Countries Of The World App 2.5 - Java

CS 3310 – Dr. Donna Kaminski

Contents

[TheLog.txt 2](#_Toc384072788)

[Main 5](#_Toc384072789)

[UserApp 6](#_Toc384072790)

[CountryIndex 7](#_Toc384072791)

[CountryData 10](#_Toc384072792)

[TheLog 12](#_Toc384072793)

[TransData 13](#_Toc384072794)

# CODE STATUS > UserApp started.

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

PROCESSING A4TransData1.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]

. . .

CODE STATUS > UserApp finished - 19 transactions processed.

CODE STATUS > UserApp started.

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

PROCESSING A4TransData2.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]

. . .

CODE STATUS > UserApp finished - 7 transactions processed.

CODE STATUS > UserApp started.

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

PROCESSING A4TransData3.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]

. . .

CODE STATUS > UserApp finished - 13 transactions processed.

**package** edu.wmich.cs3310.asgn4;

**import** java.io.File;

**import** java.io.IOException;

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

\* Countries Of The World App 2.5

\* Creates table for managing countries information.

\* **@author** Caleb Viola

\*/

# **public** **class** Main {

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

\* Deletes TheLog if it exists, Userapp according to

\* A4DemoSpecs.

\* **@param** args

\* **@throws** IOException

\*/

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

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

**if** (file.exists())

file.delete();

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

UserApp.*userAppMain*(i);

}

}

**package** edu.wmich.cs3310.asgn4;

**import** java.io.IOException;

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

\* Class containing method to manage table modifications from the TransData

\* files.

\* **@author** Caleb Viola

\*/

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

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

\* Method intended to manage table modifications from the TransData files.

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

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

TheLog tL = **new** TheLog();

tL.printThis("CODE STATUS > UserApp started.");

TransData tD = **new** TransData(fileNameSufix, tL);

CountryData cD = **new** CountryData(fileNameSufix, tL);

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

**int** trans = 0;

**while** (!tD.doneWithFile) {

tD.grabCommand(tL);

**if** (!tD.doneWithFile) {

**switch** (tD.getCommand()) {

**case** "SC":

tL.printThis("SC " + tD.getCode());

cD.selectByDRP(cI.selectByCode(tD.getCode(),

tL), tL);

tL.nodesRead(cI.getNodesRead());

trans++;

**break**;

**default**:

tL.printThis(tD.getCommand() + "\n"

+ " ERROR, invalid command.");

trans++;

}

}

}

tD.finishUp(tL);

cD.finishUp(tL, **true**);

cI.finishUp(tL, **true**);

tL.printThis("CODE STATUS > UserApp finished - " + trans

+ " transactions processed.");

tL.finishUp();

}

}

**package** edu.wmich.cs3310.asgn4;

**import** java.io.IOException;

**import** java.io.RandomAccessFile;

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

\* Handles the CodeIndex#UC.bin files

\* **@author** Caleb Viola

\*/

# **public** **class** CountryIndex {

**private** RandomAccessFile file;

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

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

**private** **short** N;

**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** tL

\* **@param** fileName

\* **@throws** IOException

\*/

**public** CountryIndex(**int** fileNameSufix, TheLog tL) **throws** IOException{

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

fileNameSufix), "r");

tL.printThis("========================================");

tL.printThis(String.*format*("PROCESSING

A4TransData%d.txt\n",fileNameSufix));

M = file.readShort();

rootPtr = file.readShort();

N = file.readShort();

sizeOfHeaderRec = (Short.*SIZE*/8)\*3;

sizeOfDataRec = (Short.*SIZE* \* M + Character.*SIZE* \* 3 \* (M - 1) +

Short.*SIZE* \* (M - 1))/8;

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

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

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

nodesRead = 0;

}

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

\* Reads a node from binary file.

\* **@throws** IOException

\*/

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

nodesRead++;

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

KV[i] = "";

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

TP[i] = file.readShort();

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

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

KV[i] += file.readChar();

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

DRP[i] = file.readShort();

}

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

\* Recursive method that searches for data in 1 node.

\* **@param** homeSubscript

\* **@return** DRP if existent, -1 if not.

