Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №5
на тему «Дослідження роботи з вбудованими датчиками»
З дисципліни «Розробка мобільних застосувань під Android»

Виконала:
студентка 3 курсу
групи ІС-21
Бєльчик Софія

Київ-2025

Мета роботи - ознайомитись з можливостями вбудованих датчиків мобільних пристроїв та дослідити способи їх використання для збору та обробки даних.

Варіант - 14

Завдання:Написати програму під платформу Андроїд, яка має інтерфейс для виведення даних з обраного вбудованого датчика (тип обирається самостійно, можна відслідковувати зміни значень і з декількох датчиків).
Посилання гітхаб:https://github.com/SofiaBielchik/task5.git

Кілька скріншотів із емулятора Android, у якому вручну змінювали положення акселерометра через Extended Controls → Additional sensors → Accelerometer:

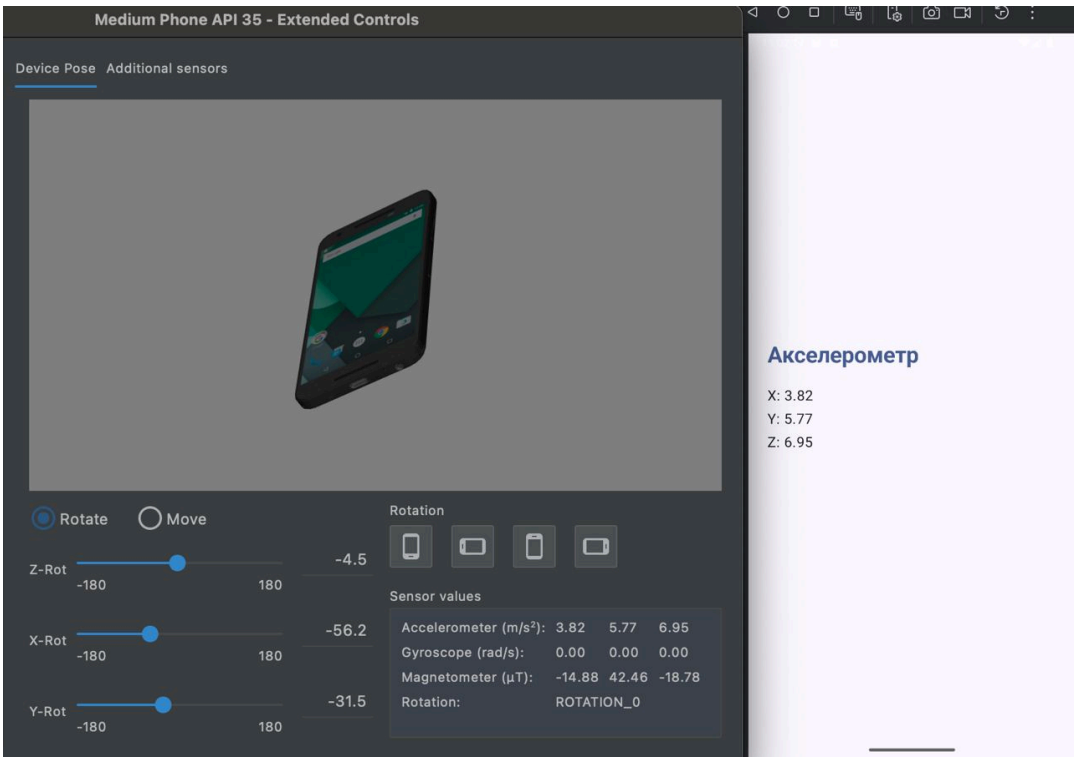

Рис. 1 Положення емулятора і показники акселерометра

- Ліворуч видно Extended Controls емулятора (вкладка Device Pose → Accelerometer).

- Значення Accelerometer (m/s²) показують:
  X: 3.82, Y: 5.77, Z: 6.95
- Праворуч — інтерфейс самого застосунку (MainActivity).
  - Поле “Акселерометр” (заголовок).
  - Значення X: 3.82, Y: 5.77, Z: 6.95 — вони точно відповідають показникам у емуляторі.

