МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий Кафедра технологий программирования

# Лабораторная работа №3

по дисциплине: «Программируемые мобильные системы»

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

Вариант №4

Выполнил Студент группы 21-ИТ-2

Зинькевич И.Ю.

Проверил Ассистент

Васильева Д.М.

Полоцк 2023

**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.

Ход работы:

1. Добавить в приложение графические кнопки и проследить, как изменяется изображение кнопки в разных состояниях, а также как функционирует обработчик нажатия на кнопку.

2. Добавить в приложение чекбокс и проследить за поведением чекбокса в разных ситуациях.

3. Добавить в приложение виджет ToggleButton.

4. Добавить в приложение виджет RadioButton:

5. Добавить в приложение виджет EditText и кнопку «Очистить» в разметку и написать обработчик, очищающий текстовое поле (используя метод setText виджета EditText).

6. Создать новый проект SQLTest, главная Активность которого будет расширять класс ListActivity.

7. В методе onCreate главной Активности сделать открытие или создание БД, используя в качестве основы информацию из пункта «Работа с СУБД без адаптера» и метода onUpgrade вспомогательного класса из предшествующего ему пункта. После создания БД создать таблицу с любым полем (плюс индекс), в которую добавьте 3..5 уникальных записей со строковым полем. Индекс должен инкрементироваться автоматически.

8. В этом же методе сделать запрос к таблице, получающий все строки и с помощью имеющегося курсора запишите данные в массив строк.

9. С помощью дополнительной разметки и адаптера ArrayAdapter отобразить полученные из СУБД данные на экране Активности.

10. Добавить обработчик клика на элемент списка, чтобы при клике запрашивался из БД индекс элемента с этой строкой и отображался с помощью Toast.

11. Добавить контекстное меню к элементам списка, содержащее пункт «удалить» и

реализовать удаление. После удаления должны производится действия из п.п. 3 и 4 для изменения состава отображаемых данных.

12. Реализовать редактирование записей по нажатию на кнопку, а удаление через контекстное меню. Работа с СУБД должна осуществляться с использованием адаптера. Пример обработчика выбора пункта контекстного меню приведен в соответствующем разделе данного методического руководства.

# Листинг программы:

# TabWidjet.java:

# public class TabWidjet extends AppCompatActivity {

# private static final String DATABASE\_NAME = "myDatabase.db";

# private static final String DATABASE\_TABLE = "mainTable";

# private static final String DATABASE\_CREATE = "create table "

# + DATABASE\_TABLE + " ( \_id integer primary key autoincrement,"

# + "username text not null);";

# SQLiteDatabase myDatabase;

# @Override

# protected void onCreate(Bundle savedInstanceState) {

# super.onCreate(savedInstanceState);

# setContentView(R.layout.activity\_tab\_widjet);

# setTitle("top 1 tabhost");

# TabHost tabHost = (TabHost) findViewById(R.id.tabHost);

# tabHost.setup();

# TabHost.TabSpec tabSpec = tabHost.newTabSpec("tag1");

# // ----------ВКЛАДКИ------------------------------------------------------------------------------------------------------------------

# tabSpec = tabHost.newTabSpec("tag1");

# tabSpec.setContent(R.id.tab1);

# tabSpec.setIndicator("кнопачьки))))))");

# tabHost.addTab(tabSpec);

# tabSpec = tabHost.newTabSpec("tag2");

# tabSpec.setContent(R.id.tab2);

# tabSpec.setIndicator("база даних))))");

# tabHost.addTab(tabSpec);

# tabHost.setCurrentTab(0);

# // ----------ЧАСТЬ ЗАДАНИЯ 1------------------------------------------------------------------------------------------------------------------

# // Button button = (Button)findViewById(R.id.button);

# // button.setFocusable(true);

# // button.setFocusableInTouchMode(true);

# // ----------ЧАСТЬ ЗАДАНИЯ 5------------------------------------------------------------------------------------------------------------------

# final EditText userName = (EditText) findViewById(R.id.user\_name);

# userName.setOnKeyListener(new View.OnKeyListener() {

# @Override

# public boolean onKey(View v, int keyCode, KeyEvent event) {

# if ((event.getAction() == KeyEvent.ACTION\_DOWN)

# && (keyCode == KeyEvent.KEYCODE\_ENTER)) {

# Toast.makeText(getApplicationContext(),

# userName.getText(), Toast.LENGTH\_SHORT).show();

# return true;

# }

# return false;

# }

# });

# // ----------БАЗЫ-----------------------------------------------------------------------------------------------------------------------------

