МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 3 по курсу «Программирование мобильных систем»**

«Использование управляющих элементов в пользовательском интерфейсе. Работа с базами данных в Android»

ВЫПОЛНИЛ студент группы 16-ИТ-3

Яблонский А.С.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Научиться использовать в интерфейсе пользователя различные управляющие элементы: кнопки с изображениями, радиокнопки, чекбоксы и другие. Научиться взаимодействовать с базой данных в Android приложении.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. Дайте определение понятию «Управляющий элемент»?

**Управляющие элементы** - это специализированные окна, которые позволяют пользователю предопределенным образом задавать или выбирать данные (кнопки с изображениями, радиокнопки, чекбоксы и другие).

1. Перечислите и охарактеризуйте основные управляющие элементы в ОС Android.

|  |  |
| --- | --- |
| **Управляющий элемент** | **Характеристика** |
| Button | Кнопка, которую пользователь может нажать для выполнения действия. |
| CheckBox | Переключатель, которым можно воспользоваться для включения или отключения функции, или компонента. Флажки следует использовать при отображении группы доступных для выбора параметров, которые не являются взаимоисключающими. |
| ToggleButton | Данный виджет хорошо подходит в качестве альтернативы радиокнопкам и чекбоксам,  когда требуется переключаться между двумя взаимоисключающими состояниями, например,  Включено/Выключено. |
| RadioButton | Радиокнопки используются для выбора между различными взаимоисключающими  вариантами. Для создания группы радиокнопок используется элемент RadioGroup, внутри  которого располагаются элементы RadioButton. |
| EditText | Виджет EditText используется для ввода текста пользователем. Установленный для этого  виджета обработчик нажатий на кнопки будет показывать введенный текст с помощью Toast. |
| Spinner | Раскрывающийся список параметров, в котором пользователь может выбрать только одно значение. |
| DatePicker, TimePicker | Элементы диалогового окна для выбора одного значения из набора с помощью кнопок со стрелками вверх и вниз или с помощью жеста пролистывания. Для ввода значений даты (дня, месяца, года) используйте виджет DatePicker, а для ввода значений времени (часов, минут и указания времени до полудня или после [AM/PM]) — виджет TimePicker. |

1. В чём заключается отличие управляющих элементов CheckBox и RadioButton?

**CheckBox** всегда можно выбрать несколько, даже если они относятся к одному вопросу. Т.е. выбрать один, несколько, все или даже не выбрать ни одного. Семантически **CheckBox** спрашивает согласны ли вы конкретно с тем что в нём написано или нет. А **RadioButton**, если он относится к одному вопросу/полю, всегда можно выбрать только один. (Их может быть несколько групп, но выбрать можно только один в каждой группе.) Семантически **RadioButton** спрашивает с каким из варианта в данной группе вы согласны больше всего.

1. Какими способами можно создавать управляющие элементы и задавать их свойства?

* **Объявление элементов пользовательского интерфейса в XML**. В Android имеется удобный справочник XML-элементов для классов View и их подклассов, например, таких, которые используются для виджетов и макетов.
* **Создание экземпляров элементов во время выполнения**. Ваше приложение может программным образом создавать объекты View и ViewGroup (а также управлять их свойствами).

1. Как происходит обработка взаимодействия с Управляющими элементами?

**Первый способ - атрибут onClick**

Относительно новый способ, специально разработанный для Android - использовать атрибут **onClick** (на панели свойств отображается как **On Click**):

*android:onClick="onMyButtonClick"*

### Второй способ - метод setOnClickListener()

Более традиционный способ в Java - через метод **setOnClickListener()**, который прослушивает нажатия на кнопку.

*button.setOnClickListener(new View.OnClickListener() {*

*@Override*

*public void onClick(View v) {*

*}});*

### Третий способ - интерфейс OnClickListener

Третий способ является родственным второму способу и также является традиционным для Java. Кнопка присваивает себе обработчика с помощью метода **setOnClickListener (View.OnClickListener l)**, т.е. подойдет любой объект с интерфейсом **View.OnClickListener**. Мы можем указать, что наш класс **Activity** будет использовать интерфейс **View.OnClickListener**.

1. Дайте определение понятию «Курсор».

**Курсор** - это набор строк в табличном виде. Для доступа курсора вы должны использовать метод **moveToFirst()**, так как курсор размещается перед первой строкой.

1. Перечислите и охарактеризуйте основные методы объекта Cursor.

Класс **Cursor** содержит немало возможностей для навигации (но не ограничивается только ими):

* moveToFirst() — перемещает курсор на первую строку в результате запроса;
* moveToNext() — перемещает курсор на следующую строку;
* moveToLast() - перемещает курсор на последнюю строку;
* moveToPrevious() — перемещает курсор на предыдущую строку;
* getCount() — возвращает количество строк в результирующем наборе данных;
* getColumnIndexOrThrow() — возвращает индекс для столбца с указанным именем (выбрасывает исключение, если столбец с таким именем не существует);
* getColumnName() — возвращает имя столбца с указанным индексом;
* getColumnNames() — возвращает массив строк, содержащий имена всех столбцов в объекте Cursor;
* moveToPosition() — перемещает курсор на указанную строку;
* getPosition() — возвращает текущую позицию курсора

1. Опишите механизм взаимодействия приложения с СУБД с использованием адаптера.

