

Java Core

Критерии соответствия знаний выпускников итоговой оценке 8-10:

1. Соблюдение Java конвенции от Oracle/Google.
2. Умение разбивать логику решения задачи на отдельные интерфейсы/классы (подготовка к JD2 и использование Spring подхода в разработке через интерфейсы, закладывание понятий services/repositories).
3. Умение разбивать процесс реализации задачи в рамках класса на отдельные методы, используя модификаторы доступа с упором на инкапсуляцию.
4. Умение создавать сущности с гетерами/сеттерами, equals/hashcode, toString() для хранения состояния. Не использовать rich entity.
5. Не использовать статические методы, за исключением main/вспомогательных util классов. Из метода main идет вызов объектов бизнес логики.
6. Не хранить состояния на уровне классов бизнес логики (рассматривать классы, как будто идет разработка в многопоточной среде).
7. Использование принципов SOLID и паттерна Singleton.
8. Студент должен понимать разницу между классом сущностью и классом бизнес логики.

Пример итоговой контрольной работы:

Сгенерировать три произвольных целых числа X, Y и Z. Если X больше Z, то вывести на экран сумму X и Y, иначе вывести на экран значение Z. Вывести на экран среднее арифметическое этих чисел.

Сгенерировать массив случайных целых чисел, величина которых, по модулю, не превышает 300 (-300 ... 300). Число элементов массива равно 10. Найти максимальное и минимальное числа в массиве и вывести на экран. Заменить максимальный элемент на минимальный, умноженный на максимальный.

Сгенерировать 10 объектов класса Car у которых характеристики name(String), carModel (Enum с тремя объектами-значениями), engineCapacity(int в диапазоне 1-3).

Сгруппировать их по engineCapacity используя Map<Integer, List<Car>>

Для произвольного значения engineCapacity записать в файл соответствующий список объектов в формате:

<name> : <carMode> : engineCapacity

Создать класс потока(PlayerGeneratorTask), который генерирует коллекцию из 10 объектов класса Player с полями name(String), age(Integer), isActive(boolean) по условию: name – любое, age – от 20 до 40, isActive – true/false и записывает ее в файл. Поток возвращает имя файла, в который он писал.

Создать класс потока(PlayerReaderTask), который читает файл и возвращает коллекцию активных Player в возрастном диапазоне от 25 до 30.

Создать пул потоков из 4 штук

Запустить 20 потоков класса PlayerGeneratorTask.

Запустить поток класса PlayerReaderTask на каждый выполненный PlayerGeneratorTask.

Отсортировать по возрасту и вывести на экран имена 5 игроков в UpperCase, удовлетворяющие условию с помощью Stream API.

