For this assignment we had to take an Ada program for grade distribution and convert it to Java. We then had to change the input loop so that the frequency was only updated in the exception handler. To accomplish this, I used a try-catch in the input loop. It first checked if the input was less than 0 or greater than 100. If it was it would throw an ArithmeticException and print the error message. If the user ended text input, it would catch the NumberFormatException and print a message. Finally, if the input was valid, it would throw an IOException and run the updateFrequency method to add to the frequency array.

/\*

\*Joseph Camacho-Terrazas

\*10/02/2020

\*Input: A grade

\*Output: The distribution of the grades among the given limits

\*Preconditions: The user must enter a grade within the range 0-89. The user must enter -1 to stop the input and print the distribution.

\*Postcondition: The limits and their frequencies will be properly printed and formatted.

\*/

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.Collections;

public class grade\_distribution {

    //Create array lists for grade limits and frequency counters

    private ArrayList<Integer> limits = new ArrayList<>();

    private ArrayList<Integer> frequencies = new ArrayList<>();

    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);

        //Create an array to store input grades

        ArrayList<Integer> grades = new ArrayList<Integer>();

        //Create a grade\_distribution object

        grade\_distribution Distribution = new grade\_distribution();

        //Read in grades, will stop input loop when -1 is typed

        int input = 0;

        System.out.println("Enter in grades. Type -1 to end input.");

        do{

            try {

                input = Integer.parseInt(reader.nextLine());

                if(input != -1){

                    //Throw ArithmeticException if input is out of bounds

                    if(input < 0 || input > 100) {

                        throw new ArithmeticException();

                    }

                    //Throw IOException to add to frequency array

                    else {

                        throw new IOException();

                    }

                }

            //Handle cases of out-of-bounds input, text input, and valid input

            } catch (ArithmeticException e) {

                System.out.println("Error -- new grade: " + input + " is out of range.");

            } catch (IOException i) {

                Distribution.updateFrequency(grades, input);

            } catch (NumberFormatException n) {

                System.out.println("Invalid input, please input integers only!");

            }

        }while(input != -1);

        //Print the final output

        Distribution.printFrequency();

    }

    //Constructor creates arraylists for the limits and sets all frequency counters to 0

    public grade\_distribution(){

        Collections.addAll(limits, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 101);

        Collections.addAll(frequencies, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

    }

    public void updateFrequency(ArrayList<Integer> gradeList, int userInput){

        //Add valid user input to grade arraylist

        gradeList.add(userInput);

        //Loop through grade limits list and check the grade against each range

        for(int i = 0; i < limits.size() -1 ; i++){

            if(userInput >= limits.get(i) && userInput < limits.get(i + 1)){

                //Add to appropriate frequency counter

                frequencies.set(i, frequencies.get(i) + 1);

            }

        }

    }

    //Print out the frequency table

    public void printFrequency(){

        System.out.println(String.format("%-2s %-20s %-20s", "", "Limits", "Frequency"));

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

            System.out.print(String.format("%-10d %-15d", limits.get(i), limits.get(i+1) - 1));

            System.out.println(frequencies.get(i));

        }

    }

}

