Министерство образования Республики Беларусь

Учреждение образования «Полоцкий государственный университет имени Евфросинии Полоцкой»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 4 по курсу «Программирование мобильных систем»**

«Вызов Активности с помощью явного намерения и получение результатов работы.

Использование неявных Намерений. Получение данных из Намерения»

ВЫПОЛНИЛ студент группы 21-ИТ-1

Чиникайло А.П.

ПРОВЕРИЛ преподаватель

Васильева Д.М.

Полоцк 2024 г.

Цель работы: Научиться вызывать Активность с использованием явного намерения и получать результаты её работы. Научиться использовать неявные Намерения и получать данные из Намерения.

**ХОД РАБОТЫ**

Задание:

1. Продолжить разработку проекта из Л.р.2 и Л.р.3.

2. Добавить в приложение вызов явного и неявного намерения.

3. Модифицировать методы onCreate Активностей так, чтобы с помощью Toast

они показывали действие вызвавшего их Намерения.

Листинг 1 – MainActivity.kt

package com.example.lab\_3  
  
import android.annotation.SuppressLint  
import android.os.Bundle  
import android.view.KeyEvent  
import android.view.View  
import android.widget.Button  
import android.widget.CheckBox  
import android.widget.EditText  
import android.widget.RadioButton  
import android.widget.Toast  
import android.widget.ToggleButton  
import androidx.activity.ComponentActivity  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.tooling.preview.Preview  
import com.example.lab\_3.ui.theme.Lab\_3Theme  
  
class MainActivity : ComponentActivity() {  
 @SuppressLint("MissingInflatedId")  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.main);  
 val userName = findViewById<EditText>(R.id.user\_name)  
 userName.setOnKeyListener { v, keyCode, event ->  
 if (event.action == KeyEvent.ACTION\_DOWN && keyCode == KeyEvent.KEYCODE\_ENTER) {  
 Toast.makeText(applicationContext, userName.text, Toast.LENGTH\_SHORT).show()  
 true  
 } else {  
 false  
 }  
 }  
 }  
 fun onButtonClicked(v: View) {  
 Toast.makeText(this, "Кнопка нажата", Toast.LENGTH\_SHORT).show()  
 }  
 fun onCheckboxClicked(v: View) {  
 if ((v as CheckBox).isChecked) {  
 Toast.makeText(this, "Отмечено", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(this, "Не отмечено", Toast.LENGTH\_SHORT).show()  
 }  
 }  
 fun onToggleClicked(v: View) {  
 if ((v as ToggleButton).isChecked) {  
 Toast.makeText(this, "Включено", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(this, "Выключено", Toast.LENGTH\_SHORT).show()  
 }  
 }  
 fun onRadioButtonClicked(v: View) {  
 val rb = v as RadioButton  
 Toast.makeText(this, "Выбрано животное: ${rb.text}", Toast.LENGTH\_SHORT).show()  
 }  
  
}

Листинг 2 – main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/smile\_button"  
 android:onClick="onButtonClicked"  
 android:padding="10dp"  
 android:maxWidth="10dp"  
 android:maxHeight="10dp"  
 android:text="Ok"/>  
 <CheckBox  
 android:id="@+id/checkbox1"  
 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="Кролик" />  
 </RadioGroup>  
  
 <EditText  
 android:id="@+id/user\_name"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Введите имя"/>  
 <ListView  
 android:id="@android:id/list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:longClickable="true"/>  
</LinearLayout>

