Countries Of The World App 1.0 - Java

CS 3310 – Dr. Donna Kaminski

Index

[Main 2](#_Toc378643056)

[RawData 3](#_Toc378643059)

[TransData 5](#_Toc378643060)

[CountryDataTable 7](#_Toc378643061)

[BstNode 10](#_Toc378643062)

[TheLog 11](#_Toc378643063)

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

/\*\*

\* Countries Of The World App 1.0

\* Creates table for managing contries information

\* **@author** Cuzox

\*/

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

/\*\*

\* (Main) Instantiates TheLog and CountryDataTable objects

\* **@param** args

\* **@throws** IOException

\*/

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

CountryDataTable cdt = **new** CountryDataTable();

TheLog tl = **new** TheLog();

*Setup*("Sample",cdt,tl);

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

*UserApp*(i, cdt, tl);

cdt.FinishUp(tl,**true**);

*Setup*("All",cdt,tl);

**for** (**int** i = 1; i <= 4; i++)

*UserApp*(i, cdt, tl);

cdt.FinishUp(tl,**false**);

tl.FinishUp();

}

/\*\*

\* Method intented to setup CDT from the RawData files.

\* **@throws** IOException

\*/

## **private** **static** **void** Setup(String fileNameSufix,CountryDataTable cdt,TheLog tl) **throws** IOException{

tl.displayThis("CODE STATUS > Setup started");

RawData rd = **new** RawData(fileNameSufix,tl);

**int** temp = 0;

**while** (!rd.doneWithInput){

rd.GrabCountry();

**if** (!rd.doneWithInput)

cdt.Insert(rd.GetName(),rd.GetContinent(),rd.GetCode(),

rd.GetArea(),rd.GetPopulation(),rd.GetLifeExpectancy(),tl, **false**);

}

temp = rd.GetTransactions();

rd.FinishUp(tl);

tl.displayThis("CODE STATUS > Setup finished - "+temp+" countries processed");

}

/\*\*

\* Method intented to setup CDT from the TransData files.

\* **@throws** IOException

\*/

## **private** **static** **void** UserApp(**int** fileNameSufix,CountryDataTable cdt,TheLog tl)

**throws** IOException{

tl.displayThis("CODE STATUS > UserApp started");

TransData td = **new** TransData(fileNameSufix,tl);

**int** temp = 0;

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

td.GrabCommand(tl);

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

**if** (td.TransCode().equals("SN")){

tl.displayThis("SN "+td.GetName());

cdt.SelectByName(td.GetName().toLowerCase(), tl);

}

**else** **if** (td.TransCode().equals("SA")){

tl.displayThis("SA");

cdt.SelectAll(tl);

}

**else** **if** (td.TransCode().equals("IN")){

tl.displayThis("IN "+td.GetCode()+","+td.GetName()+","+td.GetArea()+","+

td.GetPopulation()+","+td.GetLifeExpectancy());

cdt.Insert(td.GetName(),td.GetContinent(),td.GetCode(),td.GetArea(),

td.GetPopulation(),td.GetLifeExpectancy(),tl, **true**);

}

**else** **if** (td.TransCode().equals("DN")){

tl.displayThis("DN "+td.GetName());

cdt.Delete(td.GetName().toLowerCase(), tl);

}

**else**{

tl.displayThis(td.TransCode()+"\n ERROR, invalid command");

}

}

}

temp = td.GetTransactions();

td.FinishUp(tl);

tl.displayThis("CODE STATUS > UserApp finished - "+temp+" transactions processed");

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

/\*\*

\* Intended to obtain information for Setup in Main from

\* the RawData files in order to create table.

\* Countries Of The World App 1.0

\* **@author** Cuzox

\*/

# **public** **class** RawData {

**private** Scanner input;

**private** String name;

**private** String continent;

**private** String code;

**private** **int** area;

**private** **int** population;

**private** **int** transactions;

**private** **float** lifeExpectancy;

**boolean** doneWithInput = **false**;

/\*\*

\* Contructor to open RawData and create log entry

\* **@param** fileNameSufix Name of RawData file to open

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** RawData(String fileNameSufix, TheLog tl) **throws** IOException{

File file = **new** File("RawData"+fileNameSufix+".csv");

tl.displayThis("FILE STATUS > RawData FILE opened");

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

}

/\*\*

\* Obtain command from a line in the RawData file

\*/

**public** **void** GrabCountry(){

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

String temp = input.nextLine();

temp = temp.substring(30,temp.length()-2)

.replace("'","");