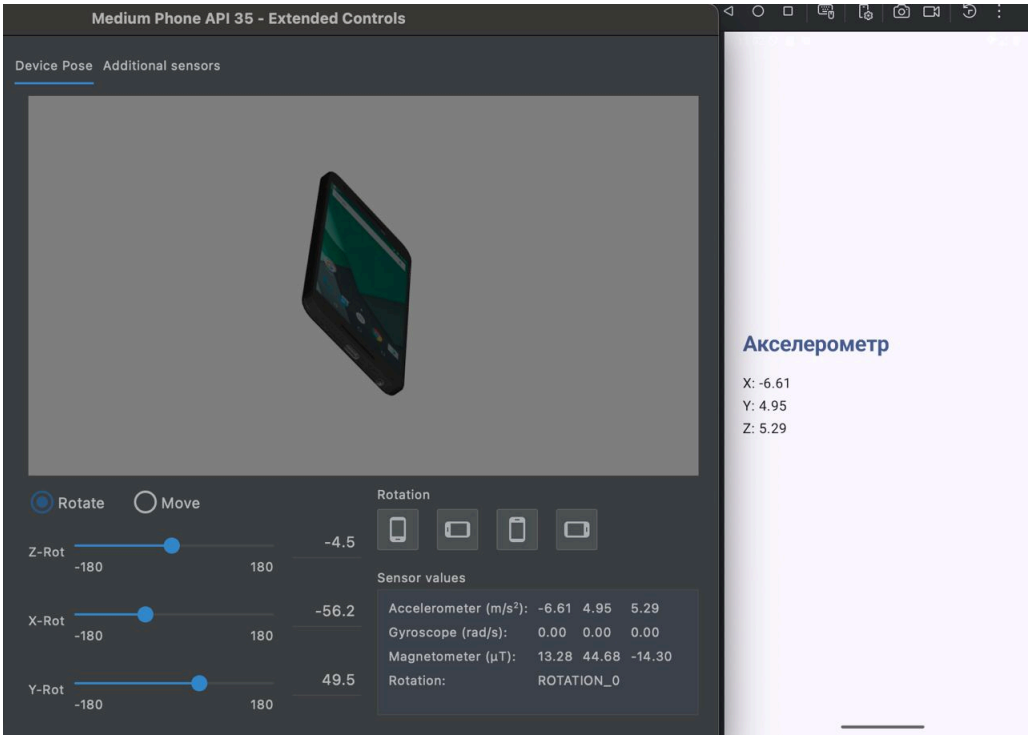

Рис. 2 Зміна положення емулятора (нахилатель по осі X, Y, Z)

- Ми змістили положення емулятора (нахилили девайс у іншому напрямку).
- Extended Controls показують:
  Accelerometer: X: -6.61, Y: 4.95, Z: 5.29
- Праворуч — використовуючи метод onSensorChanged, програма оновила TextView → відображено:
  X: -6.61, Y: 4.9, Z: 5.29

