

ДИСЦИПЛИНА	Программирование корпоративных систем
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические задание
ПРЕПОДАВАТЕЛЬ	Адышкин Сергей Сергеевич
СЕМЕСТР	5 семестр, 2025-2026 гг.

Практическое занятие № 13

Аппаратная часть мобильных устройств. Работа с геолокацией.

Работа с различными датчиками устройства

Цели занятия

- Изучить возможности работы с аппаратными датчиками и сервисами устройства.
- Освоить получение координат устройства с помощью геолокации.
- Научиться определять местоположение, направление и движение.
- Изучить работу с сенсорами: акселерометр, гироскоп, компас, освещенность и др.
- Разработать мобильное приложение, использующее геолокацию и данные сенсоров.

Теоретическая часть

Геолокация как часть аппаратной платформы

Геолокация — это определение текущих координат устройства (широты, долготы и высоты) с использованием одного или нескольких методов:

Метод	Принцип работы	Точность	Задержка	Особенности
GPS	Сигналы спутников	Высокая (до 5 м)	Средняя	Требует открытого неба
Wi-Fi	Анализ ближайших сетей	Средняя (10–50 м)	Низкая	Хорошо в помещениях
Cell ID	Определение по базовым станциям	Низкая	Быстрая	Работает без GPS
Комбинированный	Использует все источники	Средняя	Оптимальная	Используется в Android/iOS API

Работа с геолокацией во Flutter

Flutter предоставляет доступ к геолокации через плагины, которые взаимодействуют с нативными API.

Основные пакеты:

- **geolocator** — получение координат, скорости, направления, статуса разрешений.
- **geocoding** — преобразование координат в адрес (и наоборот).
- **google_maps_flutter / yandex_mapkit** — визуализация карты и отметок.

Разрешения:

Android:

```
<uses-permission  
    android:name="android.permission.ACCESS_FINE_LOCATION"/>  
  
<uses-permission  
    android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

iOS (`Info.plist`):

```
<key>NSLocationWhenInUseUsageDescription</key>
```

```
<string>Приложению требуется доступ к вашему местоположению</string>
```

Работа с сенсорами

Мобильные устройства оснащены множеством датчиков:

Сенсор	Назначение	Пакет для Flutter
Акселерометр	Измеряет ускорение по осям X, Y, Z	sensors_plus
Гироскоп	Измеряет угловое вращение	sensors_plus
Компас	Определяет направление на север	flutter_compass
Датчик освещённости	Измеряет уровень света	light
Шагомер	Считает шаги	pedometer
Барометр	Определяет давление и высоту	flutter_barometer

Принципы работы сенсоров

- Датчики возвращают **поток данных (Stream)** — значения обновляются в реальном времени.
- Эти данные можно обрабатывать, усреднять, фильтровать или визуализировать в UI.
- Работа с сенсорами требует внимания к **энергопотреблению** — частое чтение данных быстро разряжает батарею.

Безопасность и разрешения

- Перед использованием геолокации и некоторых сенсоров **нужно запросить разрешение**.
 - Используется пакет **permission_handler**.
 - Важно корректно обрабатывать ситуацию, если пользователь отклонил запрос.

Практическая часть

Задание

Создать приложение «**Geo & Sensors App**», которое:

1. Определяет текущие координаты пользователя.
2. Показывает адрес по этим координатам (через геокодинг).
3. Отображает в реальном времени данные акселерометра и гироскопа.
4. (Дополнительно) Показывает направление компаса.

Подготовка проекта

```
flutter create geo_sensors_app  
cd geo_sensors_app
```

Установка зависимостей

```
pubspec.yaml:  
dependencies:  
  flutter:  
    sdk: flutter  
  geolocator: ^12.0.0  
  geocoding: ^3.0.0  
  sensors_plus: ^4.0.3  
  flutter_compass: ^0.7.0  
  permission_handler: ^11.3.0
```

Настройка разрешений

AndroidManifest.xml:

```
<uses-permission  
  android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission  
  android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Info.plist (iOS):

```
<key>NSLocationWhenInUseUsageDescription</key>  
<string>Приложению необходимо ваше местоположение</string>
```