# // ----------ДАНИХ----------------------------------------------------------------------------------------------------------------------------

# dropDatabase();

# createDatabase();

# insertData();

# ListView listView = findViewById(R.id.listView);

# ArrayList<String> newData = receiveData();

# ArrayAdapter<String> newAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, newData);

# listView.setAdapter(newAdapter);

# listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

# @Override

# public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

# String selectedItem = (String) parent.getItemAtPosition(position);

# int dataIndex = getIndexFromDatabase(selectedItem);

# Toast.makeText(getApplicationContext(), "Record index: " + dataIndex, Toast.LENGTH\_SHORT).show();

# }

# });

# registerForContextMenu(listView);

# }

# private void createDatabase() {

# myDatabase = openOrCreateDatabase(DATABASE\_NAME, MODE\_PRIVATE, null);

# myDatabase.execSQL(DATABASE\_CREATE);

# }

# private void dropDatabase() {

# getApplicationContext().deleteDatabase(DATABASE\_NAME);

# }

# private void insertData() {

# ContentValues values = new ContentValues();

# values.put("username", "Ihar");

# myDatabase.insert(DATABASE\_TABLE, null, values);

# values.clear();

# values.put("username", "Kirill");

# myDatabase.insert(DATABASE\_TABLE, null, values);

# values.clear();

# values.put("username", "Roma");

# myDatabase.insert(DATABASE\_TABLE, null, values);

# }

# private ArrayList<String> receiveData() {

# ArrayList<String> data = new ArrayList<>();

# Cursor cursor = myDatabase.query(DATABASE\_TABLE, null, null, null, null, null, null);

# if (cursor.moveToFirst()) {

# do {

# String columnOneValue = cursor.getString(cursor.getColumnIndex("username")); // какова хуя тут ошибка если все работает =аааахахахахъа android разрабы хуесосы)))))))

# data.add(columnOneValue);

# } while (cursor.moveToNext());

# }

# cursor.close();

# return data;

# }

# private int getIndexFromDatabase(String selectedItem) {

# int index = -1;

# Cursor cursor = myDatabase.query(DATABASE\_TABLE, new String[]{"\_id"}, "username=?", new String[]{selectedItem}, null, null, null);

# if (cursor.moveToFirst()) {

# index = cursor.getInt(cursor.getColumnIndex("\_id"));

# }

# cursor.close();

# return index;

# }

# @Override

# public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {

# super.onCreateContextMenu(menu, v, menuInfo);

# MenuInflater inflater = getMenuInflater();

# inflater.inflate(R.menu.context\_menu, menu);

# }

# @Override

# public boolean onContextItemSelected(@NonNull MenuItem item) {

# try {

# AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();

# ListView listView = findViewById(R.id.listView);

# String selectedItem = (String) listView.getItemAtPosition(info.position);

# int itemId = item.getItemId();

# if (itemId == R.id.menu\_delete) {

# deleteItemFromDatabase(selectedItem);

# refreshData();

# return true;

# } else if(itemId == R.id.menu\_edit){

# EditText newUsernameEditText = findViewById(R.id.editTextNewUsername);

# String newUsername = newUsernameEditText.getText().toString();

# Toast.makeText(this, newUsername, Toast.LENGTH\_SHORT).show();

# editItemFromDatabase(selectedItem, newUsername);

# refreshData();

# return true;

# }

# else {

# return super.onContextItemSelected(item);

# }

# } catch (Exception e) {

# Log.e("YourActivity", "Error in onContextItemSelected", e);

# return false;

# }

# }

# private void deleteItemFromDatabase(String selectedItem) {

# myDatabase.delete(DATABASE\_TABLE, "username=?", new String[]{selectedItem});

# }

# private void editItemFromDatabase(String selectedItem, String newUsername) {

# ContentValues values = new ContentValues();

# values.put("username", newUsername);

# int rowsAffected = myDatabase.update(DATABASE\_TABLE, values, "username=?", new String[]{selectedItem});

# if (rowsAffected > 0) {

# Toast.makeText(getApplicationContext(), "cool", Toast.LENGTH\_SHORT).show();

# } else {

# Toast.makeText(getApplicationContext(), "vse ploho =(", Toast.LENGTH\_SHORT).show();

# }

# }

# private void refreshData() {

# ArrayList<String> newData = receiveData();

# ArrayAdapter<String> newAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, newData);

# ListView listView = findViewById(R.id.listView);

# listView.setAdapter(newAdapter);

# }

# //----------ЗАДАНИЕ 1------------------------------------------------------------------------------------------------------------------

