

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря
Сікорського» Факультет інформатики та
обчислювальної техніки Кафедра інформатики
та програмної інженерії

Звіт
до лабораторної роботи № 5 з дисципліни
«Розробка мобільних застосунків під Android»

Виконала ІК-24 Юхимець Л.

Перевірів Орленко С.П.

Київ 2025

Лабораторна робота № 5

Мета роботи

Ознайомитись з можливостями вбудованих датчиків мобільних пристроїв та дослідити способи їх використання для збору та обробки даних.

Завдання

БАЗОВЕ (10/20 балів). 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).

ПОВНЕ (20/20). Функціональність базового додатку додатково розширюється обробкою отриманих даних та виведенням їх у відповідній формі.

MainActivity.kt

```
package com.example.lab5

import android.content.pm.ActivityInfo
import android.hardware.*
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), SensorEventListener {

    private lateinit var sensorManager: SensorManager
    private var accelerometer: Sensor? = null
    private var magnetometer: Sensor? = null

    private lateinit var compassView: CompassView
    private lateinit var degreeText: TextView
    private lateinit var directionHistoryText: TextView
    private lateinit var warningText: TextView
    private lateinit var timeText: TextView

    private var gravity = FloatArray(3)
    private var geomagnetic = FloatArray(3)

    private val directionHistory = mutableListOf<String>()
    private var lastDirection: Float = 0f

    private var timeInDirection: Long = 0
    private var timerRunning: Boolean = false
```

```

private var lastTimeChecked: Long = 0

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    degreeText = findViewById(R.id.degreeText)
    compassView = findViewById(R.id.compassView)
    directionHistoryText =
findViewById(R.id.directionHistoryText)
    warningText = findViewById(R.id.warningText)
    timeText = findViewById(R.id.timeText)

    requestedOrientation =
ActivityInfo.SCREEN_ORIENTATION_PORTRAIT

    sensorManager = getSystemService(SENSOR_SERVICE) as
SensorManager
    accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
    magnetometer =
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD)
}

override fun onResume() {
    super.onResume()
    accelerometer?.also {
        sensorManager.registerListener(this, it,
SensorManager.SENSOR_DELAY_UI)
    }
    magnetometer?.also {
        sensorManager.registerListener(this, it,
SensorManager.SENSOR_DELAY_UI)
    }
}

override fun onPause() {
    super.onPause()
    sensorManager.unregisterListener(this)
}

override fun onSensorChanged(event: SensorEvent?) {
    if (event == null) return

    when (event.sensor.type) {
        Sensor.TYPE_ACCELEROMETER -> gravity =
event.values.clone()
    }
}

```

```

        Sensor.TYPE_MAGNETIC_FIELD -> geomagnetic =
event.values.clone()
    }

    val R = FloatArray(9)
    val I = FloatArray(9)
    if (SensorManager.getRotationMatrix(R, I, gravity,
geomagnetic)) {
        val orientation = FloatArray(3)
        SensorManager.getOrientation(R, orientation)
        val azimuth =
Math.toDegrees(orientation[0].toDouble()).toFloat()
        val degree = (azimuth + 360) % 360

        compassView.updateDirection(degree)
        degreeText.text = "${degree.toInt()}°"

        val directionLabel = getDirectionLabel(degree)

        if (directionHistory.isEmpty() ||
directionHistory.last() != directionLabel) {
            directionHistory.add(directionLabel)
            if (directionHistory.size > 5) {
                directionHistory.removeAt(0)
            }
            directionHistoryText.text = "History:
${directionHistory.joinToString(" → ")}"
compassView.updateDirectionHistory(directionHistory)
        }

        val currentTime = System.currentTimeMillis()

        if (directionLabel != getDirectionLabel(lastDirection))
{
            timeInDirection = 0
            lastTimeChecked = currentTime
            timerRunning = true
        }

        if (timerRunning) {
            timeInDirection = (currentTime - lastTimeChecked) /
1000
            timeText.text = "Time in the direction:
$timeInDirection s"
        }

        lastDirection = degree
    }

```

```

    }
}

override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int)
{}

private fun getDirectionLabel(degree: Float): String {
    return when (degree) {
        in 337.5..360.0, in 0.0..22.5 -> "N"
        in 22.5..67.5 -> "NE"
        in 67.5..112.5 -> "E"
        in 112.5..157.5 -> "SE"
        in 157.5..202.5 -> "S"
        in 202.5..247.5 -> "SW"
        in 247.5..292.5 -> "W"
        in 292.5..337.5 -> "NW"
        else -> "?"
    }
}
}

```

CompassView.kt

```
package com.example.lab5
```

```

import android.content.Context
import android.graphics.*
import android.util.AttributeSet
import android.view.View
import kotlin.math.cos
import kotlin.math.sin

class CompassView(context: Context, attrs: AttributeSet?) :
    View(context, attrs) {
    private var direction: Float = 0f
    private val paint = Paint()

    private var directionHistory = listOf<Float>()

    fun updateDirection(dir: Float) {
        direction = dir
        invalidate()
    }

    fun updateDirectionHistory(history: List<String>) {
        directionHistory = history.mapNotNull { labelToDegree(it) }
        invalidate()
    }
}

