

CountryData.java

```
* Countries of the World App 1.5
import java.io.*;

class CountryData {
    private TheLog tLog;
    private RandomAccessFile rAF;
    private CountryDataRec currentRecord;
    private CountryIndex index;
    private byte sizeOfRecord = 84;
    private byte sizeOfHeader = 2;
    private int counter = 0;
    private int N;
    private String mode;
    private short MAX = 30;

    public CountryData(TheLog tLog, String mode) {

        initialize();

        this.mode = mode;

        this.tLog = tLog;

        this.index = new CountryIndex(tLog);

        openFile();

        if (mode.equals("UserApp")) {

            N = readHeader();

        }
    }

    // *****
    // *****
    private short hashFunction(short id) {

        short homeAddress = (short) (id % MAX);

        if (homeAddress == 0) {

            homeAddress = MAX;

        }

        return homeAddress;

    }

    // *****
    // *****Write*****
    private void Write1Country(short id) {

        try {

            movePointer(sizeOfHeader + ((id - 1) * sizeOfRecord));

            rAF.writeShort(currentRecord.getId());
            rAF.writeChars(currentRecord.getCountryCode());
            rAF.writeChars(currentRecord.getName());
            rAF.writeChars(currentRecord.getContinent());
            rAF.writeInt(currentRecord.getSize());
            rAF.writeLong(currentRecord.getPopulation());

        }

    }

}
```

CountryData.java

```
        rAF.writeFloat(currentRecord.getLifeExp());

        N++;
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// *****
// *****Sequential Read*****
private void Read1Country() throws IOException {

    short id = rAF.readShort();

    if (id == 0) {
        return;
    }

    String code = rAF.readChar() + "" + rAF.readChar() + ""
        + rAF.readChar();

    String name = "";
    String continent = "";

    for (int i = 0; i < 16; i++) {
        name = name + rAF.readChar();
    }

    for (int i = 0; i < 13; i++) {
        continent = continent + rAF.readChar();
    }

    currentRecord = new CountryDataRec(id, code, name, continent,
        rAF.readInt(), rAF.readLong(), rAF.readFloat());
}

// *****
// *****Random Read*****
private void Read1Country(short RRN) {

    try {

        movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

        Read1Country();

    } catch (IOException e) {

        e.printStackTrace();

    }

}
```

CountryData.java

```
// *****//
public short insertSetup(short id, String countryCode, String name,
String continent, int size, long population, float lifeExp) {
    currentRecord = new CountryDataRec(id, countryCode, name, continent,
size, population, lifeExp);
    short homeRRN = hashFunction(id);

    currentRecord.setHomeAddress(homeRRN);
    if (!empty(homeRRN)) {
        homeRRN = collision(homeRRN);
    }
    Write1Country(homeRRN);

    return homeRRN;
}

// *****//
public void insertUser(short id, String countryCode, String name,
String continent, int size, long population, float lifeExp) {

    if (exists(id) == -1) {

        tLog.transProcess("    SORRY, another country has that id");

        return;
    }

    insertSetup(id, countryCode, name, continent, size, population, lifeExp);

    tLog.transProcess("    OK, country inserted");
}

public short collision(short rrn) {

    while (!empty(rrn)) {

        rrn++;

        if (rrn >= MAX) {

            rrn = 1;

        }

    }

    return rrn;
}

// *****//
// **deleteByID checks to make sure the record exists , if it exists
// **we move our file pointer to the beginning of the record and writes
// **it to all 0's
// *****//
public boolean deleteByID(int RRN) {

    tLog.transProcess("    SORRY, deleteByID not yet working");

    return false;
}

}
```

CountryData.java

```
// *****  
// *****  
public boolean deleteByCode(String code) {  
  
    tLog.transProcess("    SORRY, deleteByCode not yet working");  
  
    return false;  
  
}  
  
// *****  
// *****  
public void selectByID(short RRN) {  
  
    RRN = hashFunction(RRN);  
  
    if (RRN < 1 || RRN > getFileSize() || empty(RRN) || exists(RRN) == -1) {  
  
        tLog.transProcess("    SORRY, no country with that id");  
  
        return;  
  
    }  
  
    Read1Country(exists(RRN));  
  
    tLog.toLog(currentRecord.displayThis());  
  
    currentRecord.wipeFields();  
  
}  
  
// *****  
// *****  
public void selectByCode(String code) {  
  
    if (exists(code) == -1) {  
  
        tLog.transProcess("    " + "SORRY, no country with that code");  
  
        counter = 0;  
  
    }  
  
    else {  
  
        int home = index.findHome(code);  
  
        home = exists(code);  
  
        tLog.toLog(currentRecord.displayThis());  
    }  
    tLog.transProcess("    " + counter + " data records read");  
  
}  
  
// *****  
// **This method returns true if a records exist  
// *****  
public short exists(short RRN) {  
  
    counter = 1;  
  
    try {
```

