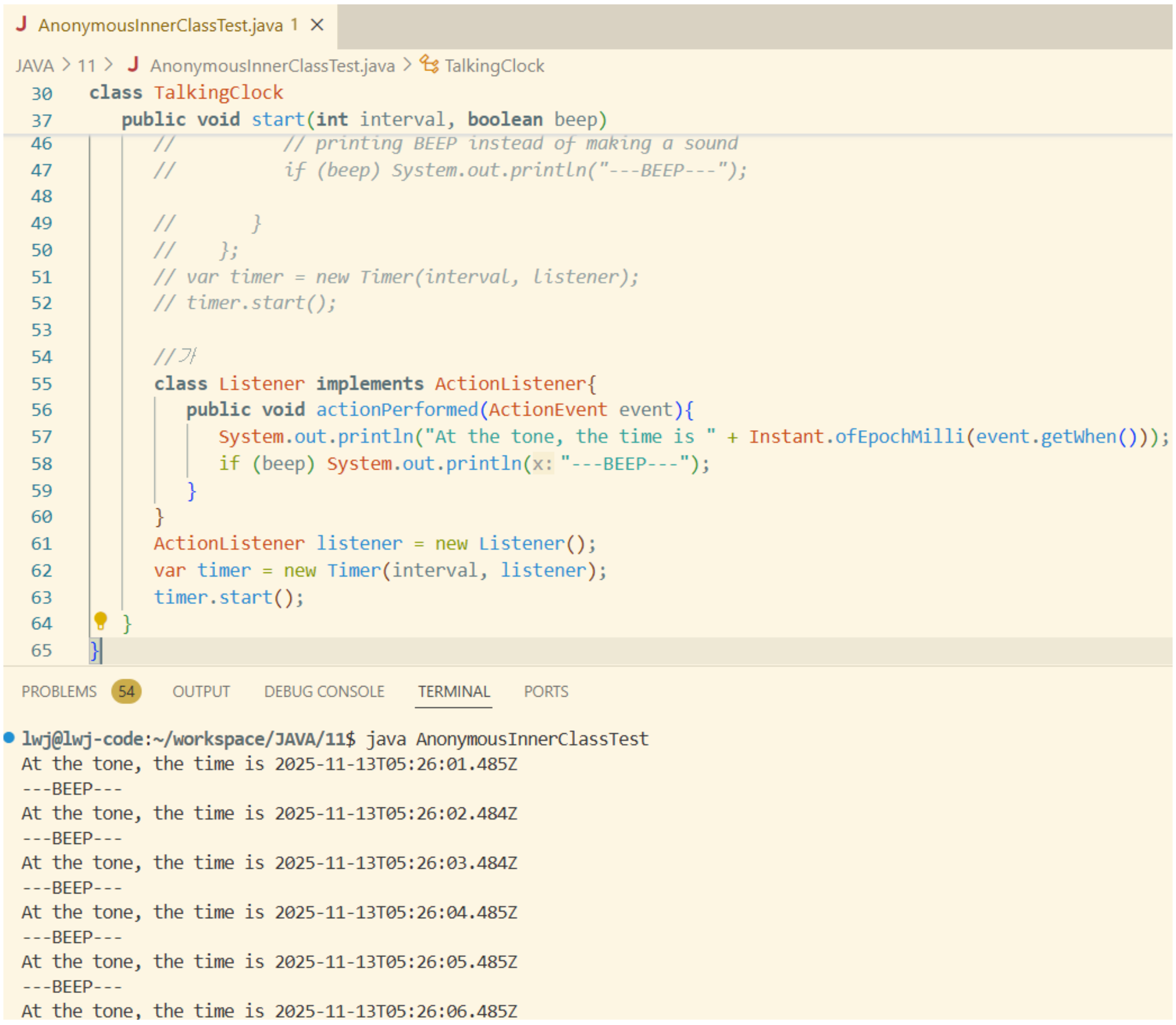
11주차 실습 보고서

강의명	객체지향프로그래밍및실습
담당교수	류기열
학과 학년	소프트웨어학과 2학년
학번	202220209
작성자	이욱준

2025.11.13.

