**TASK1 - What is a process?**

