

Комплекс оценочных материалов

Специальность: 09.02.01 «Компьютерные системы и комплексы»

Дисциплина: МДК 02.04 Разработка прикладных приложений

Задания закрытого типа на выбор правильного ответа

1. В Arduino IDE объявлен массив: `int pins[] = {A0, A1, A2};`

Какой фрагмент кода корректно определяет количество элементов в этом массиве для дальнейшего перебора в цикле?

- А) `int n = pins.length;`
- Б) `int n = sizeof(pins) / sizeof(pins[0]);`
- В) `int n = pins.size();`
- Г) `int n = length(pins);`

Правильный ответ: Б

Компетенции (индикаторы): ПК 2.1, ОК 02.

2. В Java массив строк объявлен как: `String[] logs = {"25°C", "60% RH", "ON"};`

Как получить значение последнего элемента массива?

- А) `logs[logs.length - 1];`
- Б) `logs.last();`
- В) `logs.get(logs.size() - 1);`
- Г) `logs[-1];`

Правильный ответ: А

Компетенции (индикаторы): ПК 2.1, ОК 01.

3. Какой из приведённых фрагментов кода на Java представляет корректное объявление класса с приватным полем, конструктором и геттером?

- А) `class Sensor { String type; }`
- Б) `public class Sensor { private String type; public Sensor(String t) { type = t; } public String getType() { return type; } }`
- В) `Sensor(String type) { this.type = type; }`
- Г) `class Sensor { public String type; }`

Правильный ответ: Б

Компетенции (индикаторы): ПК 2.1, ОК 09.

4. В Java имеются два списка:

`List<Sensor> sensors;` // каждый Sensor имеет поля `id` и `type`

`List<Device> devices;` // каждый Device имеет поля `sensorId` и `status`

Какой фрагмент кода выведет пары «тип сенсора — статус устройства» для всех устройств, связанных с сенсорами по совпадению `sensor.id == device.sensorId`?

- А) `System.out.println(sensors.type + " " + devices.status);`

Б) for (Device d : devices) { Sensor s = findSensorById(d.sensorId);
System.out.println(s.type + " " + d.status); }

Б) for (Sensor s : sensors) { for (Device d : devices) { if (s.id == d.sensorId)
System.out.println(s.type + " " + d.status); } }

Г) JOIN sensors, devices ON sensors.id = devices.sensorId

Правильный ответ: В

Компетенции (индикаторы): ПК 2.3, ПК 2.4.

5. При модификации класса Device в Java необходимо добавить поле batteryLevel с соблюдением принципа инкапсуляции. Какой вариант корректен?

А) public double batteryLevel;

Б) private double batteryLevel; public double getBatteryLevel() { return
batteryLevel; }

В) batteryLevel: double;

Г) field batteryLevel = 0.0;

Правильный ответ: Б

Компетенции (индикаторы): ПК 2.5, ОК 01.

6. Требуется отсортировать список объектов Device сначала по убыванию поля status (лексикографически), а при равенстве — по возрастанию поля name. Какой код реализует это?

А) Collections.sort(devices,
Comparator.comparing(Device::getStatus).thenComparing(Device::getName));

Б) Collections.sort(devices,
Comparator.comparing(Device::getStatus).reversed().thenComparing(Device::get
Name));

В) Collections.sort(devices,
Comparator.comparing(Device::getName).thenComparing(Device::getStatus).reve
rsed());

Г) devices.sort((a, b) -> b.status.compareTo(a.status));

Правильный ответ: Б

Компетенции (индикаторы): ПК 2.1, ОК 04.

7. В классе Sensor определены методы:

public void setStatus(String status) { this.status = status; }

public String getStatus() { return status; }

Какой фрагмент корректно устанавливает статус в "ACTIVE" и выводит подтверждение в консоль?

А) sensor.status = "ACTIVE"; System.out.println("OK");

Б) sensor.set("status", "ACTIVE");

В) UPDATE sensor SET status = "ACTIVE";

Г) `sensor.setStatus("ACTIVE"); System.out.println("Status: " + sensor.getStatus());`

Правильный ответ: Г

Компетенции (индикаторы): ПК 2.1, ОК 05.

8. Даны два списка:

`List<Picture> pictures; // поля: name, painterId`

`List<Painter> painters; // поля: id, fullName`

Какой фрагмент выведет все картины художника с `fullName = "Picasso"`?

