Countries Of The World App 0.5 - Java

CS 3310 – Dr. Donna Kaminski

Index

[Log.txt 2](#_Toc399227547)

[TestDriver 9](#_Toc399227548)

[Setup 10](#_Toc399227549)

[UserApp 11](#_Toc399227550)

[PrettyPrintUtility 13](#_Toc399227551)

[RawData 14](#_Toc399227552)

[UI 17](#_Toc399227553)

[DataTable 20](#_Toc399227554)

[NameIndex 23](#_Toc399227555)

[DataTableRecord 25](#_Toc399227556)

[Log 26](#_Toc399227557)

[Backup 28](#_Toc399227558)

# FILE STATUS > Log FILE opened

CODE STATUS > Setup started

FILE STATUS > RawData FILE opened (A1RawDataSample.csv)

FILE STATUS > RawData FILE closed

CODE STATUS > Setup finished - 19 countries processed

FILE STATUS > Log FILE closed

CODE STATUS > PrettyPrintUtility started

N: 24

DATA STORAGE

LOC/ CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

001/ ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

002/ USA 002 United States North America 9,363,520 278,357,000 77.1

003/ 003 ...

004/ IND 004 India Asia 3,287,263 1,013,662,000 62.5

005/ UMI 005 United States Mino Oceania 16 0 0.0

006/ MCO 006 Monaco Europe 2 34,000 78.8

007/ AND 007 Andorra Europe 468 78,000 83.5

008/ 008 ...

009/ UKR 009 Ukraine Europe 603,700 50,456,000 66.0

010/ RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

011/ SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

012/ VAT 012 Holy See (Vatican Europe 0 1,000 0.0

013/ SGS 013 South Georgia and Antarctica 3,903 0 0.0

014/ CHN 014 China Asia 9,572,900 1,277,558,000 71.4

015/ 015 ...

016/ GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

017/ SYR 017 Syria Asia 185,180 16,125,000 68.5

018/ DEU 018 Germany Europe 357,022 82,164,700 77.4

019/ 019 ...

020/ BRA 020 Brazil South America 8,547,404 170,115,000 62.9

021/ JPN 021 Japan Asia 377,829 126,714,000 80.7

022/ 022 ...

023/ ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

024/ ATA 024 Antarctica Antarctica 13,120,000 0 0.0

NAME INDEX

LOC/ NAME-------------- PTR

000/ Andorra 007

001/ Antarctica 024

002/ Brazil 020

003/ China 014

004/ Germany 018

005/ Holy See (Vatican 012

006/ India 004

007/ Japan 021

008/ Monaco 006

009/ Russian Federation 010

010/ South Africa 001

011/ South Georgia and 013

012/ Sudan 011

013/ Syria 017

014/ Ukraine 009

015/ United Kingdom 016

016/ United States 002

017/ United States Mino 005

018/ Zimbabwe 023

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > PrettyPrintUtility finished

FILE STATUS > Log FILE opened

CODE STATUS > UserApp started

FILE STATUS > TransData FILE opened (A1TransData1.txt)

AN

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

AND 007 Andorra Europe 468 78,000 83.5

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

DEU 018 Germany Europe 357,022 82,164,700 77.4

VAT 012 Holy See (Vatican Europe 0 1,000 0.0

IND 004 India Asia 3,287,263 1,013,662,000 62.5

JPN 021 Japan Asia 377,829 126,714,000 80.7

MCO 006 Monaco Europe 2 34,000 78.8

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

SGS 013 South Georgia and Antarctica 3,903 0 0.0

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

SYR 017 Syria Asia 185,180 16,125,000 68.5

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

USA 002 United States North America 9,363,520 278,357,000 77.1

UMI 005 United States Mino Oceania 16 0 0.0

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

AI

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

USA 002 United States North America 9,363,520 278,357,000 77.1

003 ...

IND 004 India Asia 3,287,263 1,013,662,000 62.5

UMI 005 United States Mino Oceania 16 0 0.0

MCO 006 Monaco Europe 2 34,000 78.8

AND 007 Andorra Europe 468 78,000 83.5

008 ...

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

VAT 012 Holy See (Vatican Europe 0 1,000 0.0

SGS 013 South Georgia and Antarctica 3,903 0 0.0

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

015 ...

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

SYR 017 Syria Asia 185,180 16,125,000 68.5

DEU 018 Germany Europe 357,022 82,164,700 77.4

019 ...

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

JPN 021 Japan Asia 377,829 126,714,000 80.7

022 ...

