SUBJECT: Data Structures and Algorithms

Submitted To: Dr. Mohsin Ali Memon

Submitted By: Asadullah Samo (21SW036)

Section: Section-III

Dated: 12/03/2023

Assignment: Problem Based Assignment

**Note: For all supporting files used in this project refer to**

<https://github.com/AsadullahSamo/DSA/tree/main/DSA%20Project/src>

Be sure to download these files to run project without errors

**Step 01:**

Use [Reverse Geocoding library for Java (daniel-braun.com)](https://www.daniel-braun.com/technik/reverse-geocoding-library-for-java/) library to find out the city and country from the given coordinates and store them in yearly earthquake collection along with magnitude. (Collection of each year means 52 collections)

ALGORITHM:

1. Make array of arraylist with 53 size (as there are 52 countries, so 52 objects, one initialization for each object) and initialize them all in for loop. One extra object to avoid “indexoutofbounds” exception.
2. Read from “yearly earthquake.txt” (it is created using all valid cities and countries where there is no unable to geocode) line by line and whenever , is encountered store it in array.
3. In nested for loop (inside while loop) store the contents of array index by index to arraylist[index]. (In this way we have successfully stored all countries, cities and magnitude of earthquake for one complete year ).
4. After storing in arraylist[index], update index to next object.
5. Loop till entire file is read.
6. After complete file is readen we will have 52 objects, with each object having details of cities, countries and magnitude for one complete year.

**Note:** Other methods used in step 1, 2, 3, OR 4 classes are supporting methods for problems. For step 1 supporting methods are getArrayList() and getAllCountriesForYear().

Source Code:

**import** eu.bitm.NominatimReverseGeocoding.Address;

**import** eu.bitm.NominatimReverseGeocoding.NominatimReverseGeocodingJAPI;

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

**import** java.nio.charset.StandardCharsets;

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**class** Step1 {

**public** ArrayList<String>[] getArrayList(){

// 53 objects of arraylist for 52 countries (one extra object to avoid indexOutofBounds Exception )

ArrayList<String>[] arrayList = **new** ArrayList[53];

**for** (**int** i = 0; i < arrayList.length; i++) {

arrayList[i] = **new** ArrayList<>();

}

**int** index = 0;

**try**{

// Read Yearly Earthquake.txt from respective location

BufferedReader br = **new** BufferedReader(**new** InputStreamReader(**new** FileInputStream("E:\\DSA\\DSA Project\\src\\Files\\Yearly Earthquake.txt"), StandardCharsets.***UTF\_8***));

String line;

**while** ((line = br.readLine())!=**null**){

String [] array = line.split(", "); // if , is encountered in line, split it and store in array

**if** (index==52){

**break**;

}

**for** (**int** i=0; i< array.length; i++) {

arrayList[index].add(array[i]); // add every element of array to arrayList[index]

}

index++; // when line ends, change index to new object

}

} **catch** (IOException e) {

**throw** **new** RuntimeException(e);

}

**return** arrayList;

} // end of getArrayList() method

**public** **void** getAllCountriesForYear(){

ArrayList<String>[] arrayList = getArrayList();

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the year to get all countries with cities and magnitude: ");

**int** year = sc.nextInt();

System.***out***.println(arrayList[year-1965]); // As the year starts from 1965, and array from 0, so year 1965 gets index 0, 1966 gets index 1 and so on

} // end of getAllCountriesForYear() method

// To get average number of earthquakes for a country, we use below method

**public** **void** getEarthquakesForACountry(){

Scanner sc = **new** Scanner(System.***in***);

ArrayList<String>[] arrayList = getArrayList();

System.***out***.print("Enter year: ");

**int** year = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter the country name: ");

String con = sc.nextLine();

System.***out***.print("[");

String [] array = arrayList[year - 1965].toArray(**new** String[0]);

**for** (**int** i=0; i<array.length; i++){

**if** (array[i].contains(con))

System.***out***.print(array[i]+", "); // Search for country in year

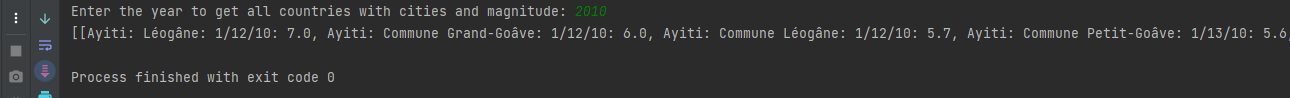
}

System.***out***.println("]");

} // end of getEarthquakesForACountry() method

} // end of class Step1

OUTPUT:



**Step 02:**

Make a queue storing biggest (with highest magnitude) quake of each year with magnitude and country, starting from 1965 to 2016. (52 elements in the queue approx.).

