CS3310 – Kaminski - Fall 2014 CountriesOfTheWorld App 1.0
Asgn 2 Project Specs – PART A

[Part B discusses the BST aspects - available later]
Also see "further notes" on asgn website.

################################ OVERVIEW ################################

A2 is a modification of A1 [Make a copy of A1 and make changes to the COPY]. If you had mistakes in A1 (whether grader caught them or not) or now find there's a better way to handle things, change things for A2 before proceeding with the new parts of A2.

Assume A2 requirements are the same as A's, unless changes are specified here or in class. Two main changes:

- dataTable is now an **external** table (file) rather than an internal table (in memory)
  which still uses a **direct address** structure on id as the key
- nameIndex is still an **internal** index
  which now uses a **binary search tree** structure on name as the key
  rather than a sequential list

########################## WHAT NEEDS CHANGING ? ##########################

Programs

- TestDriver – minor changes (to be described in the A2DemoSpecs)
- PrettyPrintUtility – reads/prints CountryData.txt file besides Backup.txt file
  (and Backup.txt is now different, so. . .)
- Setup and UserApp - no changes

Instantiable Classes

- RawData and UI – no changes
- DataTable and NameIndex
  - no changes to public interface (i.e., headers for public service methods)
  - lots of changes to:
    - private data storage
    - implementations (the body) of public methods/constructor/…
    - private methods (header & body)

2 NEW Instantiable Classes now required

- DataRecord class and BSTNode class

Data Files

- A2RawData.csv and A2TransData?.txt – same format,
  but different fileNameSuffix & different data contents
- CountryData.txt - NEW FILE – it's the dataTable
- Backup.txt – contains headerRec & dump of nameIndex, but NOT the dataTable
  - headerRec contains different fields than A1
  - nameIndex records contain 2 additional fields than A1
- Log.txt -
  - 2 additional status messages: STATUS > CountryData FILE opened
    STATUS > CountryData FILE closed
  - PrettyPrintUtility's output looks different for headerRec & nameIndex and the LOC's for dataTable are now RRN's (not subscripts)

########################## DataTable & DataRecord classes ##########################

Who's responsible?

- RawData class's methods are responsible for:
  - cleaning a RawData file's LINE (remove 1st 30 char's & the last 2 char's & the single quote chars)
  - splitting(on commas) the data line into individual fields
  - discarding (i.e., not storing) all but the 7 required fields
  - perhaps converting the numeric fields into int/long/double's
  - BUT, it is NOT RESPONSIBLE for the truncating/padding of the 7 fields to the specified sizes that dataRecord needs
- DataTable class does all handling of the countryData.txt FILE
- DataRecord class's methods are responsible for dealing with individual records/fields:
  - Truncating/padding the 7 fields as required for the DataTable file

There will only be a single dataRecord object since there is never more than a single data record in memory at once. DO NOT SET UP AN ARRAY OF dataRecord objects – this is an EXTERNAL STORAGE STRUCTURE.

Setup & UserApp (& RawData & UI & NameIndex) don't know that dataTable is: a FILE or that it uses a DirectAddress structure. PrettyPrintUtility program knows it's a file. TestDriver knows it's a file since it has to delete it.

Record Description

1st) ONE header Record containing n and maxId
2nd) rest of the records (except empty locations) contain (with fields in this order)

code	– 3 char string or charArray (truncated or space-filled on right)
id	– 3 digits (0-filled on left)
name	– 15 char string or charArray (truncated or space-filled on right)
continent	– 13 char string or charArray (truncated or space-filled on right)
size	– 8 digits (0-filled on left)
population	– 10 digits (0-filled on left)
lifeExp	– 4 char (3 digits plus decimal point) (0-filled on left)

[NOTE: truncating and/or zero/space-filling is done to ensure all records are of the exact same size (i.e., fixed-length records, necessary for direct address files).

OTHER NOTES:

- DeleteById is still a dummy stub
- SelectById and Insert MUST use direct address "search" – **Linear search → lose LOTS OF POINTS**
- Empty locations and empty-location-detection will be discussed in class
  - Allow for BOTH the "all-0-bits" case and the "read-failed" case
  - Do NOT initialize the fileSpace by writing "empty records" to locations 1 through ? – this shouldn't be necessary since the OS handles this when you open a file.
- Allow for duplicate Id's (in Insert1Country) – (e.g., 2 RawData records with same id, a RawData record &later an Insert transaction, 2 Insert transactions in the same or subsequent runs of UserApp)
  - Handling:
    - The 1st country with the id is stored in the file
    - Any subsequent countries with that ID are rejected, with an error message to Log file, and the country is NOT put into the CountryData file