Основной код приложения

lib/main.dart:

```
import 'package:flutter/material.dart';
import 'package:geolocator/geolocator.dart';
import 'package:geocoding/geocoding.dart';
import 'package:sensors_plus/sensors_plus.dart';
import 'package:flutter_compass/flutter_compass.dart';

void main() => runApp(const GeoSensorsApp());

class GeoSensorsApp extends StatelessWidget {
    const GeoSensorsApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Geo & Sensors',
            theme: ThemeData(useMaterial3: true),
            home: const GeoSensorsPage(),
        );
    }
}

class GeoSensorsPage extends StatefulWidget {
    const GeoSensorsPage({super.key});
    @override
        State<GeoSensorsPage>      createState() =>
    _GeoSensorsPageState();
}

class _GeoSensorsPageState extends State<GeoSensorsPage> {
    Position? _position;
    String _address = '-';
    double? _compass;
    List<double>? _accelerometerValues;
    List<double>? _gyroscopeValues;

    Future<void> _getLocation() async {
        bool serviceEnabled = await
Geolocator.isLocationServiceEnabled();
        if (!serviceEnabled) return;

        LocationPermission permission = await
Geolocator.requestPermission();
        if (permission == LocationPermission.denied) return;

        final pos = await Geolocator.getCurrentPosition(
            desiredAccuracy: LocationAccuracy.high,
        );
    }
}
```

```
        List<Placemark> placemarks =
            await placemarkFromCoordinates(pos.latitude,
pos.longitude);

        setState(() {
            _position = pos;
            _address =
                '${placemarks.first.locality}, ${
{placemarks.first.street ?? ''}}';
        });
    }

    @override
    void initState() {
        super.initState();

        accelerometerEvents.listen((event) {
            setState(() => _accelerometerValues = [event.x,
event.y, event.z]);
        });

        gyroscopeEvents.listen((event) {
            setState(() => _gyroscopeValues = [event.x, event.y,
event.z]);
        });

        FlutterCompass.events!.listen((event) {
            setState(() => _compass = event.heading);
        });
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: const Text('Geo & Sensors
Demo')),
            body: Padding(
                padding: const EdgeInsets.all(16),
                child: ListView(
                    children: [
                        ElevatedButton(
                            onPressed: _getLocation,
                            child: const Text('Определить
местоположение'),
                        ),
                        const SizedBox(height: 12),
                        Text(
                            _position != null
                                ? 'Координаты: ${_position!.latitude}, ${
{_position!.longitude}}'
                                : 'Координаты: -',
                        ),
                        Text('Адрес: $_address'),
                    ],
                ),
            ),
        );
    }
}
```

```
        const Divider(height: 30),
        Text('Компас: ${_compass?.toStringAsFixed(2) ?? '-'}°'),
        const Divider(height: 30),
        Text(
            'Акселерометр: ${_accelerometerValues?.map((e) => e.toStringAsFixed(2)).join(',') ?? '-'}'),
        Text(
            'Гироскоп: ${_gyroscopeValues?.map((e) => e.toStringAsFixed(2)).join(', ') ?? '-'}'),
        ],
    ),
),
);
}
}
```

Контрольные точки

1. Приложение запускается без ошибок.
2. Определяются координаты устройства.
3. Отображается адрес (через geocoding).
4. При движении телефона меняются данные акселерометра и гироскопа.
5. Компас показывает направление (если поддерживается).

Что сдаётся (отчёт)

- Скриншоты работы приложения:
 1. Отображение координат и адреса.
 2. Реакция сенсоров (акселерометр/гироскоп).
 3. Компас (по возможности).
- Краткий отчет (2–4 стр.):
 - о Цели работы
 - о Используемые пакеты
 - о Основные фрагменты кода
 - о Результаты и выводы

Требования к отчёту

- Формат .docx.
- Структура:
 1. Цель работы
 2. Ход выполнения
 3. Результаты (скриншоты и описание)
 4. Вывод

Контрольные вопросы

1. Какие методы используются для определения геопозиции устройства?
2. Что делает пакет `geolocator` и какие функции он предоставляет?
3. Какие типы сенсоров доступны в современных смартфонах?
4. Чем акселерометр отличается от гироскопа?
5. Какие разрешения нужно добавить для работы с геолокацией?
6. Как можно визуализировать положение пользователя на карте?

9) Дополнительно (+1 балл)

- Отобразите текущее положение пользователя на **карте Google** или **Яндекс**.
- Добавьте **график изменения ускорения во времени** (через `charts_flutter` или `fl_chart`).
- Реализуйте **уведомление**, если устройство сильно ускоряется или поворачивается (на основе данных сенсоров).
- Сделайте экран «О пользователе», где выводится скорость, высота и направление движения.