ALGORITHM:

1. Initialize Queue with generic type Object.
2. Read from “Max Magnitude For Queue.txt” (It contains max magnitudes for each country) character by character till end of file
3. Initialize a StringBuilder object
4. In while loop, check if ‘\n’ occurs in file (It means end of line reached). If it occurs, store entire StringBuilder object to queue. Else append the character to StringBuilder object.
5. Once entire file is read, we have our queue with 52 elements

Note: Supporting method for Step 2 is “addCountryWithLargestMagnitude()” which builds queue and “getLargestMagnitudeForEachCountry()” which displays it. Third method in step 2 is for problem 4

Source Code:

**import** java.io.BufferedReader;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.util.LinkedList;

**import** java.util.Queue;

**import** java.util.Scanner;

**class** Step2 {

**public** Queue<Object> addCountryWithLargestMagnitude(){

Queue<Object> largestMagnitude = **new** LinkedList<>();

**try** {

// Read “Max Magnitude For Queue.txt” from respective location

BufferedReader br = **new** BufferedReader(**new** FileReader("E:\\DSA\\DSA Project\\src\\Files\\Max Magnitude For Queue.txt"));

**int** c;

StringBuilder line = **new** StringBuilder();

**while** ((c = br.read()) != -1) {

**if** (c == '\n') { // End of line reached, So add magnitude and country in Queue

largestMagnitude.add(String.*valueOf*(line));

line.setLength(0); // Reset the line buffer for the next line

} **else** {

// Add the current character to the line (StringBuilder)

line.append((**char**) c);

}

} // end of while loop

} **catch** (IOException ie){

ie.printStackTrace();

} // end of try catch block

**return** largestMagnitude;

} // end of addCountryWithLargestMagnitude() method

**public** **void** getLargestMagnitudeForEachCountry(){

Queue<Object> largestMagnitude = addCountryWithLargestMagnitude();

Object [] array = largestMagnitude.toArray(); // convert queue to array, to apply index accessing operations

System.***out***.println(largestMagnitude); // Print entire queue

System.***out***.println("The size of queue is " + largestMagnitude.size());

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the year to get its largest magnitude and country: ");

**int** year = sc.nextInt();

System.***out***.println(array[year-1965]); // as year starts from 1965 and index, from 0 so year 1965 gets 0 index

} // end of getLargestMagnitudeForEachCountry() method

// Get the largest magnitude from starting to ending year, USING BELOW METHOD

**public** **void** getLargestMagnitudeForListOfCountries(){

Queue<Object> largestMagnitude = addCountryWithLargestMagnitude();

Object [] array = largestMagnitude.toArray(); // convert queue to array, to apply index accessing operations

Scanner sc = **new** Scanner(System.***in***);

**int** start, end;

System.***out***.print("Enter starting year: ");

start = sc.nextInt();

System.***out***.print("Enter ending year: ");

end = sc.nextInt();

System.***out***.print("[");

**for** (**int** i=start; i<=end; i++){

**if** (i==end){

System.***out***.print(array[i-1965]);

} **else** {

System.***out***.print(array[i-1965] + ", ");

}

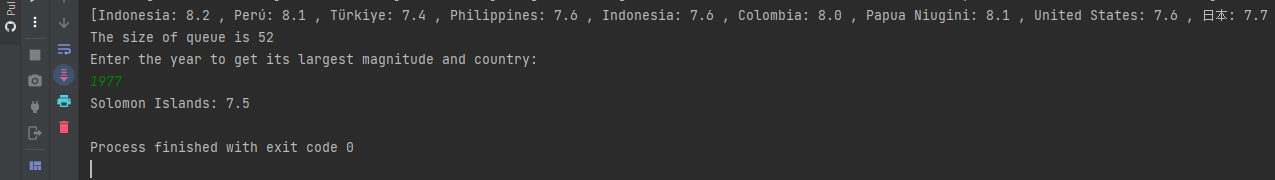
} // end of for loop

System.***out***.println("]");

} // end of getLargestMagnitudeForListOfCountries() method

} // end of class Step2

OUTPUT:

****

**Step 03:**

Make a stack from the collections, one for each country which stores earthquake and its magnitude in the order of the event (the most recent event on top).

