

# Knowledge Management Coding Exercise

May 15, 2017

## Problem Statement:

Imagine that you are tasked with managing the inventory of a library. Books come in and out of the library system every day, and while they're here they can be checked in and out of the library. The library's inventory is stored and handled using Google's Protocol Buffers (protobuf), a binary serialization protocol. More information about protobuf can be found in the Resources section below.

You have been provided with a set of books currently in the library, in the directory `books/`. These are files serialized by the protobuf message `protos/Book.proto`. At the start of the day, you have one of each of the listed books in the inventory, and they are all checked in. The library can have multiple copies of books with same bookID. For simplicity, the library only works with the books in directory `books/`.

You have also been provided with a serialized list of events occurring in the first day of running the library in `eventsDayOne.pb`. Each event is encoded with the protobufs message defined in `protos/EventWrapper.proto`. This event wrapper holds a series of transactional operations that could take place on a book. The transaction message has an enum, denoting the type of transaction, and a bookID, which identifies the book that is either being checked in or out, added to, or removed from the library system.

You will need to write Java code that reads in the protobuf data using the protoc-compiled classes and completes the series of events in order. Your code should conform to SOLID design principles, and it should be readable and well documented. At the end of the day (the events provided), you will need to serialize the current state of the library's inventory using the given `InventoryReport.proto` message definition. schema is designed to hold a set of statuses, each associated with a book of id `BookId`. Each `BookStatus` contains a count of copies of that book currently in the library (available for checkout) and a count of copies currently checked out (unavailable for checkout). Save the project in a zip file to submit.

## Inputs:

- `books/`
  - A set of books currently in the library, these files are serialized by their message definition, in `protos/Book.proto`

- `protos/InventoryReport.proto`
  - The message definition used to serialize the Library's inventory
- `eventsDayOne.pb`
  - A list of events occurring in the first day of running the library serialized by the message definition in `protos/EventWrapper.proto`

## Deliverables:

- We are providing you with a Maven 3 project with a basic configuration in the `nyt-book-library` directory. Add your code to the project and make sure that we can run the Maven project from command line.
- `InventoryReport.pb`
  - The serialized binary output of the Library's inventory after all events have been run
- `nyt-book-library/src/main/*.java`
  - The Java code you write to:
    - \* read in the protobuf books & events,
    - \* process the events, and
    - \* output the inventory.
- `nyt-book-library/src/test/*.java`
  - Unit tests supporting critical areas of your code

## Resources:

- Protocol Buffers (v3): a language-neutral, platform-neutral, extensible way of serializing structured data; a flexible, efficient, automated mechanism for serializing structured data.
  - <https://developers.google.com/protocol-buffers/docs/overview>
  - <https://developers.google.com/protocol-buffers/docs/proto3>
- By running the protobuf compiler (`protoc`), Java classes can be compiled to handle the protobuf objects.
  - <https://developers.google.com/protocol-buffers/docs/javatutorial>
- Maven: a tool that can be used for building and managing any Java-based project.
  - <https://maven.apache.org/what-is-maven.html>
- SOLID design principles
  - [https://en.wikipedia.org/wiki/SOLID\\_\(object-oriented\\_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))