

Потоки ввода/вывода, сериализация

(*) — для программистов и любителей предмета

1. Записать массив целых чисел в двоичный поток. Прочитать массив целых чисел из двоичного потока. Предполагается, что до чтения массив уже создан, нужно прочитать n чисел, где n — длина массива.
2. Аналогично, используя символьные потоки. В потоке числа должны разделяться пробелами.
3. Используя класс `RandomAccessFile`, прочитайте массив целых чисел, начиная с заданной позиции.
4. Используя класс `File`, получите список всех файлов с заданным расширением в заданном каталоге (поиск в подкаталогах выполнять не надо).
5. *Используя класс `File`, получите в заданном каталоге список всех файлов и подкаталогов, имена которых удовлетворяют заданному регулярному выражению. Поиск должен распространиться в подкаталоги. Имена найденных файлов должны быть вместе с абсолютными путями.
6. Написать классы:
 - `Person` (человек), содержит Ф.И.О. (раздельно), дату рождения,
 - `Flat` (квартира), содержит номер, площадь, данные о владельцах (список людей),
 - `House` (дом), содержит кадастровый номер дома (строка), адрес, старшего по дому (человек), список квартир.

В этих классах предусмотреть нужные конструкторы, геттеры, `equals` и `hashCode`.

Напишите сервисный класс с методами, которые сериализуют и десериализуют объект типа `House` в заданный поток средствами Java.

7. *Напишите метод сохранения объекта типа `House` в csv-файл ([о формате csv](#)). Используйте символ `;` как разделитель. Имя файла должно строиться по шаблону
`house_кадастровый_номер_дома.csv`.

Файл должен соответствовать таблице следующего вида (информация о форматировании не сохраняется):

Данные о доме

Кадастровый номер:	12345
Адрес:	г. Омск, пр. Мира, 321
Старший по дому:	Иванов Сидор Петрович

Данные о квартирах

№	Площадь, кв. м	Владельцы
1	40	Петров А.В.
2	65	Сидорова О.М.
3	58	Васечкин И.П., Золотов С.Б.

Можно создать пример такой таблицы с помощью любого редактора электронных таблиц, сохранить в формате csv, и вы получите пример требуемого файла.

8. Подключите к проекту библиотеку Jackson. Напишите методы сериализации / десериализации объектов типа House в строки. Используйте data binding.
9. *Сравнить две json-строки на равенство. Нужно учесть, что json-объект — неупорядоченный набор полей, сравнение должно распространяться вглубь объектов и массивов. (Подсказка: средствами Jackson можно разобрать json в дерево).
10. *Используя Jackson Streaming API, напишите классы — сериализаторы / десериализаторы для объектов типа House, Flat и Person. Сериализатор человека должен ФИО записывать в виде единого поля "fullName". Зарегистрируйте написанные (де)сериализаторы для указанных классов так, чтобы они применялись вместо (де)сериализаторов по умолчанию.
11. *Изучите основы работы с библиотекой mockito (см. site.mockito.org, полезно посмотреть также habr.com/ru/post/243155/). Подключите ее к проекту. Напишите с ее помощью unit-тесты для методов из заданий 4 и 5.