☐ 1

(가) local inner class



AnonymousInnerClassTest.java 1

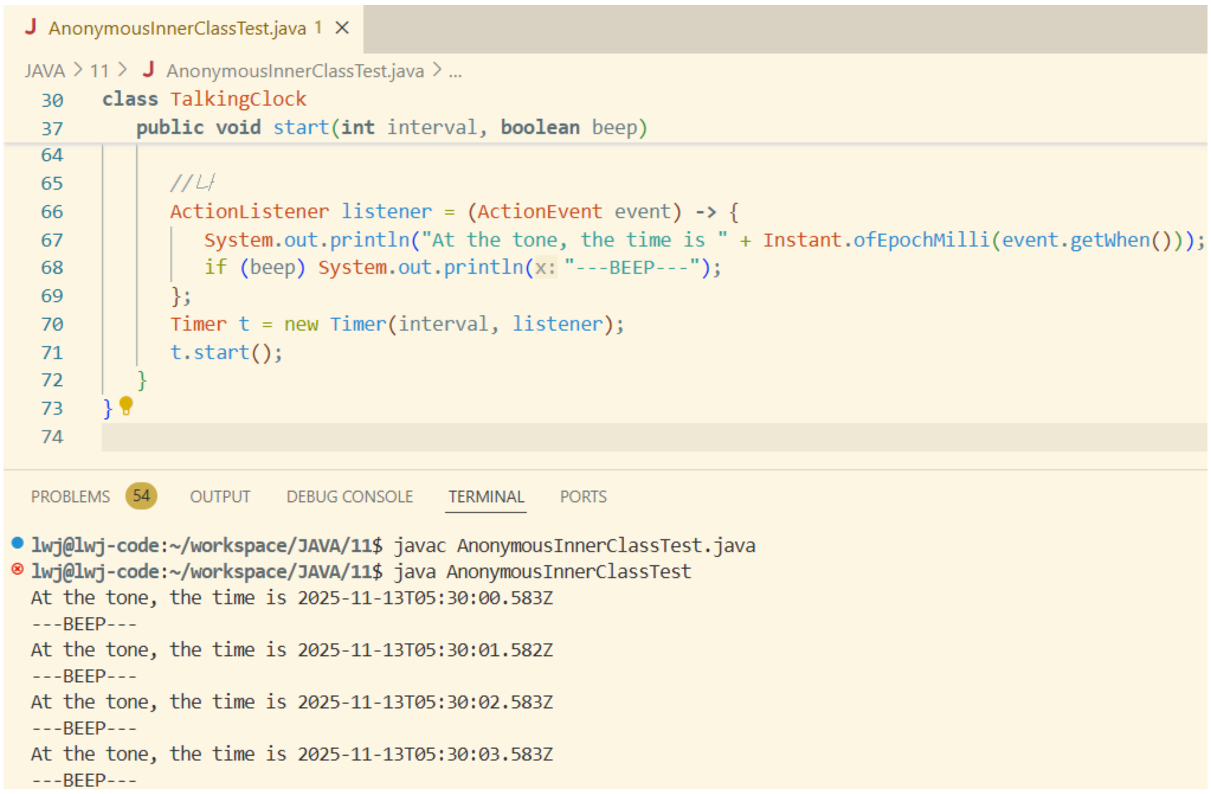
JAVA > 11 > AnonymousInnerClassTest.java > TalkingClock

```java
class TalkingClock
    public void start(int interval, boolean beep)
        //          // printing BEEP instead of making a sound
        //          if (beep) System.out.println("---BEEP---");

        //      }
        //    };
        // var timer = new Timer(interval, listener);
        // timer.start();

        //가
        class Listener implements ActionListener{
            public void actionPerformed(ActionEvent event){
                System.out.println("At the tone, the time is " + Instant.ofEpochMilli(event.getWhen()));
                if (beep) System.out.println(x: "---BEEP---");
            }
        }
        ActionListener listener = new Listener();
        var timer = new Timer(interval, listener);
        timer.start();
    }
}
```

