Test 6

SET B Debugging(26-07-2024)

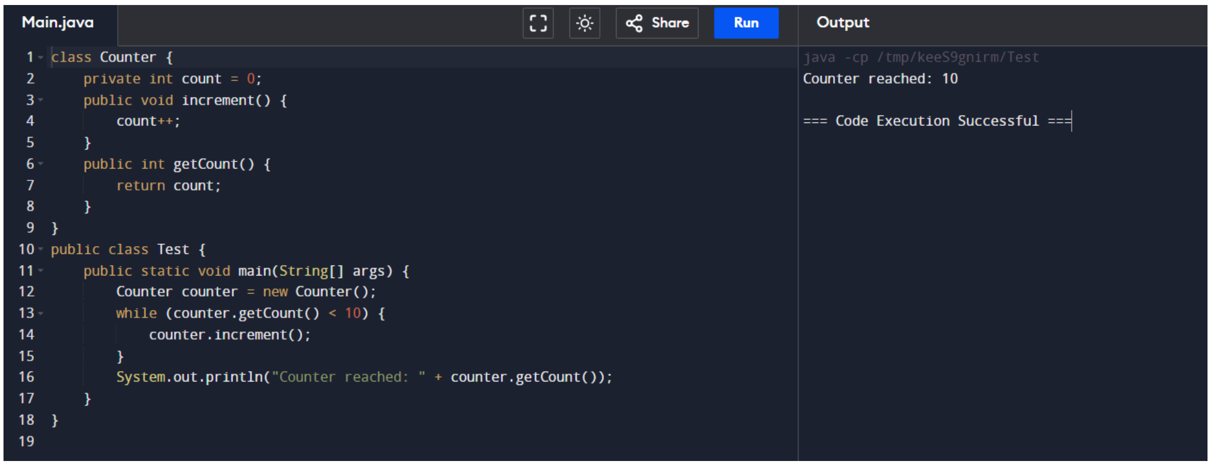
```
1. public class Counter {
private int count = 0;
public void increment() {
count++;
}
public int getCount() {
return count;
}
}
public class Test {
public static void main(String[] args) {
Counter counter = new Counter();

while (counter.getCount() &lt; 10) {
counter.increment();
}
System.out.println(&quot;Counter reached: &quot; +
counter.getCount());
}
```

}

▯ Issue: Static field not retaining value across instances.

▯ Solution: Check singleton implementation for proper instance handling.



```java
class Counter {
    private int count = 0;
    public void increment() {
        count++;
    }
    public int getCount() {
        return count;
    }
}
public class Test {
    public static void main(String[] args) {
        Counter counter = new Counter();
        while (counter.getCount() < 10) {
            counter.increment();
        }
        System.out.println("Counter reached: " + counter.getCount());
    }
}
```

```
java -cp /tmp/keeS9gnirm/Test
Counter reached: 10

=== Code Execution Successful ===
```

2. public class Employee {

private String name;

public Employee(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

public class Test {

public static void main(String[] args) {

Employee e = new Employee(&quot;John&quot;);

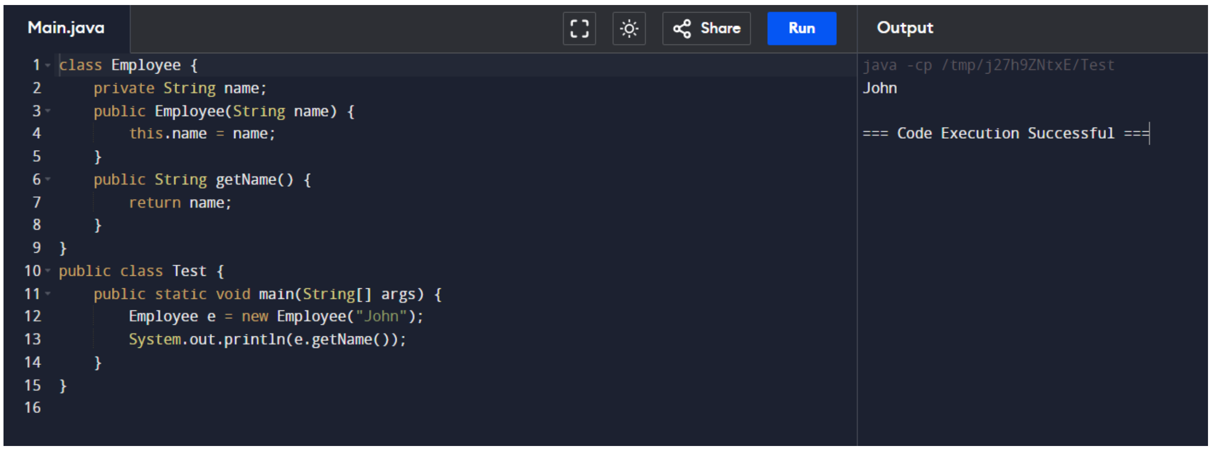
System.out.println(e.name); // Compilation error

}

}

⍰ Issue: Direct access to private field name.