ALGORITHM:

1. Make array of stack (As we have to make one object for each country, so each array index will be representing one object) and initialize each object in for loop. (We made 110 sized array because there are 110 countries available to read)
2. Read from “Stack.txt” (it is created using all countries, and at one line is one country which stores earthquake and magnitude in the order of event from 1965 to 2016) line by line and whenever “,” is encountered split line and store in array.
3. In nested for loop (inside while loop) store the contents of array index by index to stack[index]. (In this way we have stored successfully a country with earthquake and magnitude in the order of event from 1965 to 2016
4. Update index at every iteration of while loop to update object

**Note**: Supporting methods for Step 3 are “getStack()” which builds stack and “getDetailsOfCountry()” which displays it. “mostNumberOfEarthquakes” is for Problem 1 and “getNRecentEarthquakesForEachCountry” is for problem 3

Source Code:

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

**import** java.nio.charset.StandardCharsets;

**import** java.sql.\*;

**import** java.util.Scanner;

**import** java.util.Stack;

**class** Step3{

// Find index of given country

**public** **int** getIndex(String country){

String [] countries = {"United States", "中国", "Indonesia", "ایران", "Philippines", "Chile", "Vanuatu",

"臺灣", "Papua Niugini", "Ελλάς", "Argentina", "नेपाल", "日本", "México", "República Dominicana",

"Solomon Islands", "South Georgia and the South Sandwich Islands", "Colombia", "Perú",

"Antigua and Barbuda", "Zambia", "Guatemala", "မြန်မာ", "République démocratique du Congo",

"New Zealand/Aotearoa", "Venezuela", "Монгол улс ᠮᠤᠩᠭᠤᠯ ᠤᠯᠤᠰ", "Bolivia", "Россия",

"Türkiye", "Тоҷикистон", "France", "Brasil", "India", "Ecuador", "Tonga", "Viti", "Australia",

"Panamá", "Belau", "Canada", "Ísland", "افغانستان", "پاکستان", "འབྲུགཡུལ་", "Costa Rica",

"Nicaragua", "British Indian Ocean Territory", "Shqipëria", "ኢትዮጵያ", "مصر", "Italia",

"South Africa", "El Salvador", "Bosna i Hercegovina / Босна и Херцеговина", "Кыргызстан",

"Oʻzbekiston", "Djibouti جيبوتي", "Қазақстан", "România", "ประเทศไทย", "Madagasikara / Madagascar",

"Gabon", "Norge", "Cuba", "España", "Tanzania", "België / Belgique / Belgien", "Malaysia", "Србија",

"Crna Gora / Црна Гора", "ኤርትራ Eritrea إرتريا", "Algérie / ⵍⵣⵣⴰⵢⴻⵔ / الجزائر", "Việt Nam",

"ປະເທດລາວ", "Polska", "Honduras", "العراق", "Türkmenistan",

"Saint Kitts and Nevis", "Guinée", "اليمن", "საქართველო", "বাংলাদেশ",

"Kalaallit Nunaat", "Malawi", "Portugal", "Kenya", "South Sudan", "Micronesia",

"България", "Uganda", "Hrvatska", "Maroc / ⵍⵎⵖⵔⵉⴱ / المغرب", "Jamaica", "السودان",

"Trinidad and Tobago", "Հայաստան", "Северна Македонија", "Paraguay / Paraguái", "Κύπρος - Kıbrıs",

"السعودية", "Dominica", "Azərbaycan", "سوريا", "Soomaaliya الصومال", "Slovenija", "Kosova / Kosovo",

"Moçambique", "Ayiti", ""};

**for** (**int** i=0; i<countries.length; i++){

**if** (countries[i].equals(country)){

**return** i;

}

} // end of for loop

**return** -1;

} // end of getIndex() method

**public** Stack<Object>[] getStack() **throws** SQLException, ClassNotFoundException, IOException {

Stack<Object>[] stack = **new** Stack[110]; // As there are 110 countries, so stack has 110 objects, one for each country