\* **@throws** IOException

\*/

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

byteOffset(pointer);

readNode();

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

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

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

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

**else**

**return** -1;

}

**return** -1;

}

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

\* Locates code in index.

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

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

nodesRead = 0;

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

**if** (result == -1)

tL.error();

**return** result;

}

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

\* For calculating and locating byteOffset.

\* **@param** RRN

\* **@param** file

\* **@return** spot status.

\* **@throws** IOException

\*/

**public** **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.

\* **@param** tL

\* **@param** b

\* **@throws** IOException

\*/

**public** **void** finishUp(TheLog tL, **boolean** printTable) **throws** IOException {

file.close();

}

}

**package** edu.wmich.cs3310.asgn4;

**import** java.io.IOException;

**import** java.io.RandomAccessFile;

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

\* Manages the CountryData#.txt files.

\* stored in a binary file.

\* Countries Of The World App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** CountryData {

**private** RandomAccessFile file;

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

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

\* Initializes the binary file.

\* **@param** tL

\* **@param** fileName

\* **@throws** IOException

\*/

**public** CountryData(**int** fileNameSufix, TheLog tL) **throws** IOException{

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

fileNameSufix), "r");

}

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

\* Locates element in txt file by country DRP.

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

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

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

String line = "";

byteOffset(DRP);

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

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

tl.printThis(String.*format*(">>> %s",line));

}

}

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

\* For calculating byteOffset and seeking it.

\* **@param** RRN

\* **@param** file

\* **@return** spot status.

\* **@throws** IOException

\*/

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

byteOffset = (DRP-1) \* 25;

file.seek(byteOffset);

}

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

\* Closes binary file.

\* **@param** tL TheLog object

\* **@param** printTable

\* **@throws** IOException

\*/

**public** **void** finishUp(TheLog tL, **boolean** printTable) **throws** IOException{

file.close();

}

}

**package** edu.wmich.cs3310.asgn4;

**import** java.io.\*;

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

\* Writes log entries to a .txt file.

\* Countries Of The World App 2.5

\* **@author** Caleb Viola

\*/

# **public** **class** TheLog {

**private** PrintWriter tL;

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

\* Initializes the PrintWriter object.

\* **@param** append Decides append mode for PrintWriter.

\* **@throws** IOException

\*/

**public** TheLog() **throws** IOException {

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

}

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

\* Nodes read notification.

\* **@param** recRead

\* **@return**

\*/

**public** **void** nodesRead(**int** nodesRead) {

printThis(String.*format*(" [# nodes read: %d]", nodesRead));

}

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

\* Error message.

\* **@param** action

\* **@return** string

\*/

**public** **void** error() {

printThis(">>> Error - code not in index.");

}

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

\* Prints string to log file.

\* **@param** msg

\*/

**public** **void** printThis(String msg){

tL.println(msg);

}

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

\* Closes file log file.

\* **@throws** IOException

\*/

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

tL.close();

}

}

**package** edu.wmich.cs3310.asgn4;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

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

\* Intended to obtain information for UserApp in Main from the TransData files

\* in order to create table.

\* **@author** Caleb Viola

\*/

# **public** **class** TransData {

**private** Scanner input;

**private** String code;

**private** String command;

**boolean** doneWithFile = **false**;

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

\* Constructor to open a TransData file.

\* **@param** fileNameSufix number of TransData file to read.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** TransData(**int** fileNameSufix, TheLog tl) **throws** IOException {

File file = **new** File(String.*format*("A4TransData%d.txt",fileNameSufix));

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

}

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

\* Obtain command from a line in the TransData file.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** grabCommand(TheLog tl) **throws** IOException {

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

String temp = input.nextLine();

command = temp.substring(0, 2);

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

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

} **else**

doneWithFile = **true**;

}

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

\* Getter for acquired command.

\* **@return**

\*/

**public** String getCommand() {

**return** command;

}

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

\* Getter for code.

\* **@return**

\*/

**public** String getCode() {

**return** code;

}

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

\* Closes data file.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

input.close();

tL.printThis(". . .");

}

}