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SN China

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

SN CHINA

ERROR, invalid country name

SN chinA

ERROR, invalid country name

SN Chin

ERROR, invalid country name

SN China

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

SN China doll

ERROR, invalid country name

SN China=

ERROR, invalid country name

SN China

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

SI 13

SGS 013 South Georgia and Antarctica 3,903 0 0.0

SI 10

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

SI 1

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

SI 24

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

SI 0000024

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

SN United

ERROR, invalid country name

SN United We Stand

ERROR, invalid country name

SN United States

USA 002 United States North America 9,363,520 278,357,000 77.1

SN United States of America

ERROR, invalid country name

SI 0

ERROR, invalid country id

SI 999999999

ERROR, invalid country id

SI 25

ERROR, invalid country id

SN South Georgia and the South Sandwich Islands

ERROR, invalid country name

SN South Georgia

ERROR, invalid country name

SN Russia

ERROR, invalid country name

SN Russian Federation

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

SI 19

ERROR, invalid country id

SI 8

ERROR, invalid country id

SN Egypt

ERROR, invalid country name

SN Aaa

ERROR, invalid country name

SN Zoo

ERROR, invalid country name

SN Antarctica

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

SN Zimbabwe

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

SN 2

ERROR, invalid country name

SI 2

USA 002 United States North America 9,363,520 278,357,000 77.1

FILE STATUS > TransData FILE closed

CODE STATUS > UserApp finished - 35 transactions processed

FILE STATUS > Log FILE closed

FILE STATUS > Log FILE opened

CODE STATUS > UserApp started

FILE STATUS > TransData FILE opened (A1TransData2.txt)

FILE STATUS > TransData FILE closed

CODE STATUS > UserApp finished - 0 transactions processed

FILE STATUS > Log FILE closed

FILE STATUS > Log FILE opened

CODE STATUS > UserApp started

FILE STATUS > TransData FILE opened (A1TransData3.txt)

AN

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

AND 007 Andorra Europe 468 78,000 83.5

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

DEU 018 Germany Europe 357,022 82,164,700 77.4

VAT 012 Holy See (Vatican Europe 0 1,000 0.0

IND 004 India Asia 3,287,263 1,013,662,000 62.5

JPN 021 Japan Asia 377,829 126,714,000 80.7

MCO 006 Monaco Europe 2 34,000 78.8

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

SGS 013 South Georgia and Antarctica 3,903 0 0.0

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

SYR 017 Syria Asia 185,180 16,125,000 68.5

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

USA 002 United States North America 9,363,520 278,357,000 77.1

UMI 005 United States Mino Oceania 16 0 0.0

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

AI

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

USA 002 United States North America 9,363,520 278,357,000 77.1

003 ...

IND 004 India Asia 3,287,263 1,013,662,000 62.5

UMI 005 United States Mino Oceania 16 0 0.0

MCO 006 Monaco Europe 2 34,000 78.8

AND 007 Andorra Europe 468 78,000 83.5

008 ...

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

VAT 012 Holy See (Vatican Europe 0 1,000 0.0

SGS 013 South Georgia and Antarctica 3,903 0 0.0

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

015 ...

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

SYR 017 Syria Asia 185,180 16,125,000 68.5

DEU 018 Germany Europe 357,022 82,164,700 77.4

019 ...

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

JPN 021 Japan Asia 377,829 126,714,000 80.7

022 ...

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

DN United Kingdom

[SORRY, Delete By Name module not yet working]

DI 10

[SORRY, Delete By Id module not yet working]

DN Computer Science

[SORRY, Delete By Name module not yet working]

DI 0

[SORRY, Delete By Id module not yet working]

DI 25

[SORRY, Delete By Id module not yet working]

SN United Kingdom

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

SI 10

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

IN ARG,Argentina,2780400,37032000,75.1

OK, country inserted (in data storage & name index)

IN JAM,Jamaica,10990,2583000,75.2

OK, country inserted (in data storage & name index)

IN CUB,Cuba,110861,11201000,76.2

OK, country inserted (in data storage & name index)

IN AUT,Austria,83859,8091800,77.7

OK, country inserted (in data storage & name index)

IN BWA,Botswana,581730,1622000,39.3

OK, country inserted (in data storage & name index)

IN KAZ,Kazakstan,2724900,16223000,63.2

OK, country inserted (in data storage & name index)