String[] fields = temp.split(",");

code = fields[0];

name = fields[1];

continent = fields[2];

area = Integer.*parseInt*(fields[4]);

population = Integer.*parseInt*(fields[6]);

lifeExpectancy = Float.*parseFloat*(fields[7]);

transactions++;

}

**else**

doneWithInput = **true**;

}

/\*\*

\* Getter for name

\* **@return**

\*/

**public** String GetName(){

**return** name;

}

/\*\*

\* Getter for continent

\* **@return**

\*/

**public** String GetContinent(){

**return** continent;

}

/\*\*

\* Getter for code

\* **@return**

\*/

**public** String GetCode(){

**return** code;

}

/\*\*

\* Getter for area

\* **@return**

\*/

**public** **int** GetArea(){

**return** area;

}

/\*\*

\* Getter for population

\* **@return**

\*/

**public** **int** GetPopulation(){

**return** population;

}

/\*\*

\* Getter for lifeExpectancy

\* **@return**

\*/

**public** **float** GetLifeExpectancy(){

**return** lifeExpectancy;

}

/\*\*

\* Getter for transactions

\* **@return**

\*/

**public** **int** GetTransactions(){

**return** transactions;

}

/\*\*

\* Close file and create log entry

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

input.close();

tl.displayThis("FILE STATUS > RawData FILE closed");

transactions = 0;

}

}

**package** edu.wmich.cs3310.asgn1;

**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.

\* Countries Of The World App 1.0

\* **@author** Cuzox

\*/

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

**private** Scanner input;

**private** String name;

**private** String continent;

**private** String code;

**private** String command;

**private** **int** area;

**private** **int** population;

**private** **int** transactions;

**private** **float** lifeExpectancy;

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

/\*\*

\* Constructor to open TransData and create log entry

\* **@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("TransData"+fileNameSufix+".txt");

tl.displayThis("FILE STATUS > TransData FILE opened");

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("IN")){

temp = temp.substring(33,temp.length()-2)

.replace("'","");

String[] fields = temp.split(",");

code = fields[0];

name = fields[1];

continent = fields[2];

area = Integer.*parseInt*(fields[4]);

population = Integer.*parseInt*(fields[6]);

lifeExpectancy = Float.*parseFloat*(fields[7]);

}

**else** **if** (command.equals("SN") || command.equals("DN")){

**if** (temp.length() > 2)

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

**else**

name = "EMPTY";

}

transactions++;

}

**else**

doneWithFile = **true**;

}

/\*\*

\* Getter for acquired command

\* **@return**

\*/

**public** String TransCode(){

**return** command;

}

/\*\*

\* Getter for name

\* **@return**

\*/

**public** String GetName(){

**return** name;

}

/\*\*

\* Getter for continent

\* **@return**

\*/

**public** String GetContinent(){

**return** continent;

}

/\*\*

\* Getter for code

\* **@return**

\*/

**public** String GetCode(){

**return** code;

}

/\*\*

\* Getter for area

\* **@return**

\*/

**public** **int** GetArea(){

**return** area;

}

/\*\*

\* Getter for population

\* **@return**

\*/

**public** **int** GetPopulation(){

**return** population;

}

/\*\*

\* Getter for lifeExpectancy

\* **@return**

\*/

**public** **float** GetLifeExpectancy(){

**return** lifeExpectancy;

}

/\*\*

\* Getter for number of transactions

\* **@return**

\*/

**public** **int** GetTransactions() {

**return** transactions;

}

/\*\*

\* Close file and creat log entry

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

input.close();

tl.displayThis("FILE STATUS > TransData FILE opened");

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

**import** java.text.DecimalFormat;

/\*\*

\* Manages countries in a table

\* Countries Of The World App 1.0

\* **@author** Cuzox

\*/

# **public** **class** CountryDataTable {

**private** **int** RootPtr;

**private** **int** N, nextEmpty;

**private** BstNode[] BstNodes;

/\*\*

\* Constructor

\*/

**public** CountryDataTable(){

RootPtr = -1;

N = 0;

nextEmpty = 0;

BstNodes = **new** BstNode[299];

}

/\*\*

\* **@param** target Node to compare

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** SelectByName(String target, TheLog tl) **throws** IOException{

**int**[] i = SearchForName(target);

**if** (i[0] != -1 && !BstNodes[i[0]].code.equals("XXX"))

tl.displayThis(" "+tl.Country(BstNodes[i[0]].code,BstNodes[i[0]].name,BstNodes[i[0]].continent,

BstNodes[i[0]].area,BstNodes[i[0]].population, BstNodes[i[0]].lifeExpectancy) +"\n"+tl.Visited(i[1]));