CountryData.java

```

movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

if (rAF.readShort() == RRN) {

    movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

    return RRN;

}

} catch (IOException e) {

    return -1;

}

movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

return -1;

}

// *****
// **This method returns true if a records exist
// *****
public int exists(String code) {

    int RRN = currentRecord.getId();

    try {

        movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

        if (rAF.readShort() == RRN) {

            movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

            return RRN;

        }

    } catch (IOException e) {

        return -1;

    }

    movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

    return -1;

}

// *****
// **This method returns true if a position is empty
// *****
public boolean empty(short RRN) {

    try {

        movePointer(sizeofHeader + ((RRN - 1) * sizeofRecord));

        if (rAF.readShort() == 0) {

```

CountryData.java

```

        movePointer(sizeOfHeader + ((RRN - 1) * sizeOfRecord));

        return true;
    }

    } catch (IOException e) {

        return true;
    }

    movePointer(sizeOfHeader + ((RRN - 1) * sizeOfRecord));

    return false;
}

// *****
// **This method moves the file pointer and handles any exception
// *****
public void movePointer(int byteNum) {

    try {

        rAF.seek(byteNum);

    } catch (IOException e) {

        e.printStackTrace();

    }

}

// *****
// ** This method gets the size of the file being handled
// *****
public long getFileSize() {

    try {

        long fileLength = rAF.length();

        return fileLength;

    } catch (IOException e) {

        e.printStackTrace();

    }

    return -1;

}

// *****
public void initialize() {

    currentRecord = new CountryDataRec((short) 0, "", "", "", 0, 0, 0);

}

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

```

CountryData.java

```

        return N;
    }

    // *****//
    public void snapshot() {
        openFile();

        tLog.transProcess("SnapShot> N:" + N);

        tLog.transProcess(" [RRN] ID CDE NAME----- CONTINENT---- -----AREA ---POPULATION LIFE");

        snapshot(1);

        tLog.transProcess(" ++++++");

        index.snapShot();

        closeFile();
    }

    // *****//
    public void snapshot(int i) {
        if (i >= MAX) {
            return;
        }
        try {

            movePointer(sizeofHeader + ((i - 1) * sizeofRecord));

            Read1Country();

            DecimalFormat formatter = new DecimalFormat("#,###.##");

            if (currentRecord.getId() == 0) {

                tLog.toLog(String.format("[%03d]", i) + " EMPTY\n");

            } else {

                tLog.toLog(String.format(
                    "[%03d] %03d %-3s %-16s %-13s %10s %13s %-4s" + "\n",
                    i, currentRecord.getId(),
                    currentRecord.getCountryCode(),
                    currentRecord.getName(), currentRecord.getContinent(),
                    formatter.format(currentRecord.getSize()),

                    formatter.format(currentRecord.getPopulation()),
                    formatter.format(currentRecord.getLifeExp())));

            }

            currentRecord.wipeFields();

            snapshot(i + 1);

        } catch (IOException e) {

        }
    }

    // *****//
    private void writeHeader() {

```

CountryData.java

```
try {
    movePointer(0);
    rAF.writeShort(N);
} catch (IOException e) {
    e.printStackTrace();
}
}

// *****
private short readHeader() {
    try {
        movePointer(0);
        return rAF.readShort();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return 0;
}

// *****
private void closeFile() {
    try {
        rAF.close();
        tLog.statusFile("CountryData FILE closed");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// *****
public void openFile() {
    try {
        rAF = new RandomAccessFile("CountryData.bin", "rw");
        tLog.statusFile("CountryData FILE opened");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// *****
```


CountryData.java

```
public void finishUp(boolean printTable) {  
    if (mode.equals("UserApp")) {  
        writeHeader();  
    }  
    closeFile();  
    if (printTable == true) {  
        snapshot();  
    }  
}  
}
```