TEST MODE World Data App 2.5 - Java

CS 3310 – Dr. Donna Kaminski

Index

[Log.txt 2](#_Toc404651913)

[TestDriver 3](#_Toc404651914)

[UserApp 4](#_Toc404651915)

[CodeIndex 6](#_Toc404651916)

[ActualData 9](#_Toc404651917)

========================================

# PROCESSING A5TransData1.txt

SC RED -->> 50 RED wine 49172 [nodes read: 3]

SC ZIP -->> 62 ZIP code 49184 [nodes read: 3]

SC ALL -->> 40 ALL for 1 49162 [nodes read: 3]

SC CAA -->> Error - code not in index [nodes read: 3]

SC SAT -->> 06 SAT awhile 49128 [nodes read: 1]

SC JAZ -->> Error - code not in index [nodes read: 3]

SC YOU -->> 30 YOU & me 49152 [nodes read: 2]

SC DVD -->> 26 DVD or CD 49148 [nodes read: 2]

SC AAA -->> Error - code not in index [nodes read: 3]

SC ZZZ -->> Error - code not in index [nodes read: 3]

SC CON -->> 18 CON artist 49140 [nodes read: 1]

SC HAT -->> 59 HAT & coat 49181 [nodes read: 2]

SC RAT -->> 35 RAT you dirty 49157 [nodes read: 2]

SC AND -->> 09 AND so on 49131 [nodes read: 2]

SC SAM -->> 36 SAM Space 49158 [nodes read: 3]

SC YOZ -->> Error - code not in index [nodes read: 3]

SC WOZ -->> Error - code not in index [nodes read: 3]

SC CAZ -->> Error - code not in index [nodes read: 3]

SC BEG -->> 43 BEG for candy 49165 [nodes read: 2]

========================================

========================================

PROCESSING A5TransData2.txt

SC ZAP -->> 78 ZAP a bug 78901 [nodes read: 1]

SC BAR -->> 12 BAR none 12345 [nodes read: 1]

SC HOT -->> 56 HOT and cold 56789 [nodes read: 1]

SC ZIP -->> Error - code not in index [nodes read: 1]

SC ABE -->> Error - code not in index [nodes read: 1]

SC RAN -->> Error - code not in index [nodes read: 1]

SC RUN -->> 67 RUN spot run 67890 [nodes read: 1]

========================================

========================================

PROCESSING A5TransData3.txt

SC BOX -->> 01 BOX o bits 10001 [nodes read: 1]

SC BEG -->> 21 BEG borrow 10021 [nodes read: 2]

SC AAA -->> Error - code not in index [nodes read: 2]

SC HAM -->> 49 HAM spam 10049 [nodes read: 2]

SC ZZZ -->> Error - code not in index [nodes read: 2]

SC ALL -->> 11 ALL in fun 10011 [nodes read: 2]

SC HUB -->> 28 HUB usb 10028 [nodes read: 2]

SC RUN -->> 10 RUN spot run 10010 [nodes read: 2]

SC LZZ -->> Error - code not in index [nodes read: 2]

SC IKE -->> 04 IKE for Pres 10004 [nodes read: 1]

SC ZIP -->> 31 ZIP per 10031 [nodes read: 2]

SC LAA -->> Error - code not in index [nodes read: 2]

SC MUT -->> Error - code not in index [nodes read: 2]

========================================

**package** edu.wmich.cs3310.asgn5;

**import** java.io.File;

**import** java.io.IOException;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TEST MODE World Data App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** TestDriver {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Controller for TEST MODE World Data App 2.5

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** **main**(String args[]) **throws** IOException {

File file = **new** File("Log.txt");

**if** (file.exists()) file.delete();

**for** (**int** i = 1; i <= 3; i++) UserApp.*main*(i);

}

}

**package** edu.wmich.cs3310.asgn5;

**import** java.io.File;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.Scanner;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Controller for format/functionality.

\* from the TransData files

\* TEST MODE World Data App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** UserApp {

**private** **static** PrintWriter *tL*;

**private** **static** Scanner *input*;

**private** **static** String *code*;

**private** **static** String *command*;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Manage functionalities as requested by TransData files.

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

**public** **static** **void** **main**(**int** fileNameSufix)**throws** IOException {

*tL* = **new** PrintWriter(**new** FileOutputStream(**new** File("Log.txt"), **true**));

File file = **new** File(String.*format*("A5TransData%d.txt", fileNameSufix));

*input* = **new** Scanner(file);

ActualData aD = **new** ActualData(fileNameSufix);

CodeIndex cI= **new** CodeIndex(fileNameSufix, *tL*);

**while**(*grabCommand*())

**switch** (*command*) {

**case** "SC":

*tL*.print("SC " + *code*);

aD.selectByDRP(cI.selectByCode(*code*, *tL*), *tL*);

*tL*.printf("\t[nodes read: %d]\n", cI.getNodesRead());

**break**;

**default**:

*tL*.print(*command* + "\n ERROR, invalid command.");

**break**;

}

*input*.close();

aD.finishUp();

cI.finishUp();

*tL*.print("========================================\n\n");

*tL*.close();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Obtain command from line in TransData file.