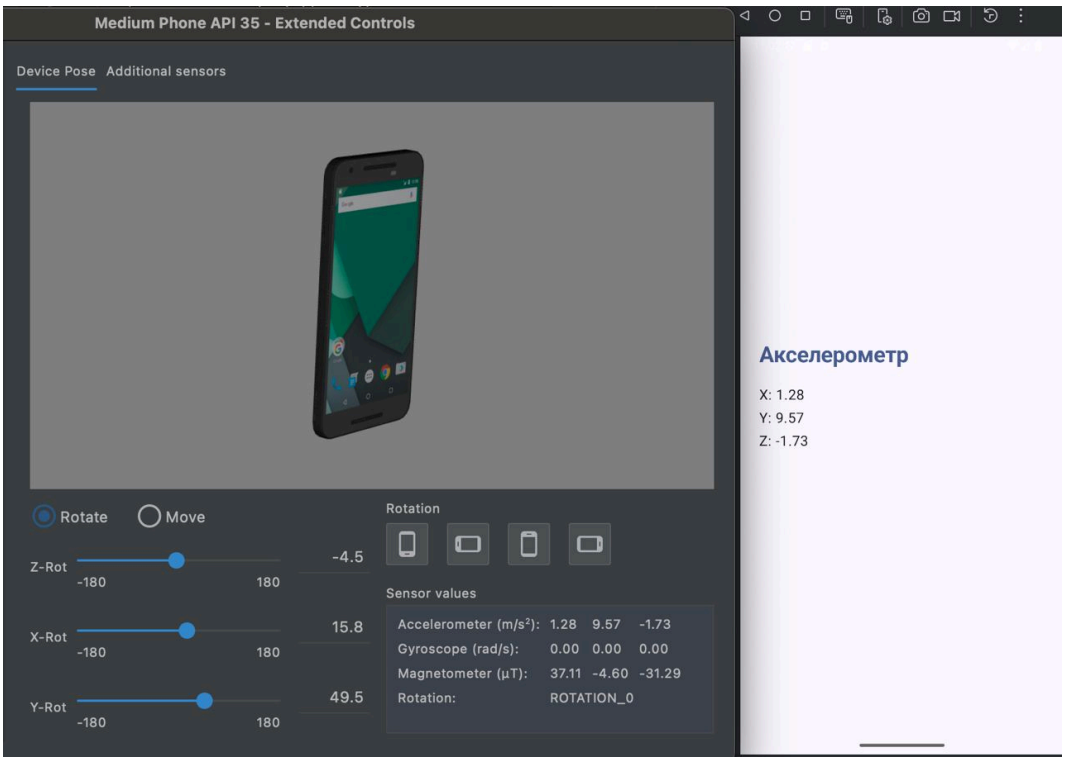

Рис. 3 Інше положення емулятора (ще один нахил)

- Змінили положення емулятора повторно.
- Extended Controls показують:
  Accelerometer: X: 1.28, Y: 9.57, Z: -1.73
- UI програми оновився автоматично та показує відповідні нові значення.

Таким чином, відбувається миттєва реакція застосунку на зміну показників датчика — це і є головна ідея роботи з вбудованими сенсорами в Android.

Висновок:

У лабораторній роботі № 5 було:

- Реалізовано Android-застосунок, який у режимі реального часу читає дані з акселерометра через SensorManager і SensorEventListener.
- Налаштовано UI із трьома полями (X, Y, Z), у яких показуються актуальні величини прискорення по трьох осях.

- Продемостровано взаємодію із емулятором Android Extended Controls → Additional sensors → Accelerometer: при зміні положення емулятора відповідні значення показів без затримки оновлювалися у застосунку.

Ця робота дала практичні навички опитування сенсорів, обробки та відображення отриманих даних, а також підкреслила важливість правильного керування ресурсами (реєстрація/відписка) у мобільних застосунках.

Контрольні запитання:

1. Наведіть приклади вбудованих датчиків та величини, які з них можна зчитати

1. Акселерометр (Accelerometer, TYPE_ACCELEROMETER)
    - Величини: прискорення по осях X, Y, Z (в м/с²), включаючи силу гравітації.
2. Гіроскоп (Gyroscope, TYPE_GYROSCOPE)
    - Величини: кутова швидкість (обертання) навколо осей X, Y, Z (в радіанах/с).
3. Магнітометр (Magnetic Field, TYPE_MAGNETIC_FIELD)
    - Величини: сила магнітного поля по осях X, Y, Z (в мікротеслах, μT).
4. Світлочутливий датчик (Light, TYPE_LIGHT)
    - Величина: інтенсивність освітлення (в люксах, lx).
5. Датчик наближення (Proximity, TYPE_PROXIMITY)
    - Величина: відстань до об’єкта перед сенсором (в сантиметрах).
6. Тиск (Pressure, TYPE_PRESSURE)
    - Величина: атмосферний тиск (в гектопаскалях, hPa).
7. Температура (Ambient Temperature, TYPE_AMBIENT_TEMPERATURE)
    - Величина: температура навколишнього середовища (в °C).
8. Датчик нахилу (Rotation Vector, TYPE_ROTATION_VECTOR)
    - Величини: вектор обертання пристрою (кватерніони або векторні компоненти, що перетворюються на кути, азимут, тангаж, крен).
9. Датчик кроків (Step Counter, TYPE_STEP_COUNTER)

- Величина: кількість зроблених кроків від останнього завантаження пристрою.