PROBLEMS 54 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/11$ java AnonymousInnerClassTest
At the tone, the time is 2025-11-13T05:26:01.485Z
---BEEP---
At the tone, the time is 2025-11-13T05:26:02.484Z
---BEEP---
At the tone, the time is 2025-11-13T05:26:03.484Z
---BEEP---
At the tone, the time is 2025-11-13T05:26:04.485Z
---BEEP---
At the tone, the time is 2025-11-13T05:26:05.485Z
---BEEP---
At the tone, the time is 2025-11-13T05:26:06.485Z
```

(나) lambda exp



AnonymousInnerClassTest.java 1 ×

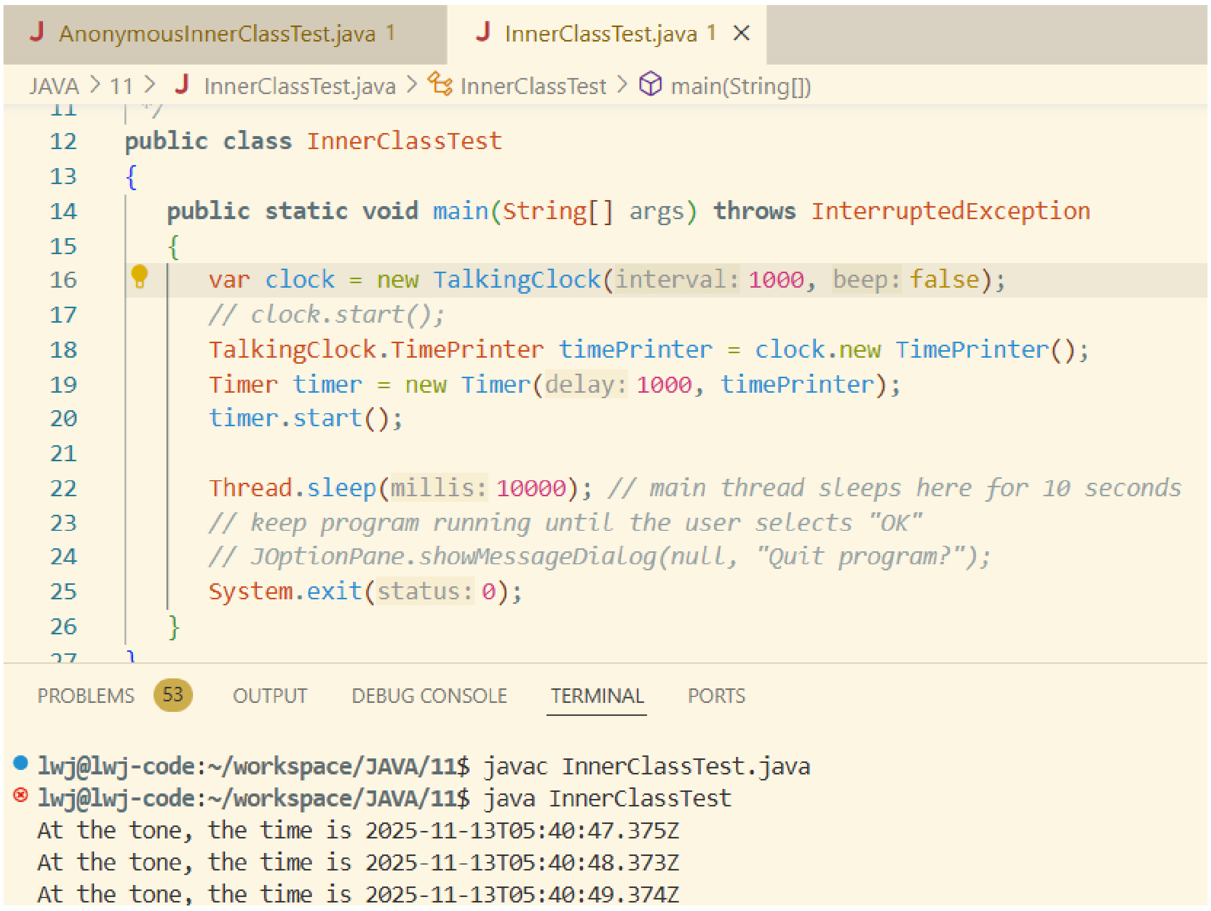
JAVA > 11 > AnonymousInnerClassTest.java > ...