**else**

tl.displayThis(tl.Sorry()+"\n"+tl.Visited(i[1]));

}

/\*\*

\* Searches for target in the BST

\* **@param** target

\* **@return**

\*/

**private** **int**[] SearchForName(String target){

**int** i = RootPtr;

**int** visited = RootPtr;

**while** (i != -1 && !target.equals(BstNodes[i].name.toLowerCase())) {

**if** (target.compareTo(BstNodes[i].name.toLowerCase()) < 0)

i = BstNodes[i].LChPtr;

**else**

i = BstNodes[i].RChPtr;

visited++;

}

**int**[] result = {i,visited};

**return** result;

}

/\*\*

\* Execution for SA command which displays table

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

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

tl.displayThis(" "+tl.Header1());

LNR(RootPtr,tl);

tl.displayThis(" "+tl.Footer());

}

/\*\*

\* Inorder traversal algorithm using recursion

\* **@param** i Search index

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**private** **void** LNR(**int** i, TheLog tl) **throws** IOException{

**if**(BstNodes[i].LChPtr != -1){

LNR(BstNodes[i].LChPtr,tl);

**if** (!BstNodes[i].code.equals("XXX"))

tl.displayThis(tl.Country(" "+BstNodes[i].code,BstNodes[i].name,

BstNodes[i].continent,BstNodes[i].area,

BstNodes[i].population,BstNodes[i].lifeExpectancy));

}

**else**

**if** (!BstNodes[i].code.equals("XXX"))

tl.displayThis(tl.Country(" "+BstNodes[i].code, BstNodes[i].name,

BstNodes[i].continent,BstNodes[i].area, BstNodes[i].population,

BstNodes[i].lifeExpectancy));

**if**(BstNodes[i].RChPtr != -1){

LNR(BstNodes[i].RChPtr,tl);

}

}

/\*\*

\* Insert country in table

\* **@param** name

\* **@param** continent

\* **@param** code

\* **@param** area

\* **@param** population

\* **@param** lifeExpectancy

\* **@param** tl TheLog object

\* **@param** userApp boolean to determine if confirmation message should be logged

\* **@throws** IOException

\*/

**public** **void** Insert(String name, String continent, String code, **int** area,

**int** population, **float** lifeExpectancy, TheLog tl,

**boolean** userApp) **throws** IOException{

BstNodes[nextEmpty] = **new** BstNode();

BstNodes[nextEmpty].name = name;

BstNodes[nextEmpty].continent = continent;

BstNodes[nextEmpty].code = code;

BstNodes[nextEmpty].area = area;

BstNodes[nextEmpty].population = population;

BstNodes[nextEmpty].lifeExpectancy = lifeExpectancy;

BstNodes[nextEmpty].LChPtr = -1;

BstNodes[nextEmpty].RChPtr = -1;

**int** visited = 0;

**if** (RootPtr == -1)

RootPtr = N;

**else** {

**int** i = RootPtr;

**int** parentI = RootPtr;

String side = "na";

**while** (i != -1) {

parentI = i;

**if** (BstNodes[nextEmpty].name.compareTo(BstNodes[i].name) < 0){

i = BstNodes[i].LChPtr;

side = "left";

}

**else** **if** (BstNodes[nextEmpty].name.compareTo(BstNodes[i].name) > 0){

i = BstNodes[i].RChPtr;

side = "right";

}

**else**

i = -1;

visited++;

}

**if** (side.equals("left"))

BstNodes[parentI].LChPtr = nextEmpty;

**else** **if** (side.equals("right"))

BstNodes[parentI].RChPtr = nextEmpty;

}

**if** (userApp)

tl.displayThis(tl.Msg("inserted")+"\n"+tl.Visited(visited));

N++;

nextEmpty++;

}

/\*\*

\* Static delete for country in table

\* **@throws** IOException

\*/

**public** **void** Delete(String target, TheLog tl) **throws** IOException{

**int**[] i = SearchForName(target);

**if** (i[0] != -1 && !BstNodes[i[0]].code.equals("XXX")){

BstNodes[i[0]].code = "XXX";

BstNodes[i[0]].continent = " ";

BstNodes[i[0]].area = 0;

BstNodes[i[0]].population = 0;

BstNodes[i[0]].lifeExpectancy = 0;

N--;

tl.displayThis(tl.Msg("deleted"));

}

**else**{

tl.displayThis(tl.Sorry());

}

}

/\*\*

\* Prints all contries and Tobstones

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**private** **void** Snapshot(TheLog tl) **throws** IOException{

tl.displayThis("CODE STATUS > Snapshot started\n");

