Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

Геолокационные возможности

Выполнил:

студент 4 курса

группы АС-56

Марук К.В.

Проверил:

Козинский А.А.

Брест 2022

**Аннотация:** Разработка приложения, демонстрирующего геолокационные возможности.

**Цель лабораторной работы:** Разработка приложения, демонстрирующего геолокационные возможности.

**Задачи лабораторной работы:**

* Разработать приложение, получающее координаты устройства и отслеживающее их изменения.

Бонусы (то, что способствует оценке выше 4)

* Справка по приложению, наличие сценария
* Использование различных функций из библиотеки для определения положения, расстояния…
* Присутствие возможности сохранения истории в базе данных (возможны различные форматы)

**Код программы:**

MainActivity.java

public class MainActivity extends Activity {  
 TextView tvOut, tvLon, tvLat;  
 public TextView tvPlace;  
 ImageButton help;  
 Button listSensorsBtn;  
 Button openDB;  
 SensorManager sensorManager;  
 Sensor sensorLight;  
 DBHelper dbHelper;  
 ContentValues cv;  
 SQLiteDatabase db;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 tvOut = findViewById(R.id.*textView1*);  
 tvLon = findViewById(R.id.*latitude*);  
 tvLat = findViewById(R.id.*longitude*);  
 help = findViewById(R.id.*imageButton*);  
 openDB = findViewById(R.id.*button2*);  
 listSensorsBtn = findViewById(R.id.*button*);  
 tvPlace = findViewById(R.id.*textView6*);  
 dbHelper = new DBHelper(this);  
 db = dbHelper.getWritableDatabase();  
 LocationManager mlocManager = (LocationManager) getSystemService(Context.*LOCATION\_SERVICE*);  
 LocationListener mlocListener = new LocationListener() {  
 public void onLocationChanged(Location location) {  
 tvLat.setText(String.*valueOf*(location.getLatitude()));  
 tvLon.setText(String.*valueOf*(location.getLongitude()));  
 cv = new ContentValues();  
 cv.put("latitude",String.*valueOf*(tvLat.getText()));  
 cv.put("longitude",String.*valueOf*(tvLon.getText()));  
 db.insert("mytable",null,cv);  
 }  
 public void onStatusChanged(String provider, int status, Bundle extras) {}  
 public void onProviderEnabled(String provider) {}  
 public void onProviderDisabled(String provider) {}  
 };  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 return;}  
 mlocManager.requestLocationUpdates  
 (LocationManager.*NETWORK\_PROVIDER*, 0, 0, mlocListener);  
 if (mlocManager.isProviderEnabled(LocationManager.*NETWORK\_PROVIDER*))  
 tvOut.setText("Подключено");  
 else tvOut.setText("Нет подключения" );  
 help.setOnClickListener(v -> CallAboutInfo());  
 listSensorsBtn.setOnClickListener(v -> CallListOfSensors());  
 sensorManager = (SensorManager) getSystemService(*SENSOR\_SERVICE*);  
 sensorLight = sensorManager.getDefaultSensor(Sensor.*TYPE\_LIGHT*);  
 sensorManager.registerListener(listenerLight, sensorLight,  
 SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 openDB.setOnClickListener(v -> readDB(db));  
 }  
 SensorEventListener listenerLight = new SensorEventListener() {  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
 }  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 tvPlace.setText(String.*valueOf*(event.values[0]));  
 }  
 };  
 public void CallAboutInfo(){  
 Intent intent = new Intent(this, About.class);  
 startActivity(intent);  
 }  
 public void CallListOfSensors(){  
 Intent intent = new Intent(this, AllSensors.class);  
 startActivity(intent);  
 }  
 public void readDB(SQLiteDatabase db){  
 Log.*d*("Logs", "--- Rows in mytable: ---");  
 Cursor c = db.query("mytable", null, null, null, null, null, null);  
 if (c.moveToFirst()) {  
 int idColIndex = c.getColumnIndex("id");  
 int nameColIndex = c.getColumnIndex("latitude");  
 int emailColIndex = c.getColumnIndex("longitude");  
 do {  
 Log.*d*("Logs",  
 "ID = " + c.getInt(idColIndex) +  
 ", latitude = " + c.getString(nameColIndex) +  
 ", longitude = " + c.getString(emailColIndex));  
 } while (c.moveToNext());  
 } else Log.*d*("Logs", "0 rows");  
 c.close();  
 }  
 class DBHelper extends SQLiteOpenHelper {  
 public DBHelper(Context context) {  
 super(context, "myDB", null, 1);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 Log.*d*("Logs", "--- onCreate database ---");  
 db.execSQL("create table mytable ("+"id integer primary key autoincrement," + "\"latitude\" text,"+ "\"longitude\" text" + ");");}  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {}  
 }  
}

Вывод: Разработал приложение демонстрирующее геолокационные возможности