SN Kazakstan

KAZ 008 Kazakstan Asia 2,724,900 16,223,000 63.2

SN Botswana

BWA 012 Botswana Africa 581,730 1,622,000 39.3

SI 8

KAZ 008 Kazakstan Asia 2,724,900 16,223,000 63.2

SI 3

CUB 003 Cuba North America 110,861 11,201,000 76.2

SI 26

ARG 026 Argentina South America 2,780,400 37,032,000 75.1

SN Argentina

ARG 026 Argentina South America 2,780,400 37,032,000 75.1

SN Jamaica

JAM 025 Jamaica North America 10,990 2,583,000 75.2

SI 25

JAM 025 Jamaica North America 10,990 2,583,000 75.2

SI 12

BWA 012 Botswana Africa 581,730 1,622,000 39.3

SN Cuba

CUB 003 Cuba North America 110,861 11,201,000 76.2

SN Austria

AUT 019 Austria Europe 83,859 8,091,800 77.7

SI 19

AUT 019 Austria Europe 83,859 8,091,800 77.7

AI

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

USA 002 United States North America 9,363,520 278,357,000 77.1

CUB 003 Cuba North America 110,861 11,201,000 76.2

IND 004 India Asia 3,287,263 1,013,662,000 62.5

UMI 005 United States Mino Oceania 16 0 0.0

MCO 006 Monaco Europe 2 34,000 78.8

AND 007 Andorra Europe 468 78,000 83.5

KAZ 008 Kazakstan Asia 2,724,900 16,223,000 63.2

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

BWA 012 Botswana Africa 581,730 1,622,000 39.3

SGS 013 South Georgia and Antarctica 3,903 0 0.0

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

015 ...

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

SYR 017 Syria Asia 185,180 16,125,000 68.5

DEU 018 Germany Europe 357,022 82,164,700 77.4

AUT 019 Austria Europe 83,859 8,091,800 77.7

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

JPN 021 Japan Asia 377,829 126,714,000 80.7

022 ...

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

JAM 025 Jamaica North America 10,990 2,583,000 75.2

ARG 026 Argentina South America 2,780,400 37,032,000 75.1

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

AN

CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

AND 007 Andorra Europe 468 78,000 83.5

ATA 024 Antarctica Antarctica 13,120,000 0 0.0

ARG 026 Argentina South America 2,780,400 37,032,000 75.1

AUT 019 Austria Europe 83,859 8,091,800 77.7

BWA 012 Botswana Africa 581,730 1,622,000 39.3

BRA 020 Brazil South America 8,547,404 170,115,000 62.9

CHN 014 China Asia 9,572,900 1,277,558,000 71.4

CUB 003 Cuba North America 110,861 11,201,000 76.2

DEU 018 Germany Europe 357,022 82,164,700 77.4

BWA 012 Botswana Africa 581,730 1,622,000 39.3

IND 004 India Asia 3,287,263 1,013,662,000 62.5

JAM 025 Jamaica North America 10,990 2,583,000 75.2

JPN 021 Japan Asia 377,829 126,714,000 80.7

KAZ 008 Kazakstan Asia 2,724,900 16,223,000 63.2

MCO 006 Monaco Europe 2 34,000 78.8

RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

SGS 013 South Georgia and Antarctica 3,903 0 0.0

SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

SYR 017 Syria Asia 185,180 16,125,000 68.5

UKR 009 Ukraine Europe 603,700 50,456,000 66.0

GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

USA 002 United States North America 9,363,520 278,357,000 77.1

UMI 005 United States Mino Oceania 16 0 0.0

ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FILE STATUS > TransData FILE closed

CODE STATUS > UserApp finished - 29 transactions processed

FILE STATUS > Log FILE closed

CODE STATUS > PrettyPrintUtility started

N: 26

DATA STORAGE

LOC/ CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE

001/ ZAF 001 South Africa Africa 1,221,037 40,377,000 51.1

002/ USA 002 United States North America 9,363,520 278,357,000 77.1

003/ CUB 003 Cuba North America 110,861 11,201,000 76.2

004/ IND 004 India Asia 3,287,263 1,013,662,000 62.5

005/ UMI 005 United States Mino Oceania 16 0 0.0

006/ MCO 006 Monaco Europe 2 34,000 78.8

007/ AND 007 Andorra Europe 468 78,000 83.5

008/ KAZ 008 Kazakstan Asia 2,724,900 16,223,000 63.2

009/ UKR 009 Ukraine Europe 603,700 50,456,000 66.0

010/ RUS 010 Russian Federation Europe 17,075,400 146,934,000 67.2

011/ SDN 011 Sudan Africa 2,505,813 29,490,000 56.6

012/ BWA 012 Botswana Africa 581,730 1,622,000 39.3

013/ SGS 013 South Georgia and Antarctica 3,903 0 0.0

014/ CHN 014 China Asia 9,572,900 1,277,558,000 71.4

015/ 015 ...