Листинг 3 – DBActivity.kt

package com.example.lab\_3  
  
import android.content.Context  
import android.content.Intent  
import android.graphics.Rect  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.TypedValue  
import android.view.ContextMenu  
import android.view.MenuItem  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import android.view.inputmethod.InputMethodManager  
  
class DBActivity : AppCompatActivity() {  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var userAdapter: UserAdapter  
 private lateinit var dbHandler: DBHandler  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_dbactivity)  
  
 var btnInsert = findViewById<Button>(R.id.btnInsert)  
 var etvName = findViewById<EditText>(R.id.etvName)  
 var etvAge = findViewById<EditText>(R.id.etvAge)  
  
 btnInsert.setOnClickListener {  
 if (etvName.text.toString().isNotEmpty() &&  
 etvAge.text.toString().isNotEmpty()  
 ) {  
 var user = User(etvName.text.toString(), etvAge.text.toString().toInt())  
 etvName.text.clear()  
 etvAge.text.clear()  
  
 dbHandler.insertData(user)  
 updateUsers()  
  
 val imm = getSystemService(Context.INPUT\_METHOD\_SERVICE) as InputMethodManager  
 imm.hideSoftInputFromWindow(it.windowToken, 0)  
 } else {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 recyclerView = findViewById(R.id.recycler\_view)  
 recyclerView.setHasFixedSize(true)  
 recyclerView.layoutManager = LinearLayoutManager(this)  
  
 dbHandler = DBHandler(this)  
 var users = dbHandler.readData()  
  
 userAdapter = UserAdapter(users)  
 recyclerView.adapter = userAdapter  
  
 userAdapter.onItemClick = {  
 Toast.makeText(this, "User ID: ${it.id}", Toast.LENGTH\_LONG).show()  
  
 val intent = Intent(this, EditUserActivity::class.java)  
 intent.putExtra("USER\_ID", it.id)  
 startActivity(intent)  
 }  
  
  
 recyclerView.addItemDecoration(object : RecyclerView.ItemDecoration() {  
 override fun getItemOffsets(outRect: Rect, view: View, parent: RecyclerView, state: RecyclerView.State) {  
 super.getItemOffsets(outRect, view, parent, state)  
  
 val position = parent.getChildAdapterPosition(view)  
 val isLastItem = position == state.itemCount - 1  
  
 if (!isLastItem) {  
 outRect.bottom = 10.dpToPx(parent.context)  
 }  
 }  
 })  
  
 registerForContextMenu(recyclerView)  
 }  
  
 override fun onResume() {  
 super.onResume()  
 updateUsers()  
 }  
  
 private fun updateUsers() {  
 val users = dbHandler.readData()  
 userAdapter.updateUsers(users)  
 }  
  
 override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo: ContextMenu.ContextMenuInfo?) {  
 super.onCreateContextMenu(menu, v, menuInfo)  
 menu.add(0, v.id, 0, "Удалить")  
 menu.add(0, v.id, 0, "Изменить")  
 }  
  
 override fun onContextItemSelected(item: MenuItem): Boolean {  
 if (item.title == "Удалить") {  
 val user = userAdapter.getSelectedUser()  
 dbHandler.deleteById(user.id)  
  
 val users = dbHandler.readData()  
 userAdapter.updateUsers(users)  
 }  
 if(item.title == "Изменить"){  
 val user = userAdapter.getSelectedUser()  
 Toast.makeText(this, "User ID: ${user.id}", Toast.LENGTH\_LONG).show()  
  
 val intent = Intent(this, EditUserActivity::class.java)  
 intent.putExtra("USER\_ID", user.id)  
 startActivity(intent)  
 }  
 return super.onContextItemSelected(item)  
 }  
  
 private fun Int.dpToPx(context: Context): Int {  
 val metrics = context.resources.displayMetrics  
 return TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP, this.toFloat(), metrics)  
 .toInt()  
 }  
}

Листинг 4 – activity\_dbactivity.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <EditText  
 android:id="@+id/etvName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:hint="Name" />  
 <EditText  
 android:id="@+id/etvAge"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:inputType="number"  
 android:hint="Age" />  
 <Button  
 android:id="@+id/btnInsert"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="Insert" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:weightSum="3">  
 </LinearLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycler\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</LinearLayout>

Листинг 5 – list\_item.xml

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

<androidx.constraintlayout.widget.ConstraintLayout

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

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

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

<androidx.cardview.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:padding="20dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