DecimalFormat noFormat = **new** DecimalFormat("#000");

tl.displayThis("N: "+N+", NextEmpty: "+nextEmpty+", RootPtr: "+RootPtr+"\n");

**int** i = 0;

tl.displayThis(tl.Header2());

**while** (BstNodes[i] != **null**){

**if**(BstNodes[i].code != "XXX")

tl.displayThis(tl.Country2(i,BstNodes[i].code,BstNodes[i].name,

BstNodes[i].continent,BstNodes[i].area,BstNodes[i].population,

BstNodes[i].lifeExpectancy,BstNodes[i].LChPtr,BstNodes[i].RChPtr));

**else**

tl.displayThis("["+noFormat.format(i)+"] "+"TOMBSTONE");

i++;

}

tl.displayThis(tl.Footer()+tl.Footer2());

tl.displayThis("CODE STATUS > Snapshot finished - "+i+" nodes displayed");

}

/\*\*

\* Calls Snapshot utility and restarts information in CDT

\* **@param** tl TheLog object

\* **@param** printTable

\* **@throws** IOException

\*/

**public** **void** FinishUp(TheLog tl, **boolean** printTable) **throws** IOException{

**if** (printTable)

Snapshot(tl);

RootPtr = -1;

N = 0;

nextEmpty = 0;

}

}

**package** edu.wmich.cs3310.asgn1;

/\*\*

\* Node to represent each country in the CDT

\* Countries Of The World App 1.0

\* **@author** Cuzox

\*/

# **public** **class** BstNode {

String name;

String continent;

String code;

**int** area;

**int** population;

**float** lifeExpectancy;

**int** LChPtr, RChPtr;

}

**package** edu.wmich.cs3310.asgn1;

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

**import** java.text.DecimalFormat;

/\*\*

\* Writes log entries to a .txt file

\* Countries Of The World App 1.0

\* **@author** Cuzox

\*/

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

**private** PrintWriter theLog;

/\*\*

\* Constructor

\* **@throws** IOException

\*/

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

theLog = **new** PrintWriter("TheLog.txt");

theLog.println("FILE STATUS > TheLog FILE opened");

}

/\*\*

\* Preformatted string

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** life

\* **@return** String with SA format for country

\*/

**public** String Country(String code, String name, String continent, **int** area,

**int** population, **float** life){

**if**(name.length()>18)

name+="\n ";

DecimalFormat myFormatter = **new** DecimalFormat("#,###");

**return** code+" "+String.*format*("%-18s %-13s %10s %13s %4.1f",

name, continent, myFormatter.format(area),

myFormatter.format(population), life);

}

/\*\*

\* Preformatted string

\* **@param** index

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** life

\* **@param** LCh

\* **@param** RCh

\* **@return** String with Snapshot format for country

\*/

**public** String Country2(**int** index,String code,String name,String continent,**int** area,**int** population,

**float** life,**int** LCh,**int** RCh){

DecimalFormat noFormat = **new** DecimalFormat("#000");

**return** "["+noFormat.format(index)+"] "+Country(code, name, continent, area,

population,life)+String.*format*(" %03d %03d", LCh, RCh);

}

/\*\*

\* Preformatted string

\* **@param** action

\* **@return** confirmation message string

\*/

**public** String Msg(String action){

**return** " OK, country "+action;

}

/\*\*

\* Preformatted string

\* **@return** invalid entry string

\*/

**public** String Sorry(){

**return** " SORRY, invalid country name";

}

/\*\*

\* Preformatted string

\* **@param** no number of nodes

\* **@return** nodes visited string

\*/

**public** String Visited(**int** no){

**return** " >> "+no+" nodes visited";

}

/\*\*

\* Preformatted string

\* **@return** SA header string

\*/

**public** String Header1(){

**return** "CDE NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE";

}

/\*\*

\* Preformatted string

\* **@return** Snapshot header string

\*/

**public** String Header2(){

**return** "[SUB] "+Header1()+" LCh RCh";

}

/\*\*

\* Preformatted string

\* **@return** footer

\*/

**public** String Footer(){

**return** "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++";

}

/\*\*

\* Preformatted string

\* **@return** footer

\*/

**public** String Footer2(){

**return** "++++++++++++++";

}

/\*\*

\* Writes string to file

\* **@param** toPrint string to write to file

\* **@throws** IOException

\*/

**public** **void** displayThis(String toPrint) **throws** IOException{

theLog.println(toPrint);

}

/\*\*

\* Creates log entry and closes file

\* **@throws** IOException

\*/

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

theLog.println("FILE STATUS > TheLog FILE closed");

theLog.close();

}

}