятся в виде XML-файлов в каталоге **res**.

**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. Выполнено размещение элементов пользовательского интерфейса с помощью LinearLayout.
2. Выполнено размещение элементов пользовательского интерфейса с помощью RelativeLayout.
3. Отредактировано содержимое файла **res/values/strings.xml**.
4. Внесены изменения в файл разметки **res/layout/activity\_main.xml**.
5. Внесены изменения в файл AndroidManifest.xml.
6. Создано activity с использованием TabWidget.
7. Создано activity с использованием WebView.
8. Создан второй проект Android.
9. В каталоге **res/values** создан и отредактирован новый файл **arrays.xml.**
10. В каталоге **res/layout** создан и отредактирован новый файл **list\_item.xml.**

**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:



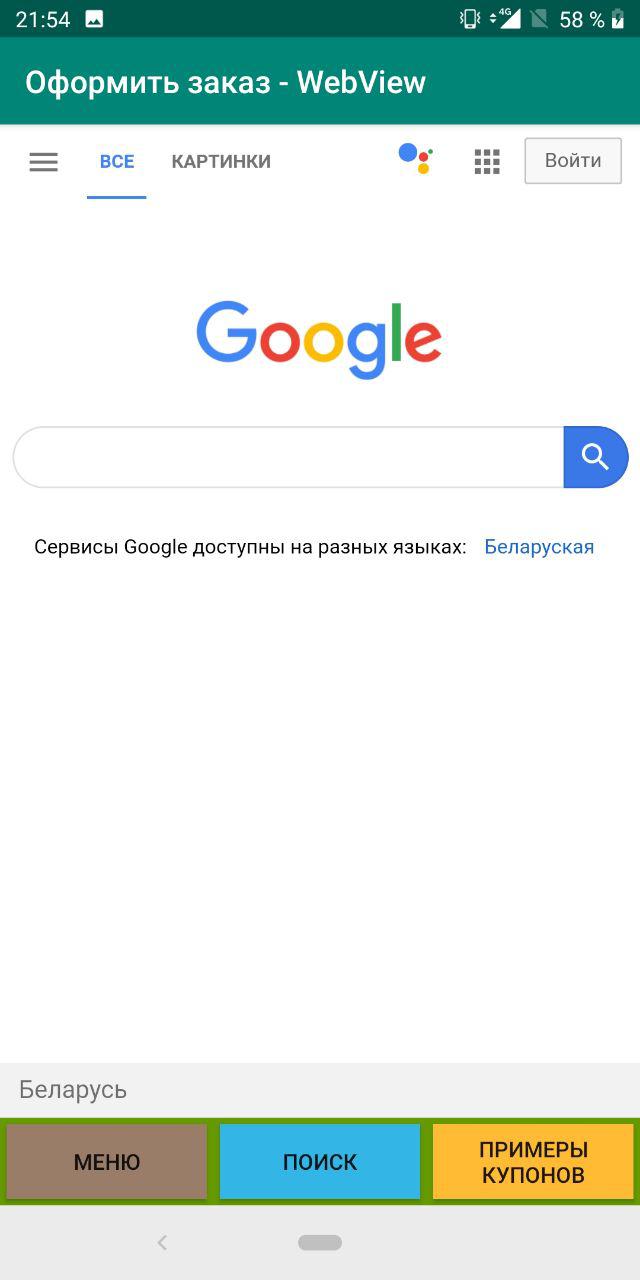


**Рисунок 1** — Вариант интерфейса с помощью LinearLayout

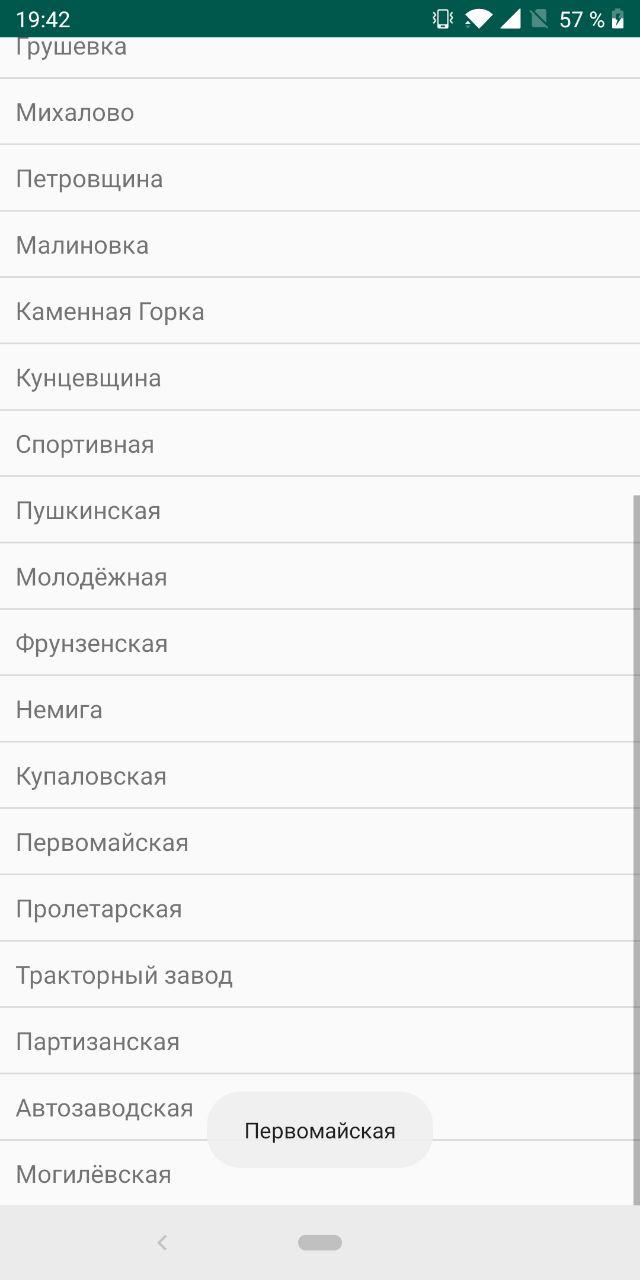
**Рисунок 2** — Вариант интерфейса



**Рисунок 3** — Вариант интерфейса с помощью TabWidget



**Рисунок 4** — Вариант интерфейса с помощью WebView



**Рисунок 5** — Создание примера активность