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

<TextView

android:id="@+id/text\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:textSize="16sp"

android:padding="10dp"

android:text="TextView"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>

Листинг 6 – EditUserActivity.kt

package com.example.lab\_3  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
  
class EditUserActivity : AppCompatActivity() {  
 private lateinit var dbHandler: DBHandler  
 private lateinit var user: User  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_edit\_user)  
  
 var nameEditText: EditText = findViewById(R.id.name\_edit\_text)  
 var ageEditText: EditText = findViewById(R.id.age\_edit\_text)  
 var saveButton: Button = findViewById(R.id.save\_button)  
  
 dbHandler = DBHandler(this)  
  
 val userId = intent.getIntExtra("USER\_ID", -1)  
 if (userId == -1) {  
 Toast.makeText(this, "Invalid user ID", Toast.LENGTH\_LONG).show()  
 finish()  
 return  
 }  
  
 user = dbHandler.getUser(userId)  
 nameEditText.setText(user.name)  
 ageEditText.setText(user.age.toString())  
  
 saveButton.setOnClickListener {  
 val name = nameEditText.text.toString()  
 val age = ageEditText.text.toString().toIntOrNull()  
 if (name.isBlank() || age == null) {  
 Toast.makeText(this, "Please enter valid data", Toast.LENGTH\_LONG).show()  
 return@setOnClickListener  
 }  
  
 user.id = userId  
 user.name = name  
 user.age = age  
 dbHandler.updateUser(user)  
  
 Toast.makeText(this, "User updated", Toast.LENGTH\_LONG).show()  
 finish()  
 }  
 }  
}

Листинг 7 – activtity\_edit\_user.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/name\_edit\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Name" />  
  
 <EditText  
 android:id="@+id/age\_edit\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Age"  
 android:inputType="number" />  
  
 <Button  
 android:id="@+id/save\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Save" />  
  
</LinearLayout>

Листинг 8 – DBHandler.kt

package com.example.lab\_3  
  
import android.content.Context  
import android.content.ContentValues  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
import android.widget.Toast  
  
const val DATABASE\_NAME = "MyDB"  
const val TABLE\_NAME = "Users"  
const val COL\_NAME = "name"  
const val COL\_AGE = "age"  
const val COL\_ID = "id"  
  
class DBHandler(var context: Context) : SQLiteOpenHelper(context, DATABASE\_NAME, null, 1) {  
 override fun onCreate(db: SQLiteDatabase?) {  
 val createTable = "CREATE TABLE " + TABLE\_NAME +" (" +  
 COL\_ID +" INTEGER PRIMARY KEY AUTOINCREMENT," +  
 COL\_NAME + " VARCHAR(256)," +  
 COL\_AGE +" INTEGER)"  
  
 db?.execSQL(createTable)  
  
 if (db != null) {  
 insertStaticData(db)  
 }  
 }  
  
 private fun insertStaticData(db: SQLiteDatabase) {  
 val users = listOf(  
 User("Misha", 24),  
 User("Dima", 22),  
 User("Lesha", 20),  
 User("Sergey", 17),  
 User("Alesya", 14)  
 )  
 for (user in users) {  
 val cv = ContentValues()  
 cv.put(COL\_NAME, user.name)  
 cv.put("age", user.age)  
 db.insert(TABLE\_NAME, null, cv)  
 }  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")  
 onCreate(db)  
 }  
  
 fun insertData(user: User) {  
 val db = this.writableDatabase  
 var cv = ContentValues()  
 cv.put(COL\_NAME, user.name)  
 cv.put(COL\_AGE, user.age)  
 var result = db.insert(TABLE\_NAME, null, cv)  
  
 if (result == (-1).toLong()) {  
 Toast.makeText(context, "Failed", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(context, "Success", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 fun readData() : MutableList<User>{  
 var list : MutableList<User> = ArrayList()  
  