⍰ Solution: Use getter method getName() to access private fields.



```java
class Employee {
    private String name;
    public Employee(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
}
public class Test {
    public static void main(String[] args) {
        Employee e = new Employee("John");
        System.out.println(e.getName());
    }
}
```

```
java -cp /tmp/j27h9ZNtxE/Test
John

=== Code Execution Successful ===
```

3. Question: Why is the FileNotFoundException not being caught when trying to open a file?

⍰ Potential Issue: Make sure the FileInputStream or FileReader is enclosed in a try-

catch block.

public class FileOpener {

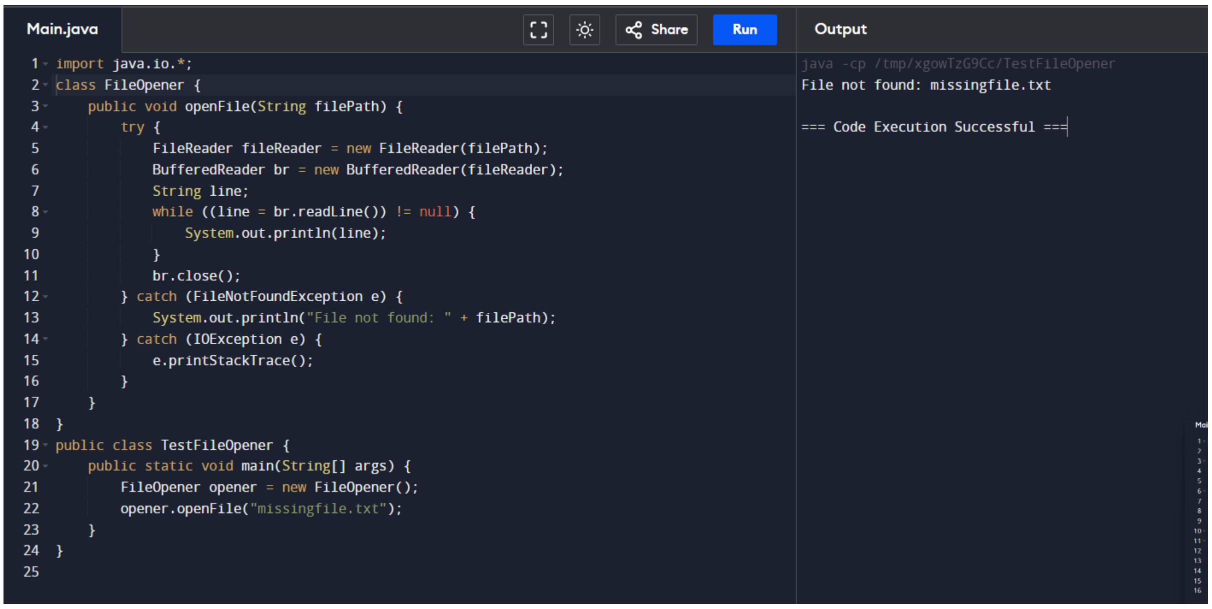
public void openFile(String filePath) {

try {

```
}

FileReader fileReader = new FileReader(filePath);
BufferedReader br = new BufferedReader(fileReader);
String line;
while ((line = br.readLine()) != null) {
System.out.println(line);
}
br.close();
} catch (FileNotFoundException e) {
System.out.println(&quot;File not found: &quot; + filePath);
} catch (IOException e) {
e.printStackTrace();
}
}
}

public class TestFileOpener {
public static void main(String[] args) {
FileOpener opener = new FileOpener();
opener.openFile(&quot;missingfile.txt&quot;);
}
}
```



```java
import java.io.*;
class FileOpener {
    public void openFile(String filePath) {
        try {
            FileReader fileReader = new FileReader(filePath);
            BufferedReader br = new BufferedReader(fileReader);
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
            br.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + filePath);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
public class TestFileOpener {
    public static void main(String[] args) {
        FileOpener opener = new FileOpener();
        opener.openFile("missingfile.txt");
    }
}
```

```
java -cp /tmp/xgowTzG9Cc/TestFileOpener
File not found: missingfile.txt

=== Code Execution Successful ===
```

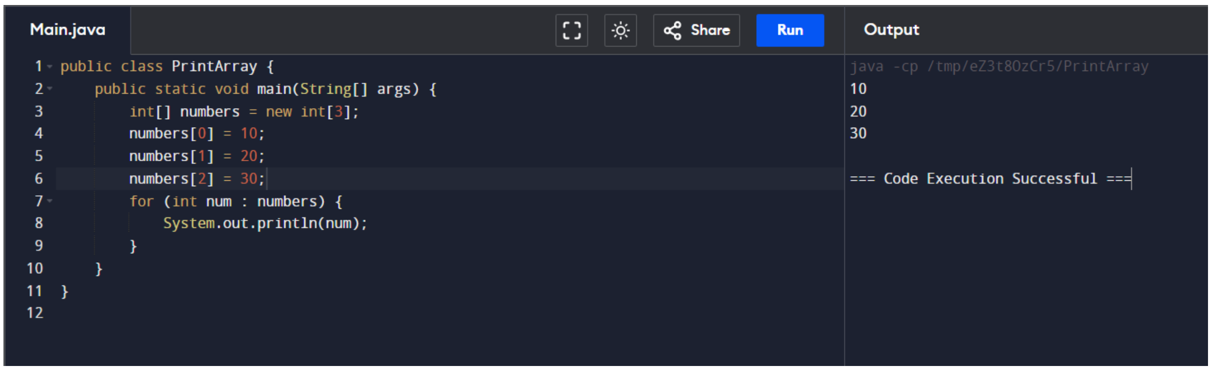
4. Question: Why is my array not printing the correct values?

⍰ Potential Issue: Ensure the array values are set correctly before printing.

```java
public class PrintArray {


public static void main(String[] args) {
int[] numbers = new int[3];
numbers[0] = 10;
numbers[1] = 20;
numbers[2] = 30;
for (int num : numbers) {
System.out.println(num);
}
}
```

}



```java
public class PrintArray {
    public static void main(String[] args) {
        int[] numbers = new int[3];
        numbers[0] = 10;
        numbers[1] = 20;
        numbers[2] = 30;
        for (int num : numbers) {
            System.out.println(num);
        }
    }
}
```

```
java -cp /tmp/eZ3t8OzCr5/PrintArray
10
20
30

=== Code Execution Successful ===
```