А) `pictures.stream().filter(p -> p.painterId == 1).forEach(System.out::println);`

Б) `pictures.stream().filter(p -> painters.stream().anyMatch(a -> a.id == p.painterId && a.fullName.equals("Picasso"))).forEach(System.out::println);`

В) `SELECT * FROM pictures WHERE painter = "Picasso";`

Г) `pictures.find("Picasso");`

Правильный ответ: Б

Компетенции (индикаторы): ПК 2.3, ОК 02.

Задания закрытого типа на установление соответствия

1. Установите соответствие между этапом разработки и уровнем проектирования:

Характеристика	Уровень
А) Настройка режимов пинов в Arduino (INPUT/OUTPUT)	1) Концептуальный
Б) Формулирование требований к функциональности системы	2) Физический
В) Определение классов Sensor, Device с методами	3) Архитектурный
Г) Разработка схемы взаимодействия модулей управления и отображения	4) Логический

Правильный ответ: А–2, Б–1, В–4, Г–3

Компетенции (индикаторы): ПК 2.1, ОК 01.

2. Соответствие между подходом к разработке и его характеристикой:

Описание	Модель
А) Обработка данных через последовательное чтение байтов из <code>InputStream</code>	1) Событийно-ориентированный
Б) Использование <code>extends</code> для расширения	2) Процедурный

Описание	Модель
функционала базового класса	
В) Реакция на нажатие кнопки через ActionListener в Swing	3) Объектно-ориентированный
Г) Последовательный опрос датчиков в цикле loop()	4) Поточковый

Правильный ответ: А–4, Б–3, В–1, Г–2

Компетенции (индикаторы): ПК 2.3, ОК 09.

3. Соответствие между типом идентификатора и его назначением:

Описание	Название
А) Поле name используется только для сортировки и поиска	1) Составной ключ
Б) Уникальный id у каждого объекта Device	2) Вспомогательный атрибут
В) Комбинация deviceId + timestamp гарантирует уникальность записи в лог	3) Первичный ключ
Г) Поле deviceId в классе Sensor ссылается на объект Device	4) Внешний ключ

Правильный ответ: А–2, Б–3, В–1, Г–4

Компетенции (индикаторы): ПК 2.4, ОК 02.

4. Соответствие между фрагментом кода и его назначением:

Описание	Название
А) double avg = sum / count;	1) Сортировка данных
Б) if (value > threshold) activate();	2) Вычисление агрегатной функции
В) Arrays.sort(devices, Comparator.comparing(...));	3) Условный оператор
Г) for (int i = 0; i < sensors.length; i++)	4) Цикл с параметром

Правильный ответ: А–2, Б–3, В–1, Г–4

Компетенции (индикаторы): ПК 2.1, ОК 01.

5. Операции над коллекциями в Java:

Описание	Название
А) Удаление из списка devices всех ID, отсутствующих в validIds	1) Пересечение
Б) Объединение двух списков показаний сенсоров в	2) Разность

Описание	Название
один	
В) Поиск сенсоров, присутствующих в обоих списках	3) Объединение
Г) Генерация всех возможных пар «устройство—сенсор»	4) Декартово произведение

Правильный ответ: А–2, Б–3, В–1, Г–4

Компетенции (индикаторы): ПК 2.3, ОК 04.

6. Работа с файлами в Java:

Результат	Оператор
А) При ошибке чтения данные не сохраняются — буфер сбрасывается	1) flush() перед close()
Б) Файл закрывается в любом случае, даже при возникновении исключения	2) Неиспользование flush()
В) Данные записываются в файл немедленно по таймеру	3) try-finally с close()
Г) Все данные из буфера гарантированно попадают на диск перед завершением	4) Планировщик задач с автосохранением

Правильный ответ: А–2, Б–3, В–4, Г–1

Компетенции (индикаторы): ПК 2.5, ОК 07.

7. Обработка сигналов в Arduino:

Описание	Название
А) Передача данных по Serial в строгом порядке, байт за байтом	1) Повторяемый
Б) Повторное чтение сигнала до достижения стабильного значения	2) Последовательный
В) Доступ к разделяемой переменной защищён блокировкой	3) Асинхронный
Г) Чтение датчика запускается по прерыванию, без ожидания в loop()	4) Синхронизированный

Правильный ответ: А–2, Б–1, В–4, Г–3

Компетенции (индикаторы): ПК 2.4, ОК 06.

