

ДИСЦИПЛИНА Разработка приложений на языке Котлин  
(полное наименование дисциплины без сокращений)

ИНСТИТУТ информационных технологий

КАФЕДРА информационных технологий в атомной энергетике  
(полное наименование кафедры)

ВИД УЧЕБНОГО  
МАТЕРИАЛА Практическая работа  
(в соответствии с пп 1-11)

ПРЕПОДАВАТЕЛЬ Золотухин Святослав Александрович  
(фамилия, имя, отчество)

СЕМЕСТР 5 семестр 2025 – 2026 учебный год  
(указать семестр обучения, учебный год)

## **ЗАДАНИЕ ДЛЯ 15 ПРАКТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «РАЗРАБОТКА ПРИЛОЖЕНИЙ НА ЯЗЫКЕ КОТЛИН»**

### **Вариант 1. Анализ ленты социальной сети**

Создайте программу для анализа постов в социальной сети.

Данные:

```
data class Post(  
    val id: Int,  
    val author: String,  
    val content: String,  
    val likes: Int,  
    val shares: Int,  
    val hashtags: List<String>  
)
```

Необходимо релизовать:

1. Создание последовательности из 10–15 постов с разными авторами, лайками и хештегами.
2. Фильтрацию по популярности – выбрать посты с количеством лайков > 100.
3. Поиск по хештегам – найти все посты, содержащие определенный хештег.
4. Трансформацию в статистику – преобразовать каждый пост в строку формата: «Автор: [author], Лайков: [likes], Репостов: [shares]».
5. Группировку постов по авторам.
6. Для каждого автора посчитать среднее количество лайков (*average*) и общее количество репостов (*sum*).
7. Вывести топ-3 авторов по суммарным лайкам.

### **Вариант 2. Игровые достижения и статистика**

Создайте программу для анализа данных игроков в видеоигре.

Данные:

```
data class Player(  
    val nickname: String,  
    val level: Int,  
    val score: Int,  
    val achievements: List<String>,  
    val playTimeHours: Int  
)
```

Необходимо релизовать:

1. Создание последовательности из 10–15 игроков.
2. Фильтрацию по уровню – выбрать игроков уровня  $> 10$ .
3. Проверку наличия достижения – проверить, есть ли хотя бы один игрок с достижением «Легенда».
4. Трансформацию в статистику - преобразовать каждого игрока в строку: «Игрок: [nickname], Уровень: [level], Очки: [score]».
5. Группировку игроков по диапазонам времени: менее 50 часов, от 50 до 200 часов, более 200 часов.
6. Сортировку игроков по убыванию очков.
7. Расчет среднего уровня игроков в каждой группе времени.

### Вариант 3. Экологические данные

Создайте программу для анализа данных о загрязнении воздуха в городах.

Данные:

```
data class CityAirQuality(  
    val city: String,  
    val country: String,  
    val pm25: Int, // концентрация PM2.5  
    val pm10: Int,  
    val lastUpdate: String  
)
```

Необходимо релизовать:

1. Создание последовательности из 10–15 городов с разными показателями качества воздуха.
2. Фильтрацию по уровню – выбрать города, где  $PM_{2.5} > 50$  (опасный уровень).
3. Проверку безопасности – проверить, все ли города в определенной стране имеют  $PM_{10} < 100$ .
4. Трансформацию в статистику – преобразовать данные в строку: «Город: [city],  $PM_{2.5}$ : [pm25], Статус:  $\text{\$}\{if(pm25 > 50) \text{ "Опасно" else "Нормально"}\}$ »
5. Группировку города по странам.
6. Сортировку города по убыванию  $PM_{2.5}$ .
7. Расчет для каждой страны средний  $PM_{2.5}$  и  $PM_{10}$ .

### **Требования к отчету:**

Титульный лист, оглавление, текст задачи, выполненные задания (краткое описание кода реализации каждого задания с указанием листинга кода и скриншотов работы каждой программы), вывод (что было сделано в ходе выполнения работы), список использованных источников.

Оформление работ обязательно должно отвечать требованиям СМК О МИРЭА.

При защите работы необходимо ответить на несколько контрольных вопросов.

### **Литература:**

- 1) Лекционный материал
- 2) <https://kotlinlang.org/docs/home.html>