Тестовое задание

Нужно создать модуль интеграционной системы, конвертирующий документы из бинарного вида в json.

Из учётной системы приходит заказ (например, на поставку каких-то товаров) в бинарном виде. Этот документ нужно преобразовать в json и передать дальше.

Поля документа в бинарном виде кодируется структурами TLV (tag-length-value):

- tag - 2 байта, little endian, определяет тип поля;

- length - 2 байта, little endian, определяет длину значения в байтах;

- value - length байт, тип данных зависит от тега.

Используются следующие типы данных:

- uint32 - целое без знака, 4 байта, little endian;

- string - строка в кодировке CP866;

- VLN - variable length number - целое число без знака варьируемой длины, little endian;

- FVLN - floating point variable length number - число без знака с плавающей десятичной точкой; первый байт определяет положение точки (количество знаков после точки).

Допустимы вложенные TLV-структуры.

В документе допустимы следующие поля:

- дата-время заказа: tag = 1, тип uint32, содержит unix time (количество секунд с 1 января 1970 года) в таймзоне UTC;

- номер заказа: tag = 2, тип VLN, максимальная длина 8;

- имя заказчика: tag = 3, тип string, максимальная длина 1000;

- позиция заказа: tag = 4, вложенная структура TLV, может быть более одной.

Позиция заказа имеет следующий состав:

- наименование товара: tag = 11, тип string, максимальная длина 200;

- цена единицы товара: tag = 12, тип VLN, максимальная длина 6, содержит цену в копейках;

- количество товара: tag = 13, тип FVLN, максимальная длина 8;

- общая стоимость позиции: tag = 14, тип VLN, максимальная длина 6, содержит стоимость в копейках, равную произведению цены на количество.

В документе в формате json используются следующие типы данных:

- целые числа;

- числа с плавающей десятичной точкой;

- строки в кодировке utf-8.

Поля бинарного документа должны отображаться на следующие поля документа json:

- дата-время заказа (tag = 1): поле dateTime, тип строка, содержит дату-время в формате yyyy-MM-dd'T'HH:mm:ss в UTC;

- номер заказа (tag = 2): поле orderNumber, тип число;

- имя заказчика (tag = 3): поле customerName, тип строка;

- позиция заказа (tag = 4): поле items, массив объектов следующей структуры:

- наименование товара (tag = 11): поле name, тип строка;

- цена единицы товара (tag = 12): поле price, целое число без знака;

- количество товара (tag = 13): поле quantity, число с плавающий точкой;

- общая стоимость позиции (tag = 14): поле sum, целое число без знака.

Пример преобразования

Исходный документ

01 00 04 00 A8 32 92 56 02 00 03 00 04 71 02 03

00 0B 00 8E 8E 8E 20 90 AE AC A0 E8 AA A0 04 00

1D 00 0B 00 07 00 84 EB E0 AE AA AE AB 0C 00 02

00 20 4E 0D 00 02 00 00 02 0E 00 02 00 40 9C

* 01 00 04 00 A8 32 92 56
* 02 00 03 00 04 71 02
* 03 00 0B 00 8E 8E 8E 20 90 AE AC A0 E8 AA A0
* 04 00 1D 00
  + 0B 00 07 00 84 EB E0 AE AA AE AB
  + 0C 00 02 00 20 4E
  + 0D 00 02 00 00 02
  + 0E 00 02 00 40 9C

Конечный документ (форматрирование не важно)

{

"dateTime": "2016-01-10T10:30:00",

"orderNumber": 160004,

"customerName": "ООО Ромашка",

"items": [

{

"name": "Дырокол",

"price": 20000,

"quantity": 2,

"sum": 40000

}

]

}

Требования к составу решения

1 уровень

- программа на языке Java, принимающая 2 аргумента командной строки: имя файла документа-источником (бинарный формат) и имя файла документа-приемника (формат json);

- файл readme, содержащий инструкции по сборке и запуску программы из командной строки.

2 уровень

Дополнительно к файлам первого уровня:

- юнит-тесты решения.