```java
class TalkingClock
    public void start(int interval, boolean beep)

        //나
        ActionListener listener = (ActionEvent event) -> {
            System.out.println("At the tone, the time is " + Instant.ofEpochMilli(event.getWhen()));
            if (beep) System.out.println(x: "---BEEP---");
        };
        Timer t = new Timer(interval, listener);
        t.start();
    }
}
```

PROBLEMS 54 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/11$ javac AnonymousInnerClassTest.java
lwj@lwj-code:~/workspace/JAVA/11$ java AnonymousInnerClassTest
At the tone, the time is 2025-11-13T05:30:00.583Z
---BEEP---
At the tone, the time is 2025-11-13T05:30:01.582Z
---BEEP---
At the tone, the time is 2025-11-13T05:30:02.583Z
---BEEP---
At the tone, the time is 2025-11-13T05:30:03.583Z
---BEEP---
```

□ 2

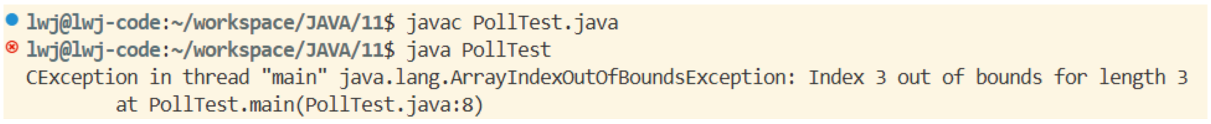

AnonymousInnerClassTest.java 1
InnerClassTest.java 1
JAVA > 11 > InnerClassTest.java > InnerClassTest > main(String[])
PROBLEMS 53
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```java
public class InnerClassTest
{
    public static void main(String[] args) throws InterruptedException
    {
        var clock = new TalkingClock(interval: 1000, beep: false);
        // clock.start();
        TalkingClock.TimePrinter timePrinter = clock.new TimePrinter();
        Timer timer = new Timer(delay: 1000, timePrinter);
        timer.start();

        Thread.sleep(millis: 10000); // main thread sleeps here for 10 seconds
        // keep program running until the user selects "OK"
        // JOptionPane.showMessageDialog(null, "Quit program?");
        System.exit(status: 0);
    }
```

```
lwj@lwj-code:~/workspace/JAVA/11$ javac InnerClassTest.java
lwj@lwj-code:~/workspace/JAVA/11$ java InnerClassTest
At the tone, the time is 2025-11-13T05:40:47.375Z
At the tone, the time is 2025-11-13T05:40:48.373Z
At the tone, the time is 2025-11-13T05:40:49.374Z
```

☐ 3

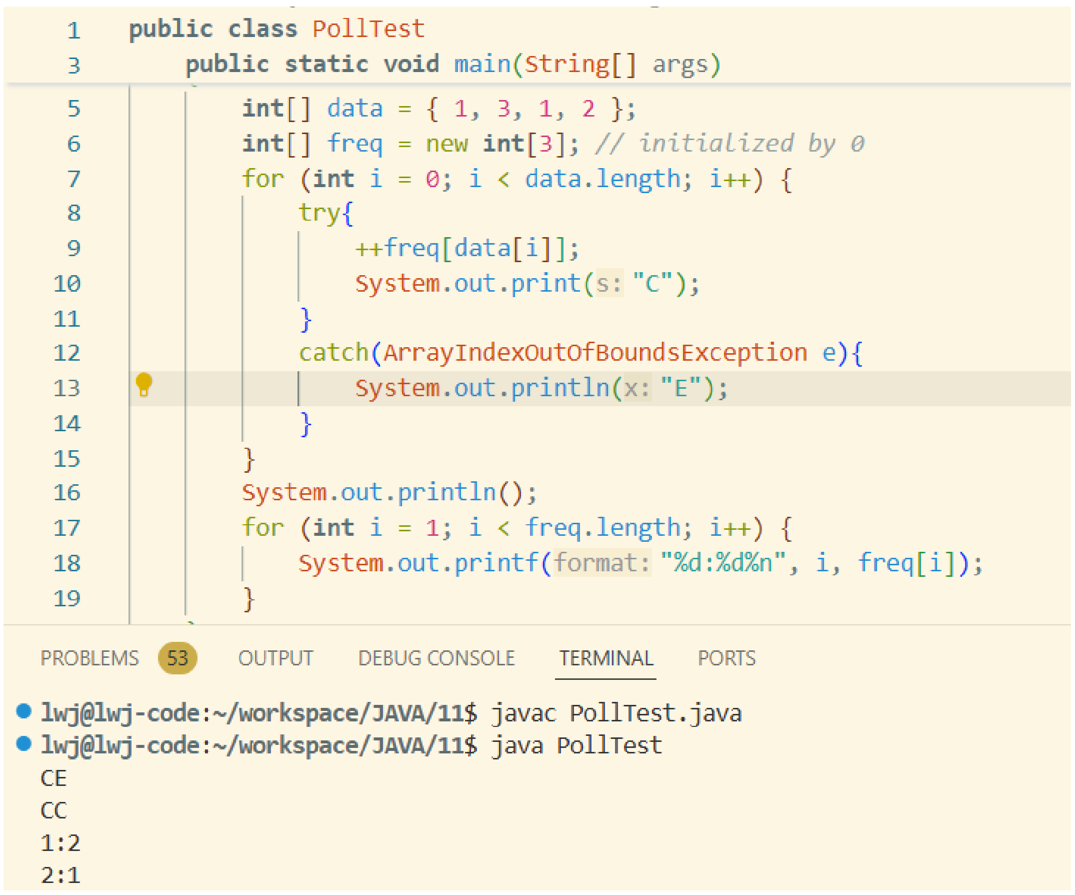
(가)