**for** (**int** i=0; i<stack.length; i++){

stack[i] = **new** Stack<>(); // initialize all 110 objects

}

**int** index = 0;

**try** {

// Read from "Stack.txt" from respective location

BufferedReader br = **new** BufferedReader(**new** InputStreamReader(**new** FileInputStream("E:\\DSA\\DSA Project\\src\\Files\\Stack.txt"), StandardCharsets.***UTF\_8***));

String line;

**while** ((line = br.readLine()) != **null**) { // Read entire line

String[] details = line.split(","); // split the line by commas and store entire string to array after , is encountered

**for** (**int** i=0; i<details.length; i++) {

stack[index].push(details[i]); // Put entire details array index by index to object of stack

}

index++; // now storing occurs in next stack object

} // end of while loop

} **catch** (IOException e) {

**throw** **new** RuntimeException(e);

} // end of try catch block

**return** stack;

} // end of method getStack()

**public** **void** getDetailsOfCountry() **throws** SQLException, ClassNotFoundException, IOException {

Stack<Object>[] stack = getStack();

// System.out.println(stack.length);

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the name of country to get its details: ");

String con = sc.nextLine();

**int** i = getIndex(con);

**while** (i==-1) { // loop till valid country is provided

**if** (i == -1) {

System.***out***.println("Please enter valid country name: ");

}

System.***out***.println("Enter the name of country to get its details: ");

con = sc.nextLine();

i = getIndex(con);

} // end of while loop

System.***out***.println(stack[i]); // Print details of country with magnitude from 1965 to 2016 (recent event on top)

} // end of getDetailsOfCountry() method

// Get n recent earthquakes through below method for particular country

**public** **void** getNRecentEarthqakesForEachCountry() **throws** SQLException, IOException, ClassNotFoundException {

Stack<Object>[] stack = getStack();

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the name of country you want to get earthquakes of: ");

String country = sc.nextLine();

System.***out***.println("How many recent earthquakes you want to get? ");

**int** recentN = sc.nextInt();

**int** index = getIndex(country);

Object[] array = stack[index].toArray(); // convert that entire stack object to array to apply index accessing operations

**int** num = array.length - recentN; // used as starting index for loop

System.***out***.println("Recent "+recentN+" earthquakes are ");

System.***out***.print("[");

**for** (**int** i=num; i<array.length; i++){

**if** (i==array.length-1)

System.***out***.print(array[i]);

**else**

System.***out***.print(array[i] + ", ");

} // end of for loop

System.***out***.println("]");

} // end of getNRecentEarthquakesForEachCountry() method

// Stack object with most size will be most vulnerable to earthquakes

**public** **int** mostNumberOfEarthquakes() **throws** SQLException, IOException, ClassNotFoundException {

Stack<Object>[] stack = getStack();

**int** max = stack[0].size();

String str = "";

**for** (**int** i=1; i<stack.length; i++){

**if** (stack[i].size() > max){

max = stack[i].size(); // if any object's size is greater that first object's size, assign max to this object

str = (String) stack[i].peek(); // get top of stack

str = str.substring(0, str.indexOf(":")); // get country from str, which is found when : is encountered (For more help refer to Stack.txt)

}

} // end of for loop

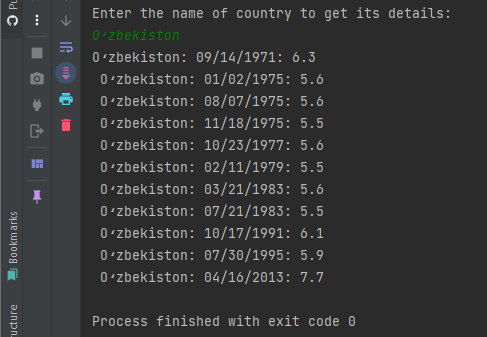
System.***out***.println("Most vulnerable country is "+str);

**return** max;

} // end of mostNumberOfEarthquakes() method

} // end of class Step3

OUTPUT:

****

**Step 04:**

Make a linked list which saves the one most recent earthquake with magnitude and country name from each country (use the stack from step 3).

ALGORITHM:

1. Call getStack() method from step3 class to build Stack with recent events for each country
2. Make a linkedList and initialize it
3. In for loop. Store most recent event for each country in every node of linkedList using peek() method of stack (As recent event will be on top)
4. Take input from user and display method accordingly

**Note**: Supporting method for Step 4 is “getRecentEventForEachCountry()” which builds LL and store recent event in it. “getEarthquakesForParticularMagnitudeRange” is for Problem 4.

Source Code:

**import** java.io.IOException;

**import** java.sql.SQLException;

**import** java.util.LinkedList;

**import** java.util.Scanner;

**import** java.util.Stack;

**class** Step4 {

**public** **void** getRecentEventForEachCountry() **throws** SQLException, ClassNotFoundException, IOException {

Step3 step3 = **new** Step3();

Stack<Object>[] stack = step3.getStack(); // call getStack() from step3 class

LinkedList<Object> linkedList = **new** LinkedList<>();

**for** (**int** i=0; i<stack.length; i++){

linkedList.add(stack[i].peek());

}

Scanner sc = **new** Scanner(System.in);

System.out.println("Enter the name of country to get its recent event: ");

String country = sc.nextLine();

**int** index = step3.getIndex(country);

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

System.out.println("Recent earthquake is " + linkedList.get(index));

}

} // end of getRecentEventForEachCountry() method

**public** **void** getEarthquakesForParticularMagnitudeRange() **throws** SQLException, IOException, ClassNotFoundException {

Step3 step3 = **new** Step3();

Stack<Object>[] stack = step3.getStack(); // call getStack() from step3 class

LinkedList<Object> linkedList = **new** LinkedList<>();

**for** (**int** i=0; i<stack.length; i++){

linkedList.add(stack[i].peek());

}

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the magnitude to get earthquakes above this magnitude: ");

**double** userMag = sc.nextDouble();

**for** (**int** i=0; i<linkedList.size(); i++){

String str = linkedList.get(i).toString();

**double** magnitude = Double.*parseDouble*(str.substring(str.length()-3)); // As magnitude is at last of every string in node, we extracted it using substring from last

**if** (magnitude > userMag){

System.***out***.println(linkedList.get(i));

}

} // end of for loop

} // end of getEarthquakesForParticularMagnitudeRange() method

} // end of class Step 4

OUTPUT:

**Text

Description automatically generated with medium confidence**

**Text

Description automatically generated with low confidence**

**Link:** [**https://github.com/AsadullahSamo/DSA/tree/main/DSA%20Project/src**](https://github.com/AsadullahSamo/DSA/tree/main/DSA%20Project/src)

**To**  **Entire project.java file which contains all problems and steps execution**

**Problem 01:**

How to find the average number of earthquakes per year for each country and which country is most vulnerable to earthquakes (which country has the greatest number of earthquakes)?

ALGORITHM:

**Note: Supporting method is “**getEarthquakesForACountry()”

Description of getEarthquakesForACountry method:

1. Call “getArrayList() method” which has details of magnitude, city and country for each year and store it in arraylist
2. Input year and country name from user
3. Make a array and populate it with the year and country details, provided by the user.
4. We used arraylist[year-1965] because year will be any (ranged from 1965 to 2016) so whichever year is given we subtract it from 1965 and we will get respective index to print year with city, country and magnitude

Description of mostNumberOfEarthquakes() method:

1. Call “getStack() method” which has magnitude for a country from 1965 to 2016(recent event on top)
2. Create a max variable and assign first object of stack to it
3. In for loop check if any object has more size than first, if yes, assign that object to max.
4. Make str and assign any value of that object of stack to it. As in each object there is a lot strings and in each string, there is country, date and magnitude separated by “:”. So we extracted substring from string using substring() method starting from index 0 to the index of “:”
5. Print str and return max from method

Note: Call “getEarthquakesForACountry()” to print average number of earthquakes for a country. And mostNumberOfEarthquakes() to get most vulnerable country

Source Code:

// Average number of earthquakes for a country

public class Project{

public static void main(String [] args){

Step1 step1 = new Step1();

step1.getEarthquakesForACountry

// (Click [here](#getEarthquakesForACountry) to Navigate to this method definition)

// Most Vulnerable country

Step1 step3 = new Step3();

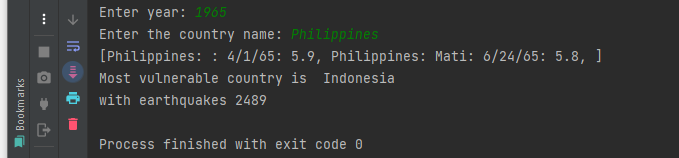
System.out.println("with earthquakes " + step3.mostNumberOfEarthquakes());

// (Click [here](#mostNumberOfEarthquakes) to navigate to method definition)

} // end of main() method

} // end of program

OUTPUT:

****

**Problem 02:**

Which are the biggest earthquakes from 2005 to 2015 and occurred and in which country (use step 2)?

ALGORITHM:

**Note: Supporting method is “**getLargestMagnitudeForListOfCountries()”

Description of getLargestMagnitudeForListOfCountries() method:

1. Call “addCountryWithLargestMagnitude() method” which stores largest magnitude and country of each year and store it in queue
2. Convert queue to array
3. Take starting and ending year from user
4. Loop and print all countries and magnitude from start to end

Source Code:

**public** **class** Project{

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

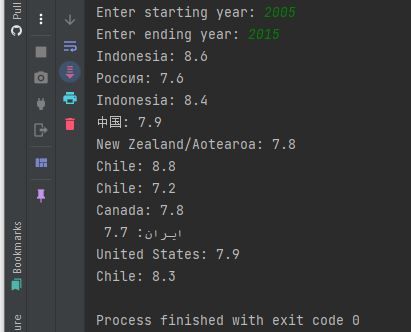
Step2 step2 = **new** Step2();

step2.getLargestMagnitudeForListOfCountries(); // (Click [here](#getLargestMagnitudeForListOfCountries) to navigate to method definition)

}

}

OUTPUT:

****

**Problem 03:**

How to determine the recent 5 earthquakes from each country?

ALGORITHM:

**Note: Supporting method is “**getNRecentEarthquakesForEachCountry()”

Description of getNRecentEarthquakesForEachCountry() method:

1. Call “getStack() method” which has details of country along with magnitude from 1965 to 2016 and store it in stack object.
2. Take country, and how many recent earthquakes to print from user
3. Get Index of country provided by user using getIndex() method
4. Convert stack[index] object to array
5. Create num variable which will be equal to (length of array – recentN) which will give starting index for for loop.
6. Print n recent earthquakes.

Source Code:

**public** **class** Project{

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

// Problem 3: Recent n earthquakes for a given country

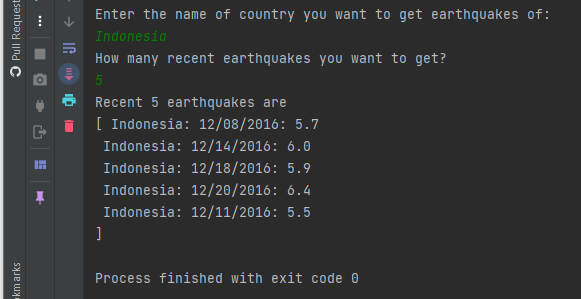
Step3 step3 = **new** Step3();

step3.getNRecentEarthqakesForEachCountry(); // (Click [here](#getNRecentEarthqakesForEachCountry) to navigate to this method's definition)

} // end of main() method

} // end of program

OUTPUT:

****

**Problem 04:**

How to find the most recent above 6 magnitude earthquakes (use step 4).

ALGORITHM:

**Note: Supporting method is “**getEarthquakesForParticularMagnitudeRange()”

Description of getEarthquakesForParticularMagnitudeRange() method:

1. Call “getStack() method” which has details of country along with magnitude from 1965 to 2016 and store it in stack object.
2. Make a linkedList and populate it with recent earthquake for each year using peek() method
3. Take magnitude from user to print all countries earthquake above this magnitude
4. In for loop, convert every stack element to string, extract magnitude from it using substring method(For help refer to file Stack.txt)
5. Check if any magnitude > userMagnitude print it

Source Code:

**public** **class** Project{

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

// Problem 04: Recent earthquakes for a particular magnitude

Step4 step4 = **new** Step4();

step4.getEarthquakesForParticularMagnitudeRange(); // (Click [here](#getEarthquakesForParticularMagRange) to navigate to this method's definition)

} // end of main() method

} // end of program

OUTPUT:

**Text

Description automatically generated**