# public void onButtonClicked(View view) {

# Toast.makeText(this, "Кнопка нажата", Toast.LENGTH\_SHORT).show();

# }

# //----------ЗАДАНИЕ 2------------------------------------------------------------------------------------------------------------------

# public void onCheckboxClicked(View v) {

# if (((CheckBox) v).isChecked()) {

# Toast.makeText(this, "Отмечено", Toast.LENGTH\_SHORT).show();

# } else {

# Toast.makeText(this, "Не отмечено", Toast.LENGTH\_SHORT).show();

# }

# }

# //----------ЗАДАНИЕ 3------------------------------------------------------------------------------------------------------------------

# public void onToggleClicked(View v) {

# if (((ToggleButton) v).isChecked()) {

# Toast.makeText(this, "Включено", Toast.LENGTH\_SHORT).show();

# } else {

# Toast.makeText(this, "Выключено", Toast.LENGTH\_SHORT).show();

# }

# }

# //----------ЗАДАНИЕ 4------------------------------------------------------------------------------------------------------------------

# public void onRadioButtonClicked(View v) {

# RadioButton rb = (RadioButton) v;

# Toast.makeText(this, "Выбрано животное: " + rb.getText(),

# Toast.LENGTH\_SHORT).show();

# }

# }

# Activity\_tab\_widjet.layout:

# <?xml version="1.0" encoding="utf-8"?>

# <TabHost

# android:id="@+id/tabHost"

# xmlns:android="http://schemas.android.com/apk/res/android"

# android:layout\_width="match\_parent"

# android:layout\_height="wrap\_content"

# android:layout\_gravity="center\_horizontal">

# <LinearLayout

# android:layout\_width="match\_parent"

# android:layout\_height="match\_parent"

# android:orientation="vertical">

# <TabWidget

# android:id="@android:id/tabs"

# android:layout\_width="match\_parent"

# android:layout\_height="wrap\_content">

# </TabWidget>

# <FrameLayout

# android:id="@android:id/tabcontent"

# android:layout\_width="match\_parent"

# android:layout\_height="match\_parent">

# <LinearLayout

# android:id="@+id/tab1"

# android:layout\_width="fill\_parent"

# android:layout\_height="match\_parent"

# android:layout\_weight="1"

# android:orientation="vertical">

# <Button

# android:id="@+id/button"

# android:layout\_width="wrap\_content"

# android:layout\_height="76dp"

# android:background="@drawable/smile\_button"

# android:onClick="onButtonClicked"

# android:padding="10dp"

# android:text="@android:string/ok" />

# <CheckBox

# android:id="@+id/checkbox"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:onClick="onCheckboxClicked"

# android:text="Выбери меня" />

# <ToggleButton android:id="@+id/togglebutton"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:textOn="Звонок включен"

# android:textOff="Звонок выключен"

# android:onClick="onToggleClicked"/>

# <RadioGroup

# android:layout\_width="fill\_parent"

# android:layout\_height="wrap\_content"

# android:orientation="vertical">

# <RadioButton

# android:id="@+id/radio\_dog"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:onClick="onRadioButtonClicked"

# android:text="Собака" />

# <RadioButton

# android:id="@+id/radio\_cat"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:onClick="onRadioButtonClicked"

# android:text="Кошка" />

# <RadioButton

# android:id="@+id/radio\_rabbit"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:onClick="onRadioButtonClicked"

# android:text="Рома" />

# <EditText

# android:id="@+id/user\_name"

# android:layout\_width="fill\_parent"

# android:layout\_height="wrap\_content"

# android:hint="Введите имя"/>

# </RadioGroup>

# </LinearLayout>

# <LinearLayout

# android:id="@+id/tab2"

# android:layout\_width="fill\_parent"

# android:layout\_height="match\_parent"

# android:layout\_weight="1"

# android:orientation="vertical">

# <ListView

# android:id="@+id/listView"

# android:layout\_width="match\_parent"

# android:layout\_height="wrap\_content" >

# </ListView>

# <EditText

# android:id="@+id/editTextNewUsername"

# android:layout\_width="wrap\_content"

# android:layout\_height="wrap\_content"

# android:hint="чтоб менять пиши сюда"/>

# </LinearLayout>

# </FrameLayout>

# </LinearLayout>

# </TabHost>

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