8. UML: связь между классами:

Фрагмент	Описание
А) JFrame (1) — (1) JPanel	1) Обязательная с обеих сторон
Б) ArduinoBoard (0..1) — (1..*) Pin	2) Композиция (часть-целое)

Фрагмент	Описание
В) Device (1) — (1..*) Sensor	3) Необязательная с одной стороны
Г) Class — (1..*) Method	4) Агрегация без обязательности

Правильный ответ: А–1, Б–3, В–1, Г–2

Компетенции (индикаторы): ПК 2.1, ОК 05.

Задания закрытого типа на установление правильной последовательности

1. Расположите фундаментальные модели программирования в порядке их исторического появления, начиная с процедурной модели в С для базовых IoT-скриптов и заканчивая объектно-ориентированной в Java для современных приложений.

- А) Объектно-ориентированная (Java для классов IoT)
- Б) Иерархическая (структуры данных в C++ для пинов Arduino)
- В) Сетевая (соединения устройств в IoT-протоколах)
- Г) Процедурная (основные функции в С для Arduino IDE)

Правильный ответ: Г, Б, В, А

Компетенции (индикаторы): ОК 02, ОК 03.

2. Расположите действия в порядке, в котором они фактически выполняются при работе цикла `for (int i = 0; i < 5; i++) { ... }`:

- А) Проверка условия `i < 5`
- Б) Выполнение тела цикла
- В) Инициализация счётчика `int i = 0`
- Г) Увеличение счётчика `i++`

Правильный ответ: В, А, Б, Г

Компетенции (индикаторы): ПК 2.4, ОК 01.

3. Массив: `["ZigbeeModule", "Actuator", "DHT22", "ESP32"]`

После сортировки по убыванию (DESC) получается определённый порядок. Укажите, в какой последовательности элементы будут выведены:

- А) ZigbeeModule
- Б) ESP32
- В) DHT22
- Г) Actuator

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ПК 2.1, ОК 04.

4. Расположите части объявления метода в классе Java в синтаксически правильном порядке:

- А) `{ return analogRead(pin); }`
- Б) `public`

- В) (int pin)
- Г) int
- Д) readValue

Правильный ответ: Б, Г, Д, В, А

Компетенции (индикаторы): ПК 2.3, ОК 09.

5. Расположите виды обработки данных в Arduino в порядке возрастания детерминированности выполнения (от наименее предсказуемой к наиболее строгой):

- А) Чтение по прерыванию (например, attachInterrupt)
- Б) Последовательная передача по UART (Serial.write)
- В) Повторяемое опросное чтение (while (!ready) read();)
- Г) Обычный опрос в loop() без синхронизации

Правильный ответ: А, Г, В, Б

Компетенции (индикаторы): ПК 2.5, ОК 07.

6. Укажите правильный порядок структурных элементов в файле Main.java:

- А) public class Main {
- Б) import java.util.List;
- В) private List<Sensor> sensors;
- Г) public static void main(String[] args) { ... }
- Д) } (закрывающая скобка класса)

Правильный ответ: Б, А, В, Г, Д

Компетенции (индикаторы): ПК 2.4, ОК 02.

7. Укажите последовательность шагов инициализации пина для чтения аналогового сенсора в Arduino на C++: включение библиотеки, объявление пина в переменной, установка режима в setup(), чтение значения в loop().

А) void loop() { int value = analogRead(sensorPin); Serial.println(value); } (чтение и вывод в основном цикле)

Б) const int sensorPin = A0; (объявление константы пина в начале скетча)

В) #include <Arduino.h> (включение стандартной библиотеки Arduino в начале файла)

Г) pinMode(sensorPin, INPUT); (установка режима ввода в функции setup())

Правильный ответ: В, Б, Г, А

Компетенции (индикаторы): ПК 2.1, ОК 01.

8. Расположите элементы объектно-ориентированной структуры в порядке их появления при запуске программы:

- А) Создание объекта: Sensor temp = new Sensor("temp", A0);

- Б) Определение класса: `public class Sensor { ... }`
- В) Объявление пакета: `package iot.devices;`
- Г) Импорт зависимостей: `import java.io.Serializable;`

Правильный ответ: В, Г, Б, А

Компетенции (индикаторы): ПК 2.3, ОК 03.