016/ GBR 016 United Kingdom Europe 242,900 59,623,400 77.7

017/ SYR 017 Syria Asia 185,180 16,125,000 68.5

018/ DEU 018 Germany Europe 357,022 82,164,700 77.4

019/ AUT 019 Austria Europe 83,859 8,091,800 77.7

020/ BRA 020 Brazil South America 8,547,404 170,115,000 62.9

021/ JPN 021 Japan Asia 377,829 126,714,000 80.7

022/ 022 ...

023/ ZWE 023 Zimbabwe Africa 390,757 11,669,000 37.8

024/ ATA 024 Antarctica Antarctica 13,120,000 0 0.0

025/ JAM 025 Jamaica North America 10,990 2,583,000 75.2

026/ ARG 026 Argentina South America 2,780,400 37,032,000 75.1

NAME INDEX

LOC/ NAME-------------- PTR

000/ Andorra 007

001/ Antarctica 024

002/ Argentina 026

003/ Austria 019

004/ Botswana 012

005/ Brazil 020

006/ China 014

007/ Cuba 003

008/ Germany 018

009/ Holy See (Vatican 012

010/ India 004

011/ Jamaica 025

012/ Japan 021

013/ Kazakstan 008

014/ Monaco 006

015/ Russian Federation 010

016/ South Africa 001

017/ South Georgia and 013

018/ Sudan 011

019/ Syria 017

020/ Ukraine 009

021/ United Kingdom 016

022/ United States 002

023/ United States Mino 005

024/ Zimbabwe 023

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > PrettyPrintUtility finished

FILE STATUS > Log FILE opened

CODE STATUS > Setup started

FILE STATUS > RawData FILE opened (A1RawData.csv)

FILE STATUS > RawData FILE closed

CODE STATUS > Setup finished - 239 countries processed

FILE STATUS > Log FILE closed

FILE STATUS > Log FILE opened

CODE STATUS > UserApp started

FILE STATUS > TransData FILE opened (A1TransData4.txt)

SN United States Minor Outlying Islands

ERROR, invalid country name

SN Afghanistan

AFG 001 Afghanistan Asia 652,090 22,720,000 45.9

SI 1

AFG 001 Afghanistan Asia 652,090 22,720,000 45.9

SN United States

USA 229 United States North America 9,363,520 278,357,000 77.1

SI 239

UMI 239 United States Mino Oceania 16 0 0.0

SN China

CHN 094 China Asia 9,572,900 1,277,558,000 71.4

SI 0

ERROR, invalid country id

SI 240

ERROR, invalid country id

SN Vatican

ERROR, invalid country name

SN Monaco

MCO 136 Monaco Europe 2 34,000 78.8

SN AAA

ERROR, invalid country name

SN ZZZ

ERROR, invalid country name

SN Zimbabwe

ZWE 231 Zimbabwe Africa 390,757 11,669,000 37.8

SI 94

CHN 094 China Asia 9,572,900 1,277,558,000 71.4

FILE STATUS > TransData FILE closed

CODE STATUS > UserApp finished - 14 transactions processed

FILE STATUS > Log FILE closed

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

**import** java.io.File;

**import** java.io.IOException;

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

\* Creates table for managing country information

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

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

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

\* According to A1DemoSpecs

\* **@param** args

\* **@throws** IOException

\*/

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

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

File backup = **new** File("Backup.txt");

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

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

Setup.*main*("Sample");

PrettyPrintUtility.*main*();

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

UserApp.*main*(i);

PrettyPrintUtility.*main*();

Setup.*main*("");

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

UserApp.*main*(i);

}

}

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

**import** java.io.IOException;

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

\* Class containing method to setup CDT from the RawData files

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** Setup {

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

\* Setup DataTable and NameIndex from RawData files

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

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

Log log = **new** Log(**true**);

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

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

DataTable dt = **new** DataTable();

NameIndex ni = **new** NameIndex();

**int** N = 0;

rd.grabCountry();

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

**if**(rd.countryGrabbed){

dt.insert(rd.getCode(), rd.getId(), rd.getName(),

rd.getContinent(), rd.getArea(), rd.getPopulation(),

rd.getLifeExpectancy(), **false**, log);

ni.insert(rd.getName(), rd.getId());

}

rd.grabCountry();

}

N = rd.getTransactions();

rd.finishUp(log);

dt.finishUp();

ni.finishUp();

log.displayThis("CODE STATUS > Setup finished - "+N+" countries processed");

log.finishUp();

}

}

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

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.util.Scanner;

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

\* Interact with DataTable from TransData files

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

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

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

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

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

Log log = **new** Log(**true**);

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

UI ui = **new** UI(fileNameSufix, log);

DataTable dt = **new** DataTable();

NameIndex ni = **new** NameIndex();

*load*(dt, ni);

**int** N = 0;

ui.grabCommand();

