**Виконавець:** Момот Роман КН-919а

**ЛАБОРАТОРНА РОБОТА 7**

**Робота з базами даних**

***Мета:*** Навчитись розробляти застосунки із використанням фрагментів та використовувати налаштування.

***Варіант:*** 16

TaskNumber = (16 mod 10) + 1 = 7

***Індивідуальне завдання:*** У розмітці активності не повинно бути закодованих чисел та констант. Усі дані повинні братись із ресурів @color/your\_id, @drawable/your\_id, @dimen/your\_id, @string/your\_id.

Створити застосунок з однією таблицею, яка складається з двох полів: номер автомобіля та рік випуску авто. Заповнити таблицю за допомогою генератора випадкових чисел, не менше п’яти рядків. Вивести значення таблиці. Вивести рік випуску авто за його номером.

**Хід виконнаня роботи**

1. Створюємо активність для роботи з базою даних та 2 окремих класи для генерації випадкових даних для заповнення таблиці. Активність на початку підключається або створює базу даних, перевіряє чи пуста таблиця і в цьому випадку заповнює її випадковими даними. Після цього отримує з таблиці дані та виводить їх на екран (рис 7.1).

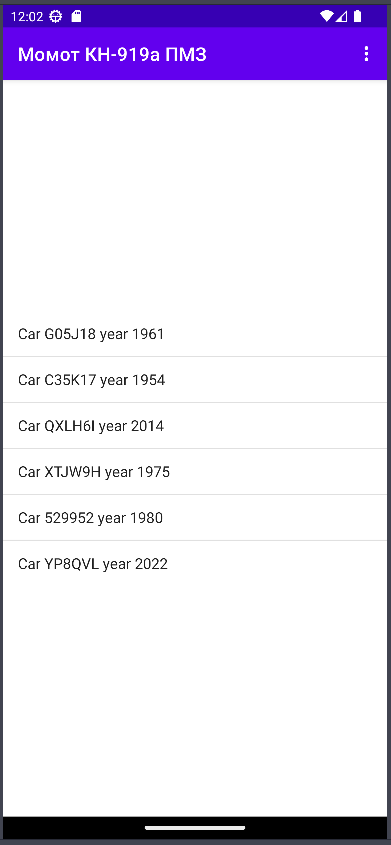


Рисунок 7.1. Отримані дані з таблиці

**Код програми**

**MainActivity.java**

package com.ntukhpi.otp.momot.fourth\_course\_dma.lab7;

import android.content.ContentValues;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import com.ntukhpi.otp.momot.fourth\_course\_dma.lab1.R;

import com.ntukhpi.otp.momot.fourth\_course\_dma.lab7.util.CarNumberGenerator;

import com.ntukhpi.otp.momot.fourth\_course\_dma.lab7.util.YearGenerator;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class MainActivity extends AppCompatActivity {

private SQLiteDatabase database;

Map<String, String> carsData;

private static final int NUMBER\_OF\_NEW\_LINES = 6;

private static final String TABLE\_NAME = "cars";

private static final String CAR\_NUMBER = "car\_number";

private static final String CAR\_PRODUCTION\_YEAR = "production\_year";

private static final String GET\_DATA\_FROM\_TABLE\_COMMAND = "SELECT \* FROM cars;";

private static final String GET\_TABLE\_SIZE\_COMMAND = "SELECT COUNT(\*) FROM cars;";

private static final String CLEAR\_TABLE = "DELETE FROM cars;";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.lab7\_activity\_main);

database = getBaseContext().openOrCreateDatabase("cars.db", MODE\_PRIVATE, null);

database.execSQL("CREATE TABLE IF NOT EXISTS cars (car\_number TEXT NOT NULL UNIQUE , production\_year TEXT NOT NULL);");

if (getTableSize() < 5) {

fillTable();

} else {

clearTable();

}

carsData = new HashMap<>();

getDataFromTable();

outputToView();

database.close();

}

private void clearTable() {

database.execSQL(CLEAR\_TABLE);

}

private int getTableSize() {

Cursor query = database.rawQuery(GET\_TABLE\_SIZE\_COMMAND, null);

int number = 0;

while (query.moveToNext()) {

number = query.getInt(0);

}

query.close();

return number;

}

private void fillTable() {

ContentValues contentValues = new ContentValues();

for (int i = 0; i < NUMBER\_OF\_NEW\_LINES; i++) {

contentValues.put(CAR\_NUMBER, CarNumberGenerator.generateCarNumber());

contentValues.put(CAR\_PRODUCTION\_YEAR, YearGenerator.getRandomYear());

database.insert(TABLE\_NAME, null, contentValues);

}

}

private void outputToView() {

ListView carListView = findViewById(R.id.lab7\_cars\_list);

List<String> temp = new ArrayList<>();

for (Map.Entry<String, String> entry : carsData.entrySet()) {

temp.add(String.format("Car %s year %s", entry.getKey(), entry.getValue()));

}

ArrayAdapter<?> adapter = new ArrayAdapter(this,

android.R.layout.simple\_list\_item\_1, temp);

carListView.setAdapter(adapter);

}

private void getDataFromTable() {

Cursor query = database.rawQuery(GET\_DATA\_FROM\_TABLE\_COMMAND, null);

while (query.moveToNext()) {

String number = query.getString(0);

String year = query.getString(1);

carsData.put(number, year);

}

query.close();

}

}

**CarNumberGenerator.java**

package com.ntukhpi.otp.momot.fourth\_course\_dma.lab7.util;

public class CarNumberGenerator {

private static final int CAR\_NUMBER\_LENGTH = 6;

private CarNumberGenerator() {}

public static String generateCarNumber() {

StringBuilder carNumber = new StringBuilder(CAR\_NUMBER\_LENGTH);

for (int i = 0; i < CAR\_NUMBER\_LENGTH; i++) {

carNumber.append(Math.round( Math.random()) == 1 ? generateDigit() : generateLetters());

}

return carNumber.toString();

}

private static char generateDigit() {

int n = '9' - '0' + 1;

return (char) ('0' + Math.random() \* n);

}

private static char generateLetters() {

int n = 'Z' - 'A' + 1;

return (char) ('A' + Math.random() \* n);

}

}

**YearGenerator.java**

package com.ntukhpi.otp.momot.fourth\_course\_dma.lab7.util;

public class YearGenerator {

private static final int MIN\_YEAR = 1950;

private static final int MAX\_YEAR = 2023;

private YearGenerator() {}

public static int getRandomYear() {

return (int) (MIN\_YEAR + Math.round(Math.random() \* (MAX\_YEAR - MIN\_YEAR)));

}

}

**Lab7\_activity\_man.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="com.ntukhpi.otp.momot.fourth\_course\_dma.lab7.MainActivity">  
  
 <ListView  
 android:id="@+id/lab7\_cars\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
 </ListView>  
</androidx.constraintlayout.widget.ConstraintLayout>

**Висновок**

У результаті виконання лабораторної роботи було ознайомлено з роботою з базою даних. Був створений клас, який додає та отримує з таблиці дані.