Задания закрытого типа на установление правильной последовательности

1. Установите правильную последовательность этапов компиляции и выполнения простого Java-приложения для IoT-устройства (от написания кода до запуска).

Варианты шагов (в перемешанном порядке):

- А) Запуск JVM: `java Main`
- Б) Компиляция: `javac Main.java` (получение `Main.class`)
- В) Написание кода в файле `Main.java` с методом `main()`
- Г) Загрузка байт-кода и выполнение метода `main()`

Правильная последовательность: В → Б → А → Г

Компетенции (индикаторы): ПК 2.1, ОК 01.

2. Установите правильную последовательность действий при создании и использовании объекта класса `Sensor` в Java.

Варианты шагов:

- А) Вызов метода объекта: `sensor.readValue()`
- Б) Импорт класса: `import iot.Sensor;` (если в пакете)
- В) Создание экземпляра: `Sensor sensor = new Sensor("temp");`
- Г) Объявление класса: `public class Sensor { ... }`

Правильная последовательность: Г → Б → В → А

Компетенции (индикаторы): ПК 2.3, ОК 09.

3. Установите правильную последовательность выполнения кода в Arduino-скетче от момента загрузки на плату.

Варианты шагов:

- А) Многократное выполнение `void loop() { ... }`
- Б) Однократное выполнение `void setup() { ... }`
- В) Инициализация аппаратных библиотек и переменных (до `setup`)
- Г) Завершение загрузки скетча и старт выполнения

Правильная последовательность: Г → В → Б → А

Компетенции (индикаторы): ПК 2.4, ОК 02.

4. Установите правильную последовательность операций Stream API в Java для фильтрации, сортировки и вывода списка устройств с `batteryLevel > 70`.

Варианты шагов:

- А) `.forEach(System.out::println)`
- Б) `.sorted(Comparator.comparing(Device::getName))`
- В) `devices.stream()`
- Г) `.filter(d -> d.getBatteryLevel() > 70)`

Правильная последовательность: В → Г → Б → А

Компетенции (индикаторы): ПК 2.4, ОК 04.

5. Установите правильную последовательность добавления элементов в `ArrayList<String> readings` и последующего получения размера.

Варианты шагов:

- А) `int size = readings.size();`
- Б) `readings.add("25°C");`
- В) `List<String> readings = new ArrayList<>();`
- Г) `readings.add("60%");`

Правильная последовательность: В → Б → Г → А

Компетенции (индикаторы): ПК 2.1, ОК 01.

6. Установите правильную последовательность шагов для обработки прерывания в Arduino (на примере `attachInterrupt`).

Варианты шагов:

- А) Выполнение функции обработчика ISR при возникновении события
- Б) Определение функции-обработчика `void handleInterrupt() { ... }`
- В) Регистрация прерывания: `attachInterrupt(digitalPinToInterrupt(pin), handleInterrupt, RISING);`

Г) Настройка пина: `pinMode(pin, INPUT_PULLUP);` в `setup()`

Правильная последовательность: Б → Г → В → А

Компетенции (индикаторы): ПК 2.5, ОК 07.

7. Установите правильную последовательность наследования и создания объектов в Java (базовый класс `Device`, дочерний `SmartDevice`).

Варианты шагов:

А) Создание объекта дочернего класса: `SmartDevice sd = new SmartDevice();`

Б) Объявление дочернего класса: `public class SmartDevice extends Device { ... }`

В) Объявление базового класса: `public class Device { ... }`

Г) Вызов переопределённого метода `sd.readAdvancedData();`

Правильная последовательность: В → Б → А → Г

Компетенции (индикаторы): ПК 2.3, ОК 05.

8. Установите правильную последовательность этапов чтения данных с аналогового сенсора в Arduino и отправки по `Serial`.

Варианты шагов:

А) `Serial.println(value);`

Б) `delay(1000);` (задержка для читаемости)

В) `int value = analogRead(A0);`

Г) `Serial.begin(9600);` в `setup()`

Правильная последовательность: Г → В → А → Б

Компетенции (индикаторы): ПК 2.1, ОК 06.

Задания открытого типа на дополнение

Напишите пропущенное слово (словосочетание), раскрывающее суть термина в контексте разработки.

1. В объектно-ориентированном программировании на Java количество атрибутов (полей) в классе, описывающем сенсор (например, `id`, `type`, `value`), называется _____ класса.

Правильный ответ: арностью

Компетенции (индикаторы): ОК 02.