10. Датчик гравітації (Gravity, TYPE_GRAVITY)
    - Величини: вектор гравітації по осях X, Y, Z (в м/с²).

2. Наведіть особливості роботи з вбудованими датчиками

1. Реєстрація і відписка

Щоб отримувати показники, потрібно зареєструвати SensorEventListener у SensorManager (часто в onResume()):

```
sensorManager.registerListener(
   listener,
   sensor,
   SensorManager.SENSOR_DELAY_NORMAL
)
```

Після закриття Activity (у onPause()) необхідно обов'язково викликати sensorManager.unregisterListener(listener) щоб зупинити оновлення та зекономити батарею.

2. Режими оновлення (delay)

    - SENSOR_DELAY_FASTEST (≈ 0 μc) — максимальна частота.
    - SENSOR_DELAY_GAME (≈ 20 000 μc) — для ігор (≈50 Гц).
    - SENSOR_DELAY_UI (≈ 60 000 μc) — для UI (≈16 Гц).
    - SENSOR_DELAY_NORMAL (≈ 200 000 μc) — за замовчуванням (≈5 Гц).
    - Вибір режиму впливає на частоту оновлення і, відповідно, на енергоспоживання.

3. Обробка onSensorChanged()
    - Метод викликається дуже часто (залежно від вибраного delay).
    - Не рекомендується виконувати важкі обчислення безпосередньо тут — краще відкладати в окремий потік або використовувати фільтрацію/згладжування (low-pass filter), щоб уникнути стрибків значень і значного навантаження на UI-потік.
4. Перевірка доступності датчика

Не всі пристрої мають кожен датчик. Тому необхідно перед реєстрацією перевіряти, чи датчик не дорівнює null:

```
val accel =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
if (accel == null) {
   // повідомити, що датчик недоступний
}
```

5. Фільтрування шуму
    - Сировинні (raw) дані часто «шумні». Для більш плавних показників радять застосовувати фільтри (low-pass, high-pass), щоб відокремити гравітаційну складову або видалити короткі сплески.

6. Вплив фонових процесів
    - Якщо застосунок працює у фоні й продовжує отримувати дані від датчика, це може сильно витрачати заряд батареї. Тому потрібне обережне дозування реєстрації (тільки тоді, коли UI активний) або використання JobScheduler/WorkManager для рідкісних зчитувань.

7. Поєднання з іншими датчиками
    - Часто для коректного отримання орієнтації пристрою використовують комбінацію акселерометра та магнітометра або спеціальний датчик Rotation Vector.
    - Наприклад, Sensor.TYPE_ROTATION_VECTOR дає вже оброблену інформацію (кватерніон), щоб не комбінувати дані вручну.

8. Точність (accuracy)
    - Для деяких датчиків (магнітометра, гіроскопа) можна отримати повідомлення про зміну точності (onAccuracyChanged(...)). Це може знаходитись у випадках сильного магнітного поля чи калібрування сенсора.

9. Категорії датчиків
    - Hardware/back-end датчики: фізичні сенсори (акселерометр, гіроскоп, магнітометр, барометр тощо).

- Software/back-end датчики (наближення, обертання) — алгоритмічно поєднані фізичні значення (наприклад, TYPE_ROTATION_VECTOR чи TYPE_GRAVITY, які є обробленими даними з кількох “hardware” датчиків).
- Мережеві/віртуальні: наприклад, Significant Motion Sensor (виявляє різку зміну руху) або Step Detector (виявляє один крок).

10. Правила використання
    - Використовуйте тільки ті датчики, які справді потрібні для функціоналу.
    - Уникайте надмірного споживання батареї, обираючи відповідний delay та відписуючись, коли датчика не потрібно.
    - Якщо датчик відсутній на пристрої — реалізуйте коректний fallback (наприклад, вимкніть функціонал, який його потребує).