 val db = this.readableDatabase  
 val query = "Select \* from $TABLE\_NAME"  
 val result = db.rawQuery(query,null)  
 if(result.moveToFirst()){  
 do {  
 var user = User()  
 user.id = result.getString(result.getColumnIndexOrThrow(COL\_ID)).toInt()  
 user.name = result.getString(result.getColumnIndexOrThrow(COL\_NAME))  
 user.age = result.getString(result.getColumnIndexOrThrow(COL\_AGE)).toInt()  
 list.add(user)  
 } while (result.moveToNext())  
 }  
  
 result.close()  
 db.close()  
 return list  
 }  
  
 fun getUser(id: Int): User {  
 val db = this.readableDatabase  
 val cursor = db.query("users", null, "id = ?",  
 arrayOf(id.toString()), null, null, null)  
  
 if (cursor.moveToFirst()) {  
 val name = cursor.getString(cursor.getColumnIndexOrThrow("name"))  
 val age = cursor.getInt(cursor.getColumnIndexOrThrow("age"))  
 return User(name, age)  
 } else {  
 throw IllegalArgumentException("No user with ID $id")  
 }  
 }  
  
 fun updateUser(user: User) {  
 val db = this.writableDatabase  
 val values = ContentValues().apply {  
 put("name", user.name)  
 put("age", user.age)  
 }  
 db.update("users", values, "id = ?", arrayOf(user.id.toString()))  
 }  
  
 fun deleteById(id: Int): Boolean {  
 val db = this.writableDatabase  
 val success = db.delete(TABLE\_NAME, "$COL\_ID=?", arrayOf(id.toString()))  
 db.close()  
 return Integer.parseInt("$success") != -1  
 }  
}

Листинг 9 – product.kt

package com.example.myapplication\_lab\_4  
  
class Product {  
 var id = 0  
 var name = ""  
 var price = 0  
  
 constructor() {}  
  
 constructor(name: String, price: Int) {  
 this.name = name  
 this.price = price  
 }  
  
 constructor(id: Int, name: String, price: Int) {  
 this.id = id  
 this.name = name  
 this.price = price  
 }  
}

Листинг 10 – UserAdapter.kt

package com.example.lab\_3  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
  
class UserAdapter(private var users: List<User>) : RecyclerView.Adapter<UserAdapter.UserViewHolder>() {  
 var onItemClick : ((User) -> Unit)? = null  
 var onItemLongClick : ((User) -> Unit)? = null  
 var selectedPosition = -1  
  
 class UserViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val textView: TextView = itemView.findViewById(R.id.text\_view)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): UserViewHolder {  
 val view = LayoutInflater.from(parent.context).inflate(R.layout.list\_item, parent, false)  
 return UserViewHolder(view)  
 }  
  
 override fun getItemCount(): Int {  
 return users.size  
 }  
  
 override fun onBindViewHolder(holder: UserViewHolder, position: Int) {  
 val user = users[position]  
 holder.textView.text = "${user.name}, ${user.age}"  
  
 holder.itemView.setOnClickListener {  
 onItemClick?.invoke(user)  
 }  
  
 holder.itemView.setOnLongClickListener {  
 selectedPosition = position  
 onItemLongClick?.invoke(user)  
 false  
 }  
 }  
  
 fun getSelectedUser(): User {  
 return users[selectedPosition]  
 }  
  
 fun updateUsers(newUsers: List<User>) {  
 this.users = newUsers  
 notifyDataSetChanged()  
 }  
}