**while** (!ui.doneWithTrans){

**switch**(ui.getTransCode()){

**case** "IN":

log.displayThis("IN " + ui.getCode() + "," + ui.getName() + ","

+ ui.getArea() + "," + ui.getPopulation() + ","

+ ui.getLifeExpectancy());

dt.insert(ui.getCode(), ui.getId(), ui.getName(),

ui.getContinent(), ui.getArea(), ui.getPopulation(),

ui.getLifeExpectancy(), **true**, log);

ni.insert(ui.getName(), ui.getId());

**break**;

**case** "DN":

log.displayThis("DN "+ui.getName());

ni.deleteByName(dt, ui.getName(), log);

**break**;

**case** "DI":

log.displayThis("DI "+ui.getId());

dt.deleteById(ui.getId(), log);

**break**;

**case** "SN":

log.displayThis("SN "+ui.getName());

ni.selectByName(dt, ui.getName(), log);

**break**;

**case** "SI":

log.displayThis("SI "+ui.getOrigId());

dt.selectById(ui.getId(), log);

**break**;

**case** "AN":

log.displayThis("AN");

ni.allByName(dt, log);

**break**;

**case** "AI":

log.displayThis("AI");

dt.allById(log);

**break**;

**default**:

log.displayThis(ui.getTransCode()+"\n" +

" ERROR, invalid command");

**break**;

}

ui.grabCommand();

}

N = ui.getTransactions();

ui.finishUp(log);

dt.finishUp();

ni.finishUp();

log.displayThis("CODE STATUS > UserApp finished - " + N

+ " transactions processed");

log.finishUp();

}

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

\* Load DataTable and NameIdex from Backup.txt

\* **@param** dt

\* **@param** ni

\* **@throws** IOException

\*/

**private** **static** **void** **load**(DataTable dt, NameIndex ni) **throws** IOException {

File backup = **new** File("Backup.txt");

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

String temp;

String line[];

**int** N = Integer.*parseInt*(input.nextLine());

**for** (**short** x = 1; x <= N; x++){

temp = input.nextLine();

**if** (!temp.equals("empty")){

line = temp.replace(",", "").split("\_");

dt.load(line[0], Integer.*parseInt*(line[1]), line[2], line[3],

Integer.*parseInt*(line[4]), Integer.*parseInt*(line[5]),

Float.*parseFloat*(line[6]));

}**else**

dt.loadNull();

}

**while** (input.hasNextLine()){

line = input.nextLine().split("\_");

ni.load(line[0], Integer.*parseInt*(line[1]));

}

input.close();

}

}

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

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

**import** java.text.DecimalFormat;

**import** java.util.Scanner;

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

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** PrettyPrintUtility {

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

\* Displays the backup file to the log file

\* **@throws** IOException

\*/

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

File backup = **new** File("Backup.txt");

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

PrintWriter log = **new** PrintWriter(**new** FileOutputStream(**new** File(

"Log.txt"), **true**));

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

String temp;

String line[];

**int** N = Integer.*parseInt*(input.nextLine());

log.println("CODE STATUS > PrettyPrintUtility started");

log.println("N: " + N);

log.println("DATA STORAGE");

log.println("LOC/ CDE ID- NAME-------------- CONTINENT---- ------AREA ---POPULATION LIFE");

**for** (**short** x = 1; x <= N; x++) {

temp = input.nextLine();

**if** (!temp.equals("empty")) {

line = temp.split("\_");

log.println(String.*format*(

"%s/ %s %s %-18s %-13s %10s %13s %4.1f",

hundredZeros.format(x), line[0], line[1], line[2],

line[3], line[4], line[5], Float.*parseFloat*(line[6])));

} **else**

log.println(String.*format*("%s/ %s ...",

hundredZeros.format(x), hundredZeros.format(x)));

}

log.println("");

log.println("NAME INDEX");

log.println("LOC/ NAME-------------- PTR");

**int** i = 0;

**while** (input.hasNextLine()){

line = input.nextLine().split("\_");

log.println(String.*format*("%s/ %-18s %s",

hundredZeros.format(i++), line[0], line[1]));

}

log.println("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

log.println("CODE STATUS > PrettyPrintUtility finished");

input.close();

log.close();

}

}

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

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

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

\* Read RawData files and organize contents for retrieval

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

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

**private** Scanner input;

**private** String name;

**private** String continent;

**private** String code;

**private** **int** id;

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

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

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

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

**private** Log log;

**boolean** doneWithInput;

**boolean** countryGrabbed;

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

\* Contructor

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

\* **@param** Log TheLog object

\* **@throws** IOException

\*/

**public** **RawData**(String fileNameSufix, Log log) **throws** IOException{

File file = **new** File("A1RawData"+fileNameSufix+".csv");

log.displayThis(String.*format*(

"FILE STATUS > RawData FILE opened (A1RawData%s.csv)",

fileNameSufix));