При создании приложений, использующих СУБД, во многих случаях удобно применять инструменты, называющиеся ORM (Object-Relationship Mapping), отображающие данные из одной или нескольких таблиц на объекты используемого языка программирования. Кроме того, ORM позволяют абстрагироваться от конкретной реализации и структуры таблиц и берут на себя обязанности по взаимодействию с СУБД.

Хорошей практикой является создание вспомогательного класса, берущего на себя работу с СУБД. Данный класс обычно инкапсулирует взаимодействия с базой данных, предоставляя интуитивно понятный строго типизированный способ удаления, добавления и изменения объектов. Такой Адаптер базы данных также должен обрабатывать запросы к БД и переопределять методы для открытия, закрытия и создания базы данных. Его также обычно используют как удобное место для хранения статических констант, относящихся к базе данных, таких, например, как имена таблиц и полей.

1. Какой метод используется для выполнения запроса к БД?

*SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db", MODE\_PRIVATE, null);*

*db.execSQL("CREATE TABLE IF NOT EXISTS users (name TEXT, age INTEGER)");*

1. Перечислите и охарактеризуйте основные операции с БД?

|  |  |
| --- | --- |
| **Основные операции** | **Характеристика** |
| Вставка строк | Метод insert хочет получать (кроме других параметров) объект ContentValues, содержащий значения полей вставляемой строки и возвращает значение индекса:  *ContentValues newRow = new ContentValues();*  *newRow.put(COLUMN\_NAME, columnValue);*  *db.insert(DATABASE\_TABLE, null, newRow);* |
| Обновление строк | Также используется ContentValues, содержащий подлежащие изменению поля, а для указания, какие именно строки нужно изменить, используется параметр where, имеющий стандартный для SQL вид:  *ContentValues updatedValues = new ContentValues();*  *// Повторяем для всех нужных полей*  *updatedValues.put(COLUMN\_NAME, newValue);*  *// Указываем условие*  *String where = COLUMN\_NAME + &quot;=&quot; + &quot;&#39;Бармаглот&#39;&quot;;*  *// Обновляем*  *db.update(DATABASE\_TABLE, updatedValues, where, null);*  Метод update возвращает количество измененных строк. Если в качестве параметра *where* передать null, будут изменены все строки таблицы. |
| Удаление строк | Выполняет похожим на update образом:  *db.delete(DATABASE\_TABLE, KEY\_ID + &quot;=&quot; + rowId, null);* |
| Использование | SimpleCursorAdapter позволяет привязать курсор к ListView. используя описание разметки  для отображения строк и полей. Для однозначного определения того, в каких элементах  разметки какие поля, получаемые через курсор, следует отображать, используются два массива:  строковый с именами полей строк, и целочисленный с идентификаторами элементов разметки. |

**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. В приложение добавлена **графическая кнопка**.
2. В приложение добавлена **чекбокс**.
3. В приложение добавлена **ToggleButton**.
4. В приложение добавлена **EditText**.
5. В приложение добавлена **RadioButton**.
6. В приложение добавленакнопка «Записать в журнал».
7. Создан новый проект **PMS\_laba3\_bd**, главная Активность которого будет расширять класс **ListActivity**.
8. В методе onCreate главной Активности сделано создание БД, использующая в качестве основы информацию из пункта «Работа с СУБД без адаптера» и метода onUpgrade вспомогательного класса из предшествующего ему пункта. После создания БД создана таблица с любым полем (плюс индекс), в которую добавлено 5 уникальных записей со строковым полем. Индекс должен инкрементироваться автоматически.
9. С помощью дополнительной разметки и адаптера ArrayAdapter отображены полученные из СУБД данные на экране Активности.
10. Добавлен обработчик клика на элемент списка, чтобы при клике запрашивался из БД индекс элемента с этой строкой и отображался с помощью Toast.
11. Добавлено контекстное меню к элементам списка, содержащее пункт «удалить» и «редактировать».
12. Реализовано редактирование записей по нажатию на кнопку, а удаление через контекстное меню. Работа с СУБД осуществляется с использованием адаптера.

**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:



**Видео 1** — Результат выполнения первой части задания



**Видео 2** — Результат выполнения второй части задания (бд)

**Вывод:** в результате выполнения лабораторной работы я научился использовать в интерфейсе пользователя различные управляющие элементы: кнопки, радиокнопки, чекбоксы и другие, а также научился взаимодействовать с базой данных в Android приложении.