2. В массиве объектов `Device`, хранящем 5 экземпляров IoT-устройств, количество этих объектов называется _____ массива.

Правильный ответ: мощностью

Компетенции (индикаторы): ПК 2.4.

3. В принципах SOLID для проектирования классов Java первый принцип (Single Responsibility Principle) требует, чтобы класс отвечал только за одну _____, например, класс `Sensor` только за чтение данных, а не за GUI. Укажите аббревиатуру принципа.

Правильный ответ: SRP

Компетенции (индикаторы): ПК 2.1.

4. В цикле `loop()` Arduino логически неделимая последовательность команд для чтения сенсора, обработки и вывода в `Serial`, переводящая систему из состояния "ожидание" в "обработано", называется _____.

Правильный ответ: итерацией цикла

Компетенции (индикаторы): ПК 2.5.

5. В интерфейсе `List<String> readings` в Java метод, возвращающий количество элементов показаний сенсоров для проверки полноты данных, называется _____.

Правильный ответ: `size`

Компетенции (индикаторы): ПК 2.3.

6. Если класс `SimpleDevice` имеет поля `id` и `name`, то его _____ равна 2, что определяет количество параметров в конструкторе.

Правильный ответ: арность

Компетенции (индикаторы): ОК 01.

7. При слиянии двух массивов в Java — одного с 8 показаниями температуры и второго с 7 показаниями влажности — общее количество элементов в результирующем массиве для анализа будет ____.

Правильный ответ: 15

Компетенции (индикаторы): ПК 2.4.

8. Метод из пакета `java.util.Objects`, который выбрасывает исключение, если значение поля `battery` равно `null`, предотвращая ошибки в IoT-коде, называется ____.

Правильный ответ: `requireNonNull`

Компетенции (индикаторы): ОК 09.

Задания открытого типа с кратким свободным ответом

1. В объектно-ориентированном программировании принцип, обеспечивающий сокрытие внутренних данных класса `Sensor` (например, `private double value`) от внешнего доступа с помощью геттеров/сеттеров, называется ____.

Правильный ответ: инкапсуляция

Компетенции (индикаторы): ПК 2.1 (проектирование с защитой данных).

2. Принцип ООП, позволяющий классу `AdvancedDevice` наследовать методы `readData()` от базового `Device` и добавлять новые, расширяя функциональность для сложных IoT-систем, называется ____.

Правильный ответ: наследование

Компетенции (индикаторы): ПК 2.3.

3. В коде Arduino для формирования строки сообщения "Температура: 25°C" из переменных, оператор соединения строк, аналогичный `+` в Java, но с использованием `String`, — это ____.

Правильный ответ: `+` (конкатенация)

Компетенции (индикаторы): ПК 2.1.

4. В файле `.java` для создания базового класса, описывающего устройство с конструктором, стандартная конструкция объявления начинается с ____.

Правильный ответ: `public class`

Компетенции (индикаторы): ПК 2.5.

5. В ООП принцип, позволяющий методам разных классов (`Sensor` и `Actuator`) обрабатывать один интерфейс `readInput()` по-разному, обеспечивая гибкость в IoT, называется ____.

Правильный ответ: полиморфизм

Компетенции (индикаторы): ОК 07.

6. Оператор Java, позволяющий классу ChildDevice расширять базовый ParentDevice, добавляя новые поля без изменения родителя, — это ____.

Правильный ответ: extends

Компетенции (индикаторы): ПК 2.5.

7. Конструкция языка Java для проверки условия (например, if (battery > 50) activateDevice();) и выбора пути выполнения в ветвящемся алгоритме обработки сигнала, называется ____.

Правильный ответ: if-else

Компетенции (индикаторы): ПК 2.4.

8. Функция в Arduino IDE, устанавливающая режим пина (INPUT или OUTPUT) для подключения сенсора перед чтением данных в setup(), называется ____.

Правильный ответ: pinMode

Компетенции (индикаторы): ОК 06.

Задания открытого типа с развернутым ответом

1. Выполните рефакторинг универсального класса UniversalIoT (с полями sensor_id, type, value, device_id, location, battery_level) для соответствия принципам ООП в Java, где один класс не должен отвечать за все аспекты (нарушение SRP).

Задачи:

- проанализируйте предметную область IoT (сенсоры, устройства, данные);
- выявите нарушения (смешение данных устройства и сенсора);
- предложите декомпозицию на 2–3 отдельных класса с объяснением ролей каждого.