```
lwj@lwj-code:~/workspace/JAVA/11$ javac PollTest.java
lwj@lwj-code:~/workspace/JAVA/11$ java PollTest
CException in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
        at PollTest.main(PollTest.java:8)
```

Array의 size를 넘는 index에 대한 접근을 처리하는 Exception. int[] data의 length는 4인데, int[] freq[3]으로 선언된 것에서 freq[3]에 대해 접근을 할 수 없는데, 코드의 반복문 과정에서는 freq[3]을 접근하려고 하니 Exception이 발생함.

(나)



```java
public class PollTest
    public static void main(String[] args)
        int[] data = { 1, 3, 1, 2 };
        int[] freq = new int[3]; // initialized by 0
        for (int i = 0; i < data.length; i++) {
            try{
                ++freq[data[i]];
                System.out.print(s: "C");
            }
            catch(ArrayIndexOutOfBoundsException e){
                System.out.println(x: "E");
            }
        }
        System.out.println();
        for (int i = 1; i < freq.length; i++) {
            System.out.printf(format: "%d:%d%n", i, freq[i]);
        }
```

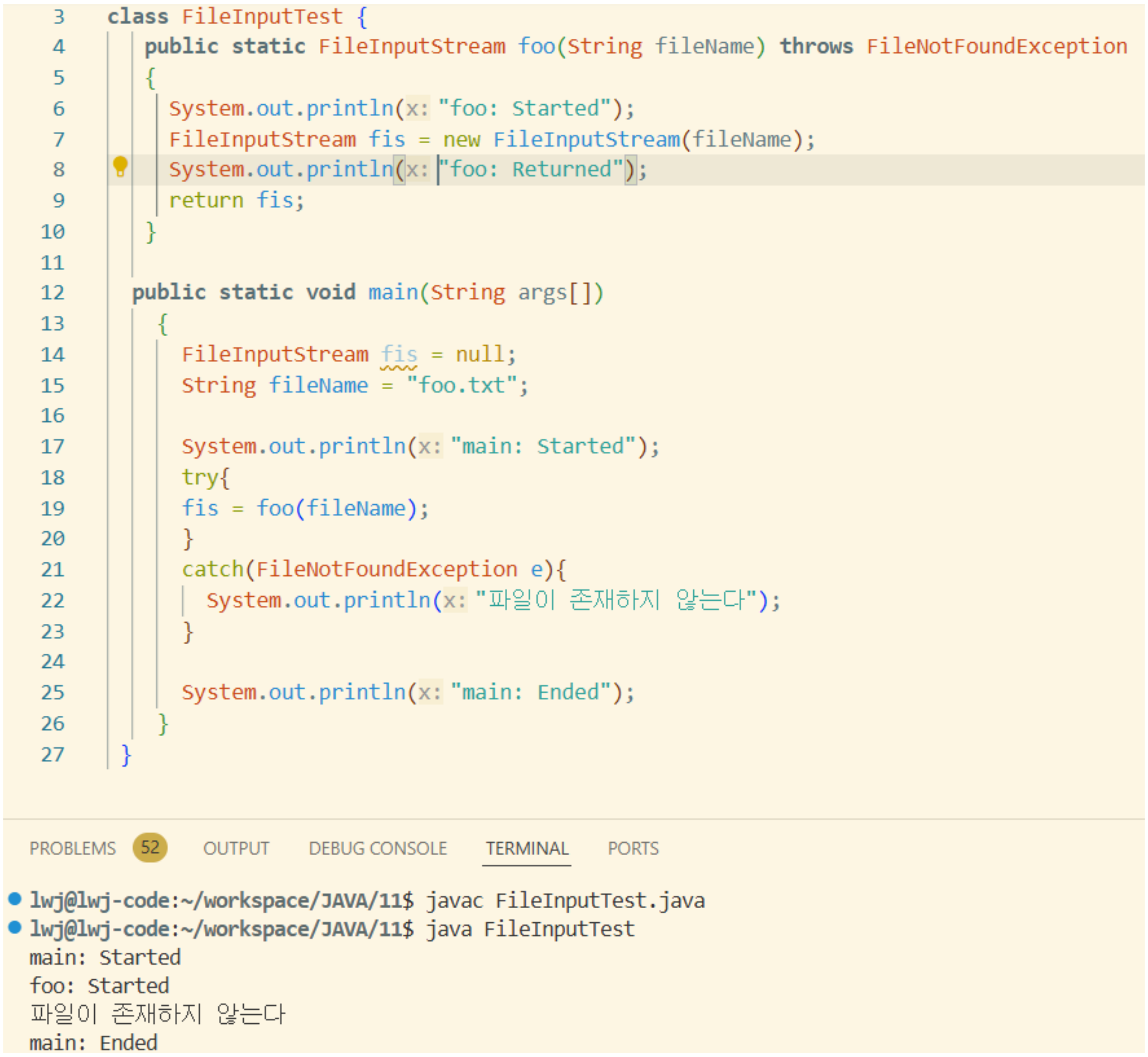
PROBLEMS 53   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
lwj@lwj-code:~/workspace/JAVA/11$ javac PollTest.java
lwj@lwj-code:~/workspace/JAVA/11$ java PollTest
CE
CC
1:2
2:1
```