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

}

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

\* Obtain country from a line from the RawData file

\*/

**public** **void** **grabCountry**(){

doneWithInput = **false**;

countryGrabbed = **false**;

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

String temp = input.nextLine();

**if** (temp.length() > 2 && temp.substring(0,6).equals("INSERT")){

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

.replace("'","");

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

code = fields[0];

id = Integer.*parseInt*(fields[1]);

name = fields[2];

continent = fields[3];

area = Math.*round*(Float.*parseFloat*(fields[5]));

population = Integer.*parseInt*(fields[7]);

lifeExpectancy = Float.*parseFloat*(fields[8]);

transactions++;

countryGrabbed = **true**;

}**else**

countryGrabbed = **false**;

}**else**

doneWithInput = **true**;

}

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

\* Getter for code

\* **@return**

\*/

**public** String **getCode**(){

**return** code;

}

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

\* Getter for id

\* **@return**

\*/

**public** **int** **getId**(){

**return** id;

}

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

\* Getter for name

\* **@return**

\*/

**public** String **getName**(){

**return** name;

}

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

\* Getter for continent

\* **@return**

\*/

**public** String **getContinent**(){

**return** continent;

}

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

\* 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** Log TheLog object

\* **@throws** IOException

\*/

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

input.close();

log.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;

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

\* Read TransData files prepare for UserApp

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** UI {

**private** Scanner input;

**private** String name;

**private** String continent;

**private** String code;

**private** String transCode;

**private** String origId;

**private** **int** id;

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

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

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

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

**boolean** doneWithTrans = **false**;

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

\* Constructor to open TransData and create log entry

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

\* **@param** log

\* **@throws** IOException

\*/

**public** **UI**(**int** fileNameSufix, Log log) **throws** IOException{

File file = **new** File("A1TransData"+fileNameSufix+".txt");

log.displayThis(String.*format*(

"FILE STATUS > TransData FILE opened (A1TransData%s.txt)",

fileNameSufix));

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

}

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

\* Obtain command from a line in the TransData file

\* **@throws** IOException

\*/

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

doneWithTrans = **false**;

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

String temp = input.nextLine();

transCode = temp.substring(0,2);

**if** (transCode.equals("IN")){

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

.replace("'","");

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

code = fields[0];

id = Integer.*parseInt*(fields[1]);

name = fields[2];

continent = fields[3];

area = Math.*round*(Float.*parseFloat*(fields[5]));

population = Integer.*parseInt*(fields[7]);

lifeExpectancy = Float.*parseFloat*(fields[8]);

}

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

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

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

**else**

name = "EMPTY";

}**else** **if** (transCode.endsWith("SI") || transCode.endsWith("DI"))

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

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

id = Integer.*parseInt*(origId.replaceFirst("^0+(?!$)", ""));

}

**else**

id = 0;

transactions++;

}

**else**

doneWithTrans = **true**;

}

/\*\*

\* Getter for acquired command

\* **@return**

\*/

**public** String **getTransCode**(){

**return** transCode;

}

/\*\*

\* Getter for code

\* **@return**

\*/

**public** String **getCode**(){

**return** code;

}

/\*\*

\* Getter for id

\* **@return**

\*/

**public** **int** **getId**(){

**return** id;

}

/\*\*

\* Getter for origId

\* **@return**

\*/

**public** String **getOrigId**(){

**return** origId;

}

/\*\*

\* Getter for name

\* **@return**

\*/

**public** String **getName**(){

**return** name;

}

/\*\*

\* Getter for continent

\* **@return**

\*/

**public** String **getContinent**(){

**return** continent;

}

/\*\*

\* 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 create log entry

\* **@param** ppu TheLog object

\* **@throws** IOException

\*/

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

input.close();

log.displayThis("FILE STATUS > TransData FILE closed");

}

}

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

**import** java.io.IOException;

**import** java.util.ArrayList;

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

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** DataTable {

**private** ArrayList<DataTableRecord> dataTable;

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

\* Initialize array and set index 0 as null

\*/

**public** **DataTable**(){

dataTable = **new** ArrayList<DataTableRecord>();

dataTable.add(**null**);

}

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

\* IN command

\* **@param** code

\* **@param** id

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** lifeExpectancy

\* **@param** userApp

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **insert**(String code, **int** id, String name, String continent,

**int** area, **int** population, **float** lifeExpectancy, **boolean** userApp,

Log log) **throws** IOException {

DataTableRecord record = **new** DataTableRecord();

record.code = code;

record.id = id;

record.name = name;

record.continent = continent;

record.area = area;

record.population = population;

record.lifeExpectancy = lifeExpectancy;