Время выполнения – 15 мин.

Ожидаемый ответ (один из вариантов):

Класс Device (device_id, location, battery_level) — отвечает за состояние устройства (геттеры для battery, метод updateLocation()).

Класс Sensor (sensor_id, device_id, type, value) — отвечает за данные сенсора (метод readValue(), ссылка на Device).

Это устраняет SRP: Device — аппаратная часть, Sensor — данные. Атрибуты распределены полностью.

Критерии оценивания:

- наличие 2+ классов с ролями;
- соответствие SRP и инкапсуляции;
- полное распределение атрибутов.

Компетенции (индикаторы): ПК 2.1, ОК 01.

2. Составьте фрагмент кода на Java для обработки списка сенсоров: выведите типы сенсоров и среднее battery_level только для тех, где battery_level > 50%, группируя по типу (используйте Map или цикл).

Задачи:

- проанализируйте структуру (класс Sensor: String type, double battery_level);
- выберите средства (цикл for, HashMap для группировки, расчет avg);
- напишите код с выводом в консоль (например, "Temp: count=2, avg=70.5").

Время выполнения – 10 мин.

Ожидаемый результат (вариант):

```
Map<String, List<Double>> groups = new HashMap<>();
for (Sensor s : sensors) {
    if (s.battery_level > 50) {
        groups.computeIfAbsent(s.type, k -> new
ArrayList<>()).add(s.battery_level);
    }
}
for (Map.Entry<String, List<Double>> e : groups.entrySet()) {
    double avg =
e.getValue().stream().mapToDouble(Double::doubleValue).average().orElse(0);
    System.out.println(e.getKey() + ": count=" + e.getValue().size() + ",
avg=" + avg);
}
```

Критерии оценивания:

- правильный вывод с count и avg;
- фильтр >50;
- группировка и расчет; использование полей.

Компетенции (индикаторы): ПК 2.4, ОК 02.

3. Дана UML-диаграмма классов для IoT-приложения (Device с полем location, Sensor с type и value, связь 1-M). Дайте развернутое описание особенностей предметной области.

Задачи:

- проанализируйте область (управление устройствами через сенсоры);
- выявите сущности (Device, Sensor) и классифицируйте атрибуты (статические/динамические);

– опишите связи (название, класс принадлежности, тип: 1-М обязательная).

Время выполнения – 30 мин.

Ожидаемый результат (вариант):

Сущности: Device (атрибуты: id — статический, location — статический, battery — динамический); Sensor (type — статический, value — динамический).

Связь "Подключен": Device (1, обязательный) — Sensor (М, обязательный) — тип "один-ко-многим", обеспечивает иерархию в IoT.

Критерии оценивания:

- все сущности с атрибутами и классификацией;
- связи с названием, классом, типом;
- все сущности в связях.

Компетенции (индикаторы): ПК 2.2, ОК 04, ОК 05.

4. Составьте Java-код для анализа логов: из списка devices и logs (Log: device_id, sensor_id, value) выведите ID и name устройств с >100 записями логов только для сенсоров, как у "MainDevice", отсортировав по name.

Задачи:

- проанализируйте классы (Device: id, name; Log: device_id, sensor_id);
- выберите средства (Stream.filter, count, Comparator);
- напишите код с выводом (id + " " + name).

Время выполнения – 15 мин.

Ожидаемый результат (вариант):

```
Device      main      =      devices.stream().filter(d      ->
d.name.equals("MainDevice")).findFirst().orElse(null);
    if (main != null) {
        devices.stream().filter(d -> logs.stream().filter(l -> l.device_id == d.id
&& logs.stream().anyMatch(l2 -> l2.sensor_id == l.sensor_id && l2.device_id ==
main.id)).count() > 100)
            .sorted(Comparator.comparing(Device::getName)).forEach(d      ->
System.out.println(d.id + " " + d.name));
    }
```

Критерии оценивания:

- вывод id/name;
- фильтр >100 и совпадение сенсоров;
- сортировка; циклы/Stream.

Компетенции (индикаторы): ПК 2.4, ОК 02.

5. Перепишите код для поиска max value по pin в классе Sensor (поля pin, value) с использованием Stream API, без вложенных циклов, выводя "pin max_value" для каждого.