반복문은 총 4회 반복된다. 1번 시도에서는 freq[1]의 값을 1증가, 2번 시도에서는 freq[3]의 값을 1증가하는데 이 때 freq[3]은 접근할 수 없는 영역이므로 Exception이 발생하기에 try문에서 catch문으로 이동하여 sout(“E”); 3번 시도에서는 freq[1]의 값을 1증가, 3번 시도에서는 freq[2]의 값을 1증가, 첫 번째 for문을 통해 freq[1] = 2, freq[2]=1이 되어 2번째 반복문에서 1:2, 2:1이 출력됨.

☐ 4

(가)

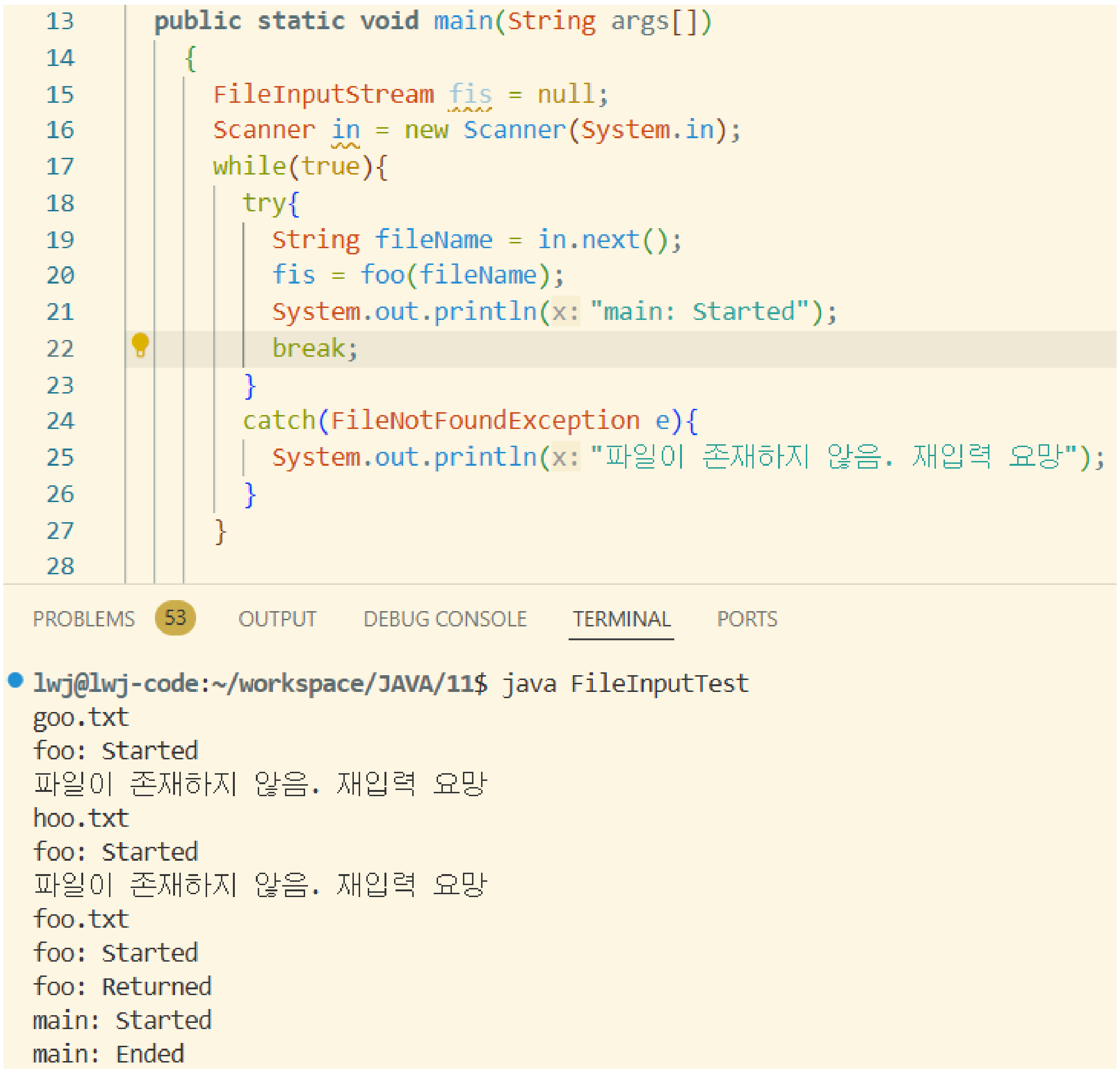


```
class FileInputTest {
    public static FileInputStream foo(String fileName) throws FileNotFoundException
    {
        System.out.println(x: "foo: Started");
        FileInputStream fis = new FileInputStream(fileName);
        System.out.println(x: "foo: Returned");
        return fis;
    }

  public static void main(String args[])
    {
      FileInputStream fis = null;
      String fileName = "foo.txt";

      System.out.println(x: "main: Started");
      try{
      fis = foo(fileName);
      }
      catch(FileNotFoundException e){
        System.out.println(x: "파일이 존재하지 않는다");
      }

      System.out.println(x: "main: Ended");
    }
}
```

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/11$ javac FileInputTest.java
lwj@lwj-code:~/workspace/JAVA/11$ java FileInputTest
main: Started
foo: Started
파일이 존재하지 않는다
main: Ended
```

(다)



```java
public static void main(String args[])
{
    FileInputStream fis = null;
    Scanner in = new Scanner(System.in);
    while(true){
        try{
            String fileName = in.next();
            fis = foo(fileName);
            System.out.println(x: "main: Started");
            break;
        }
        catch(FileNotFoundException e){
            System.out.println(x: "파일이 존재하지 않음. 재입력 요망");
        }
    }

```

PROBLEMS 53 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/11$ java FileInputTest
goo.txt
foo: Started
파일이 존재하지 않음. 재입력 요망
hoo.txt
foo: Started
파일이 존재하지 않음. 재입력 요망
foo.txt
foo: Started
foo: Returned
main: Started
main: Ended
```