**int** diff = id - dataTable.size();

**if** (diff >= 1){

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

dataTable.add(**null**);

dataTable.add(record);

}

**else** **if** (diff == 0)

dataTable.add(record);

**else**

dataTable.set(id, record);

**if** (userApp)

log.displayThis(" OK, country inserted (in data storage & name index)");

}

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

\* Loads directly to table

\* **@param** code

\* **@param** id

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** lifeExpectancy

\* **@throws** IOException

\*/

**public** **void** **load**(String code, **int** id, String name, String continent,

**int** area, **int** population, **float** lifeExpectancy) **throws** IOException {

DataTableRecord record = **new** DataTableRecord();

record.code = code;

record.id = id;

record.name = name;

record.continent = continent;

record.area = area;

record.population = population;

record.lifeExpectancy = lifeExpectancy;

dataTable.add(record);

}

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

\* Places null un table

\*/

**public** **void** **loadNull**(){

dataTable.add(**null**);

}

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

\* DI command

\* **@param** id

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **deleteById**(**int** id, Log log) **throws** IOException{

log.displayThis(" [SORRY, Delete By Id module not yet working]");

}

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

\* SI command

\* **@param** id

\* **@throws** IOException

\*/

**public** **void** **selectById**(**int** id, Log log) **throws** IOException{

**if** ((id > 0) && (id < dataTable.size()) && (dataTable.get(id) != **null**))

log.displayThis(" "+log.country(dataTable.get(id)));

**else**

log.displayThis(" ERROR, invalid country id");

}

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

\* AI command which displays table in id order

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **allById**(Log log) **throws** IOException{

log.displayThis(" "+log.header());

**for** (**short** x = 0; x < dataTable.size(); x++)

**if** (dataTable.get(x) != **null**)

log.displayThis(" "+log.country(dataTable.get(x)));

**else** **if** (x != 0)

log.displayThis(" "+log.empty(x));

log.displayThis(" "+log.footer());

}

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

\* Dumps dataTable into Backup.txt

\* **@throws** IOException

\*/

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

Backup backup = **new** Backup(**false**);

backup.displayThis((dataTable.size()-1));

**for** (**short** x = 0; x < dataTable.size(); x++)

**if** (dataTable.get(x) != **null**)

backup.displayThis(backup.dataFormat(dataTable.get(x)));

**else** **if** (x != 0)

backup.displayThis(backup.empty());

backup.finishUp();

}

}

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

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.jar.Attributes.Name;

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

\* Countries Of The World App 0.5

\* **@author** Cuzox

\*/

# **public** **class** NameIndex {

**private** ArrayList<String> keyValue;

**private** ArrayList<Integer> drp;

**private** **int** N, index;

**public** **NameIndex**(){

keyValue = **new** ArrayList<String>();

drp = **new** ArrayList<Integer>();

N = 0;

}

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

\* Insertion sort to index

\* **@param** keyValue

\* **@param** drp

\*/

**public** **void** **insert** (String keyValue, **int** drp){

**if** (N > 0){

index = **this**.keyValue.size()-1;

**boolean** allocated = **false**;

**while** (!allocated){

**if** (keyValue.compareToIgnoreCase(**this**.keyValue.get(index)) > 0){

**if** (index == **this**.keyValue.size()-1){

**this**.keyValue.add(keyValue);

**this**.drp.add(drp);

allocated = **true**;

N++;

}**else**{

**this**.keyValue.add(index+1, keyValue);

**this**.drp.add(index+1, drp);

allocated = **true**;

N++;

}

}**else** **if** (keyValue.compareToIgnoreCase(**this**.keyValue.get(index)) < 0)

**if** (index == 0){

**this**.keyValue.add(index, keyValue);

**this**.drp.add(index, drp);

allocated = **true**;

N++;

}**else**

index--;

}

}**else**{

**this**.keyValue.add(keyValue);

**this**.drp.add(drp);

N++;

}

}

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

\* Loads directly to index

\* **@param** keyValue

\* **@param** drp

\*/

**public** **void** **load** (String keyValue, **int** drp){

**this**.keyValue.add(keyValue);

**this**.drp.add(drp);

N++;

}

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

\* SN command

\* **@param** dt

\* **@param** name

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **selectByName** (DataTable dt, String name, Log log) **throws** IOException{

**if** (keyValue.contains(name)){

dt.selectById(drp.get(keyValue.indexOf(name)), log);

}**else**

log.displayThis(" ERROR, invalid country name");