\* **@throws** IOException

\*/

**private** **static** **boolean** **grabCommand**() {

**if** (*input*.hasNextLine()) {

String temp = *input*.nextLine();

*command* = temp.substring(0, 2);

**if** (*command*.equals("SC"))

*code* = temp.substring(3, temp.length()).trim();

**return** **true**;

} **else return** **false**;

}

**package** edu.wmich.cs3310.asgn5;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.io.RandomAccessFile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Handles the CodeIndex files.

\* TEST MODE World Data App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** CodeIndex {

**private** RandomAccessFile file;

**private** **short** M;

// private short N;

**private** **short** rootPtr;

**private** **short**[] TP;

**private** String[] KV;

**private** **short**[] DRP;

**private** **int** byteOffset;

**private** **int** sizeOfHeaderRec;

**private** **int** sizeOfDataRec;

**private** **int** nodesRead;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Initializes objects such as the binary file.

\* **@param** fileNameSufix

\* **@param** tL PrintWriter object

\* **@throws** IOException

\*/

**public** **CodeIndex**(**int** fileNameSufix, PrintWriter tL) **throws** IOException {

file = **new** RandomAccessFile(String.*format*("CodeIndex%d.bin",

fileNameSufix), "r");

tL.print("========================================\n");

tL.printf("PROCESSING A5TransData%d.txt\n", fileNameSufix);

M = file.readByte();

rootPtr = file.readByte();

// N = file.readByte();

sizeOfHeaderRec = 3;

sizeOfDataRec = M + 3 \* (M - 1) + (M - 1);

TP = **new** **short**[M];

KV = **new** String[M-1];

DRP = **new** **short**[M-1];

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Locates code in index.

\* **@param** code Element id to locate

\* **@param** tl PrintWriter object

\* **@throws** IOException

\*/

**public** **int** **selectByCode**(String code, PrintWriter tL) **throws** IOException{

nodesRead = 0;

**int** result = searchOneNode(rootPtr, code);

**if** (result == -1) tL.print(" -->> Error - code not in index");

**return** result;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Reads a node from binary file.

\* **@throws** IOException

\*/

**private** **void** **readOneNode**() **throws** IOException{

nodesRead++;

**for**(**int** i = 0; i < M-1; i++)

KV[i] = "";

**for**(**int** i = 0; i < M; i++)

TP[i] = file.readByte();

**for**(**int** i = 0; i < M-1; i++)

**for**(**int** j = 0; j < 3; j++)

KV[i] += (**char**) file.readByte();

**for**(**int** i = 0; i < M-1; i++)

DRP[i] = file.readByte();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Brings node to memory to check and search further.

\* **@param** pointer

\* **@param** code

\* **@return**

\* **@throws** IOException

\*/

**private** **int** **searchOneNode**(**int** pointer, String code) **throws** IOException {

byteOffset(pointer);

readOneNode();

**for** (**int** i = 0; i < M-1; i++)

**if**(code.compareTo(KV[i]) < 0)

**if** (TP[i] != -1)

**return** searchOneNode(TP[i], code);

**else** **return** -1;

**else** **if** (code.compareTo(KV[i]) == 0)

**return** DRP[i];

**else** **if** ((code.compareTo(KV[i]) > 0 && i+1 == M-1)

|| (code.compareTo(KV[i]) > 0 && KV[i+1].equals("]]]")))

**if** (TP[i+1] != -1)

**return** searchOneNode(TP[i+1], code);

**else** **return** -1;

**return** -1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* For calculating and locating byteOffset.

\* **@param** rootPtr

\* **@throws** IOException

\*/

**private** **void** **byteOffset**(**int** rootPtr) **throws** IOException{

byteOffset = sizeOfHeaderRec + ((rootPtr - 1) \* sizeOfDataRec);

file.seek(byteOffset);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Returns number of nodes read in the search.

\* **@return** nodesRead

\*/

**public** **int** **getNodesRead**() {

**return** nodesRead;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Closes binary file.

\* **@throws** IOException

\*/

**public** **void** **finishUp**() **throws** IOException{

file.close();

}

}

**package** edu.wmich.cs3310.asgn5;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.io.RandomAccessFile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Manages the FakeActualData files.

\* TEST MODE World Data App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** ActualData {

**private** RandomAccessFile file;

**private** **int** byteOffset;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Initializes the binary file.

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

**public** **ActualData**(**int** fileNameSufix) **throws** IOException{

file = **new** RandomAccessFile(String.*format*("FakeActualData%d.txt",

fileNameSufix), "r");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Locates element in txt file by country DRP.

\* **@param** DRP Element id to locate

\* **@param** tl PrintWriter object

\* **@throws** IOException

\*/

**public** **void** **selectByDRP**(**int** DRP, PrintWriter tl) **throws** IOException{

**if** (DRP != -1){

String line = "";

byteOffset(DRP);

**for** (**int** i = 0; i < 23; i++)

line += (**char**)file.readByte();

tl.printf(" -->> %s\t", line);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* For calculating byteOffset and seeking it.

\* **@param** DRP

\* **@throws** IOException

\*/

**private** **void** **byteOffset**(**int** DRP) **throws** IOException{

byteOffset = (DRP-1) \* 25;

file.seek(byteOffset);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Closes binary file.

\* **@throws** IOException

\*/

**public** **void** **finishUp**() **throws** IOException{

file.close();

}

}