□ 코드 전문

AnonymousInnerClassTest.java

```java
import java.awt.*;
import java.awt.event.*;
import java.time.*;
import javax.swing.*;

/**
 * This program demonstrates anonymous inner classes.
 * @version 1.12 2017-12-14
 * @author Cay Horstmann
 */
public class AnonymousInnerClassTest
{
   public static void main(String[] args) throws InterruptedException
   {
      var clock = new TalkingClock();
      clock.start(1000, true);

       Thread.sleep(10000); // main thread sleeps here for 10 seconds

      // keep program running until the user selects "OK"
      // JOptionPane.showMessageDialog(null, "Quit program?");
      System.exit(0);
   }
}

/**
 * A clock that prints the time in regular intervals.
 */
class TalkingClock
{
   /**
    * Starts the clock.
    * @param interval the interval between messages (in milliseconds)
    * @param beep true if the clock should beep
    */
   public void start(int interval, boolean beep)
   {
      // var listener = new ActionListener()
      //    {
      //       public void actionPerformed(ActionEvent event)
      //       {
      //          System.out.println("At the tone, the time is "
      //             + Instant.ofEpochMilli(event.getWhen()));
      //         //  if (beep) Toolkit.getDefaultToolkit().beep();
      //         // printing BEEP instead of making a sound
      //         if (beep) System.out.println("---BEEP---");

      //       }
      //    };
      // var timer = new Timer(interval, listener);
      // timer.start();

      //가
      // class Listener implements ActionListener{
      //    public void actionPerformed(ActionEvent event){
      //       System.out.println("At the tone, the time is " +
Instant.ofEpochMilli(event.getWhen()));
      //       if (beep) System.out.println("---BEEP---");
      //    }
      // }
      // ActionListener listener = new Listener();
      // var timer = new Timer(interval, listener);
      // timer.start();

      //나
      ActionListener listener = (ActionEvent event) -> {
         System.out.println("At the tone, the time is " +
Instant.ofEpochMilli(event.getWhen()));
         if (beep) System.out.println("---BEEP---");
      };
      Timer t = new Timer(interval, listener);
      t.start();
   }
}
```

InnerClassTest.java

```
import java.awt.*;
import java.awt.event.*;
import java.time.*;
import javax.swing.*;

/**
 * This program demonstrates the use of inner classes.
 * @version 1.11 2017-12-14
 * @author Cay Horstmann
 */
public class InnerClassTest
{
```

```java
   public static void main(String[] args) throws InterruptedException
   {
      var clock = new TalkingClock(1000, false);
      // clock.start();
      TalkingClock.TimePrinter timePrinter = clock.new TimePrinter();
      Timer timer = new Timer(1000, timePrinter);
      timer.start();

      Thread.sleep(10000); // main thread sleeps here for 10 seconds
      // keep program running until the user selects "OK"
      // JOptionPane.showMessageDialog(null, "Quit program?");
      System.exit(0);
   }
}

/**
 * A clock that prints the time in regular intervals.
 */
class TalkingClock
{
   private int interval;
   private boolean beep;

   /**
    * Constructs a talking clock
    * @param interval the interval between messages (in milliseconds)
    * @param beep true if the clock should beep
    */
   public TalkingClock(int interval, boolean beep)
   {
      this.interval = interval;
      this.beep = beep;
   }

   /**
    * Starts the clock.
    */
   public void start()
   {
      var listener = new TimePrinter();
      var timer = new Timer(interval, listener);
      timer.start();
   }

   public class TimePrinter implements ActionListener
   {
      public void actionPerformed(ActionEvent event)
      {
        System.out.println("At the tone, the time is "
           + Instant.ofEpochMilli(event.getWhen()));
        //   if (beep) Toolkit.getDefaultToolkit().beep();
        // printing BEEP instead of making a sound
        if (beep) System.out.println("---BEEP---");
      }
   }
}
```

PollTest.java

```
public class PollTest
{
    public static void main(String[] args)
    {
        int[] data = { 1, 3, 1, 2 };
        int[] freq = new int[3]; // initialized by 0
        for (int i = 0; i < data.length; i++) {
            try{
                ++freq[data[i]];
                System.out.print("C");
            }
            catch(ArrayIndexOutOfBoundsException e){
                System.out.println("E");
            }
        }
        System.out.println();
        for (int i = 1; i < freq.length; i++) {
            System.out.printf("%d:%d%n", i, freq[i]);
        }
    }
}
```

FileInputTest.java

```
import java.io.*;
import java.util.Scanner;

class FileInputTest {
   public static FileInputStream foo(String fileName) throws
FileNotFoundException
   {
     System.out.println("foo: Started");
     FileInputStream fis = new FileInputStream(fileName);
```

```
    System.out.println("foo: Returned");
    return fis;
  }

 public static void main(String args[])
   {
     FileInputStream fis = null;
     Scanner in = new Scanner(System.in);
     while(true){
       try{
         String fileName = in.next();
         fis = foo(fileName);
         System.out.println("main: Started");
         break;
       }
       catch(FileNotFoundException e){
         System.out.println("파일이 존재하지 않음. 재입력 요망");
       }
     }

     System.out.println("main: Ended");
   }
}
```