}

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

\* DN command

\* **@param** dr

\* **@param** name

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **deleteByName** (DataTable dr, String name, Log log) **throws** IOException{

log.displayThis(" [SORRY, Delete By Name module not yet working]");

}

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

\* AN command

\* **@param** dt

\* **@param** log

\* **@throws** IOException

\*/

**public** **void** **allByName**(DataTable dt, Log log) **throws** IOException {

log.displayThis(" "+log.header());

**for** (Integer id : drp)

dt.selectById(id, log);

log.displayThis(" "+log.footer());

}

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

\* Returns object count

\* **@return**

\*/

**public** **int** **getN**(){

**return** N;

}

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

\* Dumps nameIndex to Backup.txt

\* **@throws** IOException

\*/

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

Backup backup = **new** Backup(**true**);

**for** (**short** x = 0; x < keyValue.size(); x++)

backup.displayThis(backup.nameFormat(keyValue.get(x), drp.get(x)));

backup.finishUp();

}

}

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

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

\* Record to represent each country in the DataTable

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** DataTableRecord {

String code;

**int** id;

String name;

String continent;

**int** area;

**int** population;

**float** lifeExpectancy;

}

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

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

**import** java.text.DecimalFormat;

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

\* Manages Log.txt

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** Log {

**private** PrintWriter log;

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

\* Opens Log.txt in append or truncate mode

\* **@param** append

\* **@throws** IOException

\*/

**public** **Log**(**boolean** append) **throws** IOException{

**if** (!append)

log = **new** PrintWriter("Log.txt");

**else**

log = **new** PrintWriter(**new** FileOutputStream(**new** File("Log.txt"),

append));

log.println("FILE STATUS > Log FILE opened");

}

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

\* Formatted string for printing country

\* **@param** record

\* **@return**

\*/

**public** String **country**(DataTableRecord record){

String name = record.name;

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

name = name.substring(0, 18); //Cuts off part of names longer than 18 characters

}

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

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

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

hundredZeros.format(record.id), name, record.continent,

thousand.format(record.area),

thousand.format(record.population), record.lifeExpectancy);

}

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

\* Formatted string for printing empty location

\* **@param** id

\* **@return**

\*/

**public** String **empty**(**int** id){

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

**return** String.*format*(" %s ...", hundredZeros.format(id));

}

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

\* Formatted string for header in AI and AN

\* **@return** SA header string

\*/

**public** String **header**(){

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

}

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

\* Formatted screen for footer in AI and AN

\* **@return** footer

\*/

**public** String **footer**(){

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

}

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

\* Prints line to file

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

\* **@throws** IOException

\*/

**public** <**T**> **void** **displayThis**(**T** message) **throws** IOException{

log.println(message);

}

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

\* Creates log entry and closes file

\* **@throws** IOException

\*/

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

log.println("FILE STATUS > Log FILE closed");

log.close();

}

}

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

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

**import** java.text.DecimalFormat;

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

\* Manages Backup.txt

\* Countries Of The World App 0.5

\* **@author** Caleb Viola

\*/

# **public** **class** Backup {

**private** PrintWriter backup;

**private** DecimalFormat hundredZeros;

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

\* Constructor

\* **@throws** IOException

\*/

**public** **Backup**(**boolean** append) **throws** IOException{

**if** (!append)

backup = **new** PrintWriter("Backup.txt");

**else**

backup = **new** PrintWriter(**new** FileOutputStream(**new** File("Backup.txt"), append));

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

}

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

\* String format for dataTable print on backup file

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** life

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

\*/

**public** String **dataFormat**(DataTableRecord record){

String name = record.name;

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

name = name.substring(0, 18); //Cuts off part of names longer than 18 characters

}

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

**return** String.*format*("%s\_%s\_%s\_%s\_%s\_%s\_%f",

record.code, hundredZeros.format(record.id), name, record.continent,

thousand.format(record.area), thousand.format(record.population),

record.lifeExpectancy);

}

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

\* String format for nameIndex print on backup file

\* **@param** name

\* **@param** ptr

\* **@return**

\*/

**public** String **nameFormat**(String name, **int** ptr){

String nameCopy = name;

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

nameCopy = nameCopy.substring(0, 18); //Cuts off part of names longer than 18 characters

}

**return** String.*format*("%s\_%s",

nameCopy, hundredZeros.format(ptr));

}

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

\* Represent empty location in backup file

\* **@return**

\*/

**public** String **empty**(){

**return** String.*format*("empty");

}

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

\* Prints line to file

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

\* **@throws** IOException

\*/

**public** <**T**> **void** **displayThis**(**T** message) **throws** IOException{

backup.println(message);

}

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

\* Closes backup file

\* **@throws** IOException

\*/

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

backup.close();

}

}