Задачи:

- проанализируйте класс (Sensor: int pin, double value);
- выберите Stream (groupingBy, maxBy);
- напишите фрагмент с выводом, гарантируя уникальность max.

Время выполнения – 20 мин.

Ожидаемый результат (вариант):

```
sensors.stream().collect(Collectors.groupingBy(Sensor::getPin)).values().stream()
    .map(list                                     ->
list.stream().max(Comparator.comparing(Sensor::getValue)).orElse(null))
    .filter(Objects::nonNull).forEach(s -> System.out.println(s.pin + " " +
s.value));
```

Компетенции (индикаторы): ПК 2.1, ПК 2.4, ОК 01.

6. Составьте Java-код для вывода названий улиц и городов из списков Street (name, cityId) и City (id, name), соединив по cityId и отсортировав по имени улицы.

Задачи:

- проанализируйте классы (Street: String name, int cityId; City: int id, String name);
- выберите средства (Map для City, Stream.map и sorted);
- напишите код с выводом "street_name city_name".

Время выполнения – 10 мин.

Ожидаемый результат (вариант):

```
Map<Integer,          String>          cityNames          =
cities.stream().collect(Collectors.toMap(City::getId, City::getName));
streets.stream().map(s          ->          s.name          +          "          "          +
cityNames.get(s.cityId)).sorted().forEach(System.out::println);
```

Критерии оценивания:

- вывод пар;
- соединение по id;
- сортировка.

Компетенции (индикаторы): ПК 2.3, ОК 02.

7. Составьте Java-код для подсчета количества улиц по типу (Street: String typstr) с выводом "typstr count" , отсортировав по убыванию count.

Задачи:

- проанализируйте класс (Street: String typstr);

- выберите HashMap или groupingBy, counting;
- напишите код с sorted reversed.

Время выполнения – 10 мин.

Ожидаемый результат (вариант):

```
Map<String, Long> counts =
streets.stream().collect(Collectors.groupingBy(Street::getTypstr,
Collectors.counting()));
counts.entrySet().stream().sorted(Map.Entry.<String,
Long>comparingByValue().reversed()).forEach(e -> System.out.println(e.getKey()
+ " " + e.getValue()));
```

Критерии оценивания:

- вывод typstr count;
- группировка counting;
- сортировка desc.

Компетенции (индикаторы): ПК 2.4, ОК 04.

8. Составьте Java-код для нумерации картин по художнику: из Picture (name, painterId) и Painter (id, fio) выведите "name fio number" , нумеруя по алфавиту name для каждого fio.

Задачи:

- проанализируйте классы (Picture: String name, int painterId; Painter: int id, String fio);
- выберите группировку, sorted, индексацию;
- напишите код с выводом.

Время выполнения – 10 мин.

Ожидаемый результат (вариант):

```
Map<Integer, String> painterNames =
painters.stream().collect(Collectors.toMap(Painter::getId, Painter::getFio));
Map<String, List<Picture>> grouped =
pictures.stream().collect(Collectors.groupingBy(p
painterNames.get(p.painterId)));
grouped.forEach((fio, pics) -> IntStream.range(0, pics.size()).mapToObj(i
-> pics.get(i).name + " " + fio + " " + (i+1)).forEach(System.out::println)); // после
sorted pics
```

Критерии оценивания:

- вывод name fio number;
- группировка по fio;
- нумерация после sort.

Компетенции (индикаторы): ПК 2.1, ОК 05.

9. Составьте Java-код для вывода адресов: из Abon (shifrcit, shifstr, nomerdoma), Street (id, name), City (id, name) выведите "city_name street_name house_number", соединив по id.

Задачи:

- проанализируйте классы (Abon: int shifrcit, shifstr, String nomerdoma; etc.);

- выберите Map для справочников, Stream.map;

- напишите код с выводом.

Время выполнения – 10 мин.

Ожидаемый результат (вариант):

```
Map<Integer, String> cityMap =
cities.stream().collect(Collectors.toMap(City::getId, City::getName));
Map<Integer, String> streetMap =
streets.stream().collect(Collectors.toMap(Street::getId, Street::getName));
abons.stream().map(a -> cityMap.get(a.shifrcit) + " " +
streetMap.get(a.shifstr) + " " + a.nomerdoma).forEach(System.out::println);
```

Критерии оценивания:

- вывод трех полей;

- соединение по id;

- использование Map/Stream.

Компетенции (индикаторы): ПК 2.3, ОК 02.