```

```

private fun labelToDegree(label: String): Float? {
    return when (label) {
        "N" -> 0f
        "NE" -> 45f
        "E" -> 90f
        "SE" -> 135f
        "S" -> 180f
        "SW" -> 225f
        "W" -> 270f
        "NW" -> 315f
        else -> null
    }
}

override fun onDraw(canvas: Canvas) {
    super.onDraw(canvas)

    val cx = width / 2f
    val cy = height / 2f
    val radius = (width.coerceAtMost(height) / 2 *
0.8).toFloat()

    paint.color = Color.LTGRAY
    paint.style = Paint.Style.STROKE
    paint.strokeWidth = 5f
    canvas.drawCircle(cx, cy, radius, paint)

    paint.color = Color.argb(100, 0, 0, 255)
    paint.strokeWidth = 6f
    for (dir in directionHistory) {
        val angleRad = Math.toRadians((-dir).toDouble())
        val hx = (cx + radius * sin(angleRad)).toFloat()
        val hy = (cy - radius * cos(angleRad)).toFloat()
        canvas.drawLine(cx, cy, hx, hy, paint)
    }

    paint.color = Color.RED
    paint.strokeWidth = 10f
    val angle = Math.toRadians((-direction).toDouble())
    val x = (cx + radius * sin(angle)).toFloat()
    val y = (cy - radius * cos(angle)).toFloat()
    canvas.drawLine(cx, cy, x, y, paint)

    paint.color = Color.BLACK
    paint.textSize = 40f

```

```

        paint.style = Paint.Style.FILL
        canvas.drawText("N", cx - 20, cy - radius + 50, paint)
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <com.example.lab5.CompassView
        android:id="@+id/compassView"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerInParent="true"/>

    <TextView
        android:id="@+id/degreeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0°"
        android:textSize="24sp"
        android:layout_below="@id/compassView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"/>

    <TextView
        android:id="@+id/directionHistoryText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="History"
        android:textSize="16sp"
        android:layout_below="@id/degreeText"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:ellipsize="end"
        android:maxLines="1"/>

    <TextView
        android:id="@+id/warningText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textColor="@android:color/holo_red_dark"
        android:textSize="18sp"
        android:layout_below="@id/directionHistoryText"

```

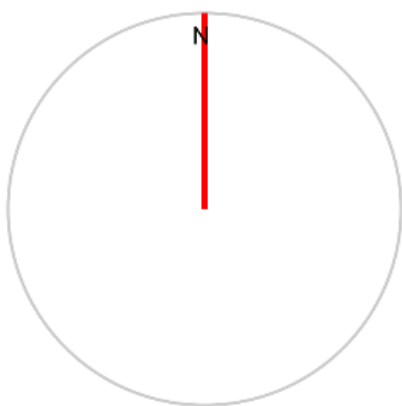
```
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"/>

        <TextView
            android:id="@+id/timeText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Час в напрямку: 0 с"
            android:textSize="18sp"
            android:layout_below="@id/warningText"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"/>
    </RelativeLayout>
```

Скріни виконання програми у віртуальному телефоні

У програмі виконано:

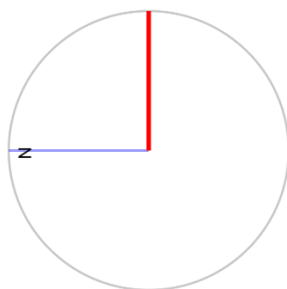
- Візуалізація: Створено компас, який показує напрямок за допомогою сенсорів пристрою. Лінія, що виходить з центру екрану, змінюється при повороті.
- Збереження напрямків: Додано функцію, яка зберігає останні 5 напрямків у списку, візуалізуючи їх на компасі як напівпрозорі лінії. Це дозволяє відстежувати нещодавні напрямки.
- Таймер: Відстежується час, проведений у кожному напрямку. При його зміні таймер скидається для нового напрямку.



0°

History: N

Time in the direction: 0 c

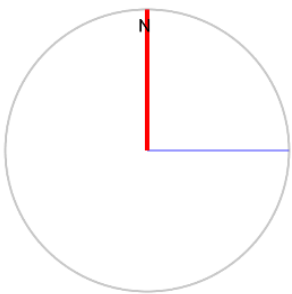


270°

History: N → W

Time in the direction: 18 c



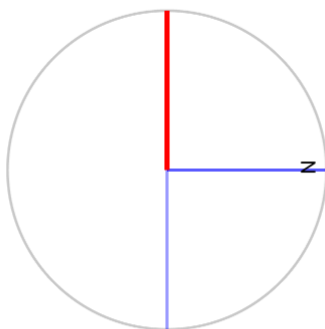


0°

History: N → W → N

Time in the direction: 2 c

Акции
Чтобы
"Парад"



90°

History: N → W → N → E

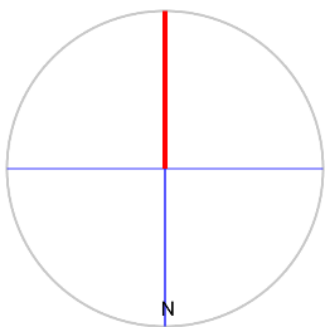
Time in the direction: 2 c



Time in the direction: 1 c

History: N → W → N → E → S

180°



Акти
Чтобы