Листинг 11 – DBHandlerSecond.kt

package com.example.myapplication\_lab\_4  
  
import android.content.Context  
import android.content.ContentValues  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
import android.widget.Toast  
  
const val DATABASE\_NAME\_second = "MyDB2"  
const val SECOND\_TABLE\_NAME = "MyProducts"  
  
class DBHandlerSecond(var context: Context) : SQLiteOpenHelper(context, DATABASE\_NAME, null, 1) {  
 override fun onCreate(db: SQLiteDatabase?) {  
 val secondCreateTable = "CREATE TABLE " + SECOND\_TABLE\_NAME +" (" +  
 COL\_ID +" INTEGER," +  
 COL\_NAME + " VARCHAR(256)," +  
 COL\_PRICE +" INTEGER)"  
  
 db?.execSQL(secondCreateTable)  
  
  
 if (db != null) {  
 insertStaticData(db)  
 }  
 }  
  
  
 private fun insertStaticData(db: SQLiteDatabase) {  
// val users = listOf(  
// Product("Apple", 24),  
// Product("Bananas", 22),  
// Product("Potato", 20),  
// Product("Tomato", 17),  
// Product("grape", 14)  
// )  
// for (user in users) {  
// val cv = ContentValues()  
// cv.put(COL\_NAME, user.name)  
// cv.put("price", user.price)  
// db.insert(TABLE\_NAME, null, cv)  
// }  
 val cv = ContentValues()  
 cv.put(COL\_ID, 60)  
 cv.put(COL\_NAME, "first")  
 cv.put("price", 1)  
 db.insert(SECOND\_TABLE\_NAME, null, cv)  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")  
 onCreate(db)  
 }  
  
 fun insertData(user: Product) {  
 val db = this.writableDatabase  
 var cv = ContentValues()  
 cv.put(COL\_NAME, user.name)  
 cv.put(COL\_PRICE, user.price)  
 var result = db.insert(TABLE\_NAME, null, cv)  
  
 if (result == (-1).toLong()) {  
 Toast.makeText(context, "Failed", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(context, "Success", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 fun insertDataSecond(product: Product) {  
 val db = this.writableDatabase  
 var cv = ContentValues()  
 cv.put(COL\_ID, product.id)  
 cv.put(COL\_NAME, product.name)  
 cv.put(COL\_PRICE, product.price)  
 var result = db.insert(SECOND\_TABLE\_NAME, null, cv)  
  
 if (result == (-1).toLong()) {  
 Toast.makeText(context, "Failed", Toast.LENGTH\_SHORT).show()  
 } else {  
 Toast.makeText(context, "Success", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 fun readData() : MutableList<Product>{  
 var list : MutableList<Product> = ArrayList()  
  
 val db = this.readableDatabase  
 val query = "Select \* from $TABLE\_NAME"  
 val result = db.rawQuery(query,null)  
 if(result.moveToFirst()){  
 do {  
 var user = Product()  
 user.id = result.getString(result.getColumnIndexOrThrow(COL\_ID)).toInt()  
 user.name = result.getString(result.getColumnIndexOrThrow(COL\_NAME))  
 user.price = result.getString(result.getColumnIndexOrThrow(COL\_PRICE)).toInt()  
 list.add(user)  
 } while (result.moveToNext())  
 }  
  
 result.close()  
 db.close()  
 return list  
 }  
  
 fun readDataSecond() : MutableList<Product>{  
 var list : MutableList<Product> = ArrayList()  
  
 val db = this.readableDatabase  
 val query = "Select \* from $SECOND\_TABLE\_NAME"  
 val result = db.rawQuery(query,null)  
 if(result.moveToFirst()){  
 do {  
 var user = Product()  
 user.id = result.getString(result.getColumnIndexOrThrow(COL\_ID)).toInt()  
 user.name = result.getString(result.getColumnIndexOrThrow(COL\_NAME))  
 user.price = result.getString(result.getColumnIndexOrThrow(COL\_PRICE)).toInt()  
 list.add(user)  
 } while (result.moveToNext())  
 }  
  
 result.close()  
 db.close()  
 return list  
 }  
  
 fun getUser(id: Int): Product {  
 val db = this.readableDatabase  
 val cursor = db.query("users", null, "id = ?",  
 arrayOf(id.toString()), null, null, null)  
  
 if (cursor.moveToFirst()) {  
 val name = cursor.getString(cursor.getColumnIndexOrThrow("name"))  
 val price = cursor.getInt(cursor.getColumnIndexOrThrow("price"))  
 return Product(name, price)  
 } else {  
 throw IllegalArgumentException("No user with ID $id")  
 }  
 }  
  
 fun updateUser(user: Product) {  
 val db = this.writableDatabase  
 val values = ContentValues().apply {  
 put("name", user.name)  
 put("price", user.price)  
 }  
 db.update("users", values, "id = ?", arrayOf(user.id.toString()))  
 }  
  
 fun deleteById(id: Int): Boolean {  
 val db = this.writableDatabase  
 val success = db.delete(SECOND\_TABLE\_NAME, "$COL\_ID=?", arrayOf(id.toString()))  
 db.close()  
 return Integer.parseInt("$success") != -1  
 }  
}

Листинг 12 – ListDBActivity.kt

package com.example.myapplication\_lab\_4  
  
import android.app.Activity  
import android.content.Intent  
import android.graphics.Rect  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.ContextMenu  
import android.view.MenuItem  
import android.view.View  
import android.widget.ArrayAdapter  
import android.widget.ListView  
import android.widget.Toast  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
  
class ListDBActivity : AppCompatActivity() {  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var productAdapter: ProductAdapter  
 private lateinit var dbHandlersecond: DBHandlerSecond  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_list\_dbactivity)  
 val callingActivity = intent.getStringExtra("callingActivity")  
 Toast.makeText(this, "Вызвала: $callingActivity", Toast.LENGTH\_SHORT).show()  
  
 dbHandlersecond = DBHandlerSecond(this)  
 //dbHandlersecond.onCreate(dbHandlersecond.writableDatabase)  
 val stationsArray = dbHandlersecond.readDataSecond()  
  
 recyclerView = findViewById(R.id.recycler\_view)  
 recyclerView.setHasFixedSize(true)  
 recyclerView.layoutManager = LinearLayoutManager(this)  
  
 productAdapter = ProductAdapter(stationsArray)  
 recyclerView.adapter = productAdapter  
  
 productAdapter.onItemClick = {  
 Toast.makeText(this, "selected: ${it.name}", Toast.LENGTH\_LONG).show()  
 //Toast.makeText(applicationContext, selectedItem, Toast.LENGTH\_LONG).show()  
 val resultIntent = Intent().apply {  
 putExtra("fromActivity", "EditActivity")  
 putExtra("name", it.name)  
 }  
 setResult(Activity.RESULT\_OK, resultIntent)  
 finish()  
 }  
  
 recyclerView.addItemDecoration(object : RecyclerView.ItemDecoration() {  
 override fun getItemOffsets(outRect: Rect, view: View, parent: RecyclerView, state: RecyclerView.State) {  
 super.getItemOffsets(outRect, view, parent, state)  
  
 val position = parent.getChildAdapterPosition(view)  
 val isLastItem = position == state.itemCount - 1  
  
// if (!isLastItem) {  
// outRect.bottom = 10.dpToPx(parent.context)  
// }  
 }  
 })  
 val products = mutableListOf<String>()  
 for(product in stationsArray){  
 products.add(product.name)  
 }  
  
// val adapter = ArrayAdapter(this, android.R.layout.simple\_list\_item\_1, products)  
//  
// val listView = ListView(this)  
// listView.adapter = adapter  
//  
// setContentView(listView)  
//  
// listView.setOnItemClickListener { parent, view, position, id ->  
// val selectedItem = parent.getItemAtPosition(position).toString()  
// //Toast.makeText(applicationContext, selectedItem, Toast.LENGTH\_LONG).show()  
// val resultIntent = Intent().apply {  
// putExtra("fromActivity", "EditActivity")  
// putExtra("name", selectedItem)  
// }  
// setResult(Activity.RESULT\_OK, resultIntent)  
// finish()  
// }  
  
 registerForContextMenu(recyclerView)  
 }  
 override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo: ContextMenu.ContextMenuInfo?) {  
 super.onCreateContextMenu(menu, v, menuInfo)  
 menu.add(0, v.id, 0, "Удалить")  
 }  
  
 override fun onContextItemSelected(item: MenuItem): Boolean {  
 if (item.title == "Удалить") {  
 Toast.makeText(this, "delete", Toast.LENGTH\_SHORT).show()  
 val user = productAdapter.getSelectedUser()  
 dbHandlersecond.deleteById(user.id)  
  
 val users = dbHandlersecond.readDataSecond()  
 productAdapter.updateUsersSecond(users)  
 }  
 return super.onContextItemSelected(item)  
 }  
}

Листинг 13 – activity\_list\_dbactivity.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ListDBActivity">  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycler\_view"  
 android:layout\_width="398dp"  
 android:layout\_height="466dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

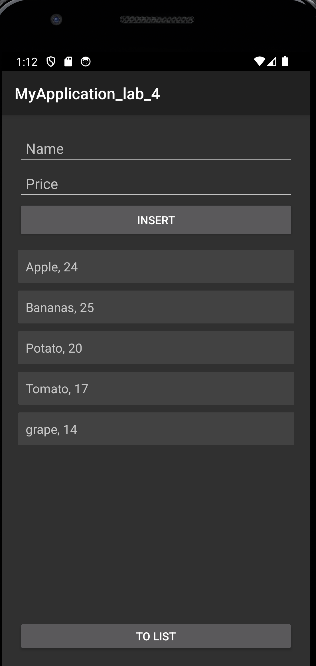


Рисунок 1 – Демонстрация главной активности

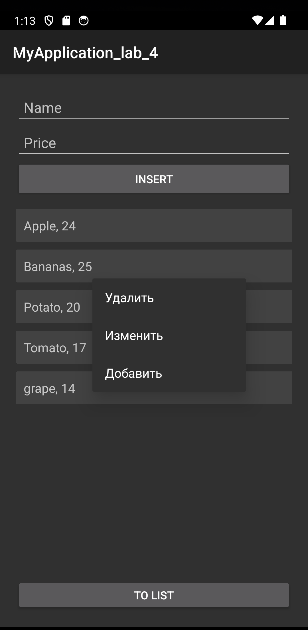


Рисунок 2 – Демонстрация меню

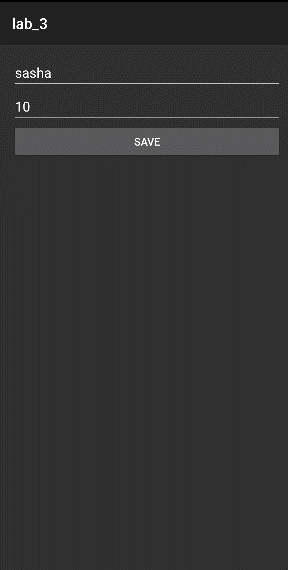


Рисунок 3 – Демонстрация активности изменения данных

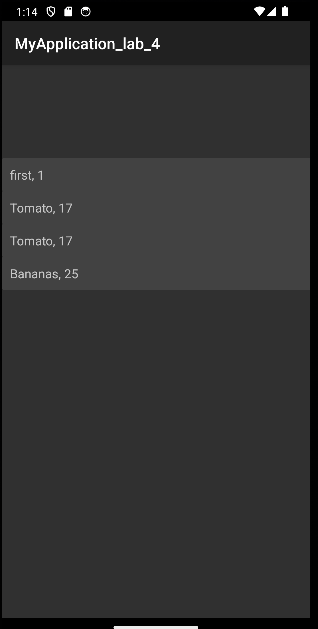
****

Рисунок 3 – Демонстрация корзины

Вывод: В результате выполнения лабораторной работы я научился вызывать Активность с использованием явного намерения и получать результаты её работы. Научился использовать неявные Намерения и получать данные из Намерения.