**Задача 1**

В текущий проект в Maven добавить следующие зависимости:

com.google.code.gson:gson

com.fasterxml.jackson.dataformat

версии взять последние стабильные на момент выполнения задания

В данной задаче необходимо создать классы-помощники, которые будут сериализовывать и десериализовывать объекты.

Необходимо создать пакет model, в котором создать интерфейс Generatable. Данный интерфейс будет предоставлять классам метод generate(), который будет заполнять поля объекта сгенерированными значениями. Метод generate() должен возвращать generic-тип T, который необходимо ограничить только классами, имплементирующими интерфейс Generatable.

Создать классы House, Flat, Room (дом, квартира, комната), каждый из которых должен реализовывать интерфейс Generatable. Описание полей и поведения классов в таблице ниже:

|  |  |  |  |
| --- | --- | --- | --- |
| Класс | House | Flat | Room |
| Поля | List<Flat> flats | List<Room> rooms | int width, int height |
| Метод generate() | Заполняет список квартир пятью сгенерированными квартирами | Заполняет список комнат двумя сгенерированными комнатами | Устанавливает значения width = 5, height = 3. |

Рядом с пакетом model создать пакет helpers. Данный пакет будет содержать классы, реализующую работу с сериализацией и десериализацией. Создать интерфейс SerializeHelper, содержащий 2 метода:

T deserialize(String data, Class<T> clazz);

String serialize(T object);

Интерфейс должен работать с generic-типом Т, который необходимо органичить классами, имплементирующими интерфейс Generatable.

Создать 2 класса – JsonHelper и XmlHelper, реализовать интерфейс SerializeHelper таким образом, чтобы класс JsonHelper работал с форматом Json, а XmlHelper работал с форматом XML.

К заданию приложены файлы, которые помогут протестировать написанные методы. К реализации необходимо написать следующие тесты:

* Сгенерировать дом с помощью метода new House().generate(), сериализовать его в форматы JSON и XML
* Из файлов room.\*\*\* десериализовать комнату с помощью метода deserialize() из форматов JSON и XML
* Из файлов flat.\*\*\* десериализовать квартиру с помощью этого же метода
* Из файлов house.\*\*\* десериализовать дом.

Тесты необходимо распределить по разным классам, по операции (десериализация/сериализация) и по форматам (xml/json).

|  |  |
| --- | --- |
| Имя файла | Содержимое |
| expected\_json.txt | Ожидаемое содержимое при сериализации в json сгенерированного дома. |
| expected\_xml.txt | Ожидаемое содержимое при сериализации в xml сгенерированного дома. |
| room.json | JSON, содержащий информацию о комнате. Необходимо проверить, то что он десериализуется корректно и поля заполняются корректными значениями. |
| room.xml | XML, содержащий аналогичную информацию. |
| flat.json | JSON, содержащий информацию о квартире. |
| flat.xml | XML, содержащий информацию о квартире. |
| house.json | JSON, содержащий информацию о доме. |
| house.xml | XML, содержащий информацию о доме. |

**Задача 2**

К данной задаче приложен Excel-файл data.xlsx, содержащий данные на нескольких листах. Форматы всех ячеек являются текстовыми. Данные являются произвольным набором ключей и значений. Первая строка содержит в себе ключи. Каждая последующая строка – значения произвольного объекта. Например, следующая таблица:

|  |  |  |  |
| --- | --- | --- | --- |
| last\_name | first\_name | patronymic | birth\_date |
| Иванов | Петр | Михайлович | 12.04.1990 |
| Петров | Константин | Николаевич | 25.01.1984 |
| Кузнецов | Александр | Сергеевич | 20.07.1987 |

содержит ключи «last\_name», «first\_name», «patronymic», «birth\_date» и 3 записи со значениями этих ключей.

Необходимо создать класс Helper с двумя методами:

public static String getJsonArrayFromXls(String path, String sheetName);

public static String getXmlFromXls(String path, String sheetName);

Первый метод должен возвращать строку, содержащую массив в формате json, где каждый из элементов соответствует записи в таблице в Excel-файле с ключами из первой строки и значениями из текущих последующих строк.

Второй метод должен возвращать строку, содержащую массив в формате xml аналогично.

Методы должны быть универсальными и работающими при добавлении нового листа в XLS-документ с произвольными ключами. Модельные классы для решения задачи не создавать.

Протестировать данное решение, используя данные из файлов <Название листа>.\*\*\*, приложенные к заданию, в качестве ожидаемого результата.

**Задача 3**

К заданию приложены несколько файлов, в которых хранятся json-объекты. Файлы между собой не связаны. Необходимо создать несколько пакетов, названия которых будут соответствовать именам файлам без расширения. На основе содержимого файла необходимо создать модельные классы. Главный класс также будет иметь название в соответствие с названием файла. Классы, объекты которых находятся внутри главного класса, называть необходимо по названию ключа (см пример). В классах сгенерировать методы equals() и hashcode()

Реализовать тесты, в которых необходимо десериализовать json в объект главного класса, проверить, что поля действительно десериализовались нужными значениями. Для тех полей json-объекта, у которых значение null, выбрать типом класс Object. Для проверки пользоваться возможностями метода equals()

Пример:

Имеется файл person.json, содержащий следующий json:

|  |
| --- |
| {  "firstName": "value1",  "passport": {  "seria": "5400",  "number": "123000"  },  "address": {  "street": "ул.Мясницкая",  "houseNumber": 15,  "flatNumber": 268  },  "registration": null  } |

Для такого файла название главного класса будет Person, этот класс будет иметь поля – строку firstName, Passport passport, Address address, Object registration. В пакете person будет 3 класса: Person, Passport, Address.

Рекомендуемая последовательность при выполнении: Сначала person, затем project, в конце document.