Lab Report – Exception Handling: Index Out of Bounds

John Marshall
CSCI112 Spring 2023

Assignment Analysis and Design

In your own words, describe the problem including input and output. Briefly describe how you developed your code. Briefly describe what your code does and how it works – including anything different or unique or special that you did in your software. If the software is long or complicated, describe how it is organized.
Include a copy of any pseudocode or other design documents you used. If you worked with anyone else, asked anyone for help, or looked anything up, then mention it here. Include proper references to source material.

The purpose of this assignment was to test the ways exceptions should be handled. Previously exceptions had been thrown instead of caught so this assignment was primarily about creating different cases where exceptions might occur and catching them. The first task was to create a method that would create an array of random integers. After the next task was to write the array to a file created in the project file while catching any exceptions that might be thrown. Finally the last task was to write a method that would prompt users to enter the index of an array element then print that element to the console. This last method must have a valid exit point and catch any exceptions. All exceptions must be thrown back to the main method and then caught there.

Pseudocode:

```
  main {
        int[] randomIntArray = createRandomIntArray(100); //method to be defined below
        Scanner Instance

        //exceptions handled by main

     try {
        writeArrayToFile(array and file name )  //method to be defined below}
 catch (IOException) {
       "Error, connection with target location might have failed. Please try again. "
         } catch (Exception) { print exception error message}

     while (!condition) {
        try {
           printArrayIndexValue(randomIntArray) //METHOD to be defined below
           condition= true }
```

```
catch (ArrayIndexOutOfBoundsException) {
            Print ("The Index requested is out of bounds, press Enter to try again or Exit to quit.")
            Get  INPUT
            if (INPUT == "exit"or "EXIT") {
                Break loop }
          }//end Catch


catch (Exception) {
            Print ("Invalid Input, press ENTER to try again, type EXIT to quit")
            Get  INPUT
            if (INPUT == "exit"or "EXIT") {
                Break loop}
          }// End Catch



         }//loop

  }//end Main


  Method INT[] RandomIntArray(int arrayLength) {
      Int[] randomArray = new int[arrayLength];

      //populate random integer array loop (from 0-10
      for(i=0; i<arrayLength; i ++){
         randomArray[i] = (int) (Math.random()*10)
      }
      return randomArray
  }

METHOD writeArrayToFile( int[] intArray, String fileName) THROWS EXCEPTIONS{

  File  =  new File( fileName + ".txt")
          Writer x = new printwriter(y)

         x.print(intArray)
         x.close()      }// end method



Method  printArrayIndexValue(int[] randomArray) THROWS EXCEPTIONS{
         System.out.println("Enter the index of the value you would like to know? enter an integer
between 0 " +
            "and " + (randomArray.length -1));

         Scanner y = new Scanner
            int UserInput
            Print("The value of index " + UserInput + " is " + randomArray[UserInput])

  } // end createRandomInArray
```

Assignment Code

Include the code for your assignment unless otherwise directed by the assignment or by your instructor, which will be a copy of your Python project submitted with the report. You can put the report and the Python project all in one submission. In the report, either tell the reader that it is attached file or include the code.

```
/* John Marshall Week 3 Lab:  Exception Handling
CSCI112
Last Edited 2/13/2023
*/


import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;

public class Main {
  public static void main(String[] args) {
     int[] randomIntArray = createRandomIntArray(100);

     Scanner userIO = new Scanner(System.in);

     // Write array to file
     try {
        writeArrayToFile(randomIntArray, "Test");
```

```
      //specific Catch for IO exceptions
    } catch (IOException e) {
      e.printStackTrace();
      System.err.println("FileNotFoundException: " + e.getMessage());

      // catch any exceptions
    } catch (Exception e) {
      System.err.println("Error, connection with target location might have failed. Please try again.");
      e.printStackTrace();
    }//end catch



    //loop exit condition
    boolean validInput = false;


    //loop through to return from exceptions that might be thrown
    while (!validInput) {
      try {

        //call method that prompts user and prints element Value.
        printArrayIndexValue(randomIntArray);
        validInput = true;


        //Catch for out of bound index with exit Option
      } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("The Index requested is out of bounds, press Enter to try again or Exit to quit.");

        String termRequest = userIO.nextLine();
        if (termRequest.equals("exit") || termRequest.equals("EXIT")) {
          break;
        }

        //Catch for everything else with exit Option
      } catch (Exception e) {
        System.out.println("Invalid Input, press ENTER to try again, type EXIT to quit");
        String termRequest = userIO.nextLine();
        if (termRequest.equals("exit") || termRequest.equals("EXIT")) {
          break;
        }
```

```
            }


        } // end loop
    }//end Main method

    //create random array method, rakes length of the array as argument
    public static int[] createRandomIntArray(int arrayLength) {
        int i;
        int[] randomArray = new int[arrayLength];

        //populate random integer array loop (from 0-10
        for(i=0; i<arrayLength; i ++){
            randomArray[i] = (int) (Math.random()*10);
        }

        return randomArray;
    }// end createRandomInArray


    //write array to file method, takes array and file name as argument. Exceptions handled in
Main
    public static void writeArrayToFile(int[] intArray, String fileName) throws Exception{


            //passing Name argument as file name
            java.io.File y =  new java.io.File(fileName + ".txt");

            //start stream
            java.io.PrintWriter x = new java.io.PrintWriter(y);


            // print array in file and close stream
            x.print(Arrays.toString(intArray));
            x.close();
        }// end printArrayIndexValue




// Print Array index value prompts user and displays array value. Exceptions handled in Main
    public static void printArrayIndexValue(int[] randomArray) throws Exception{

            System.out.println("Enter the index of the value you would like to know? enter an integer
between 0 " +
            "and " + (randomArray.length -1));

         Scanner y = new Scanner(System.in);
            int UserInput = Integer.parseInt(y.nextLine());
            System.out.println("The value of index " + UserInput + " is " +
randomArray[UserInput]+".");

  } // end createRandomInArray



}//End main Class
```

Assignment Testing

Describe how you tested this program to verify that it runs correctly. Assignment Evaluation
Briefly describe what you learned from this project. What things did you struggle with? What was easy? Give your opinions of the process, including what you liked about the project and any suggestions you have for improving the project.

All testing should be done in the main method . All exceptions are thrown and handled in the main method thus this is where testing will be taken place

writeArrayToFile(int[] intArray, String fileName) - testing should be performed by passing the correct arguments and checking the output file. Exception Catch would need to be tested on a remote server or using some other filing system that is outside the scope of this project. Returns Void. Passing if the correct number of elements are printed in the correct range of values.

createRandomIntArray[int] - Test simply by printing the values of a created array to the Console. All arguments are hard coded there for all possible exceptions will be checked Exceptions. Expected return: integer array.

printArrayIndexValue(int[] randomArray) - 1.) test by passing A correct integer when the method is being called and run through as normal. Passes if the correct element value is displayed for imputed index 2.) Test by inputting an out of bond index. Passes if the catch block has been called and the user is prompted to try again or exit. Perform the test twice picking both options in catch. 3.) Pass an invalid input during the function. If the catch block is triggered and the user is prompted to try again or exit, it is a pass. Perform the test twice picking both options in catch.

Conclusion:

After finishing this project I felt that it was a bit of a challenge but in a way that made me think through the process of throwing and catching. I had originally missed the last requirement about the main handling of the exceptions and I had handled them locally in the beginning. While it did create more work for myself, it also taught me that I think more in a local scope when coding. I like to encapsulate as much as I can when I can. I know that this will become more difficult when it comes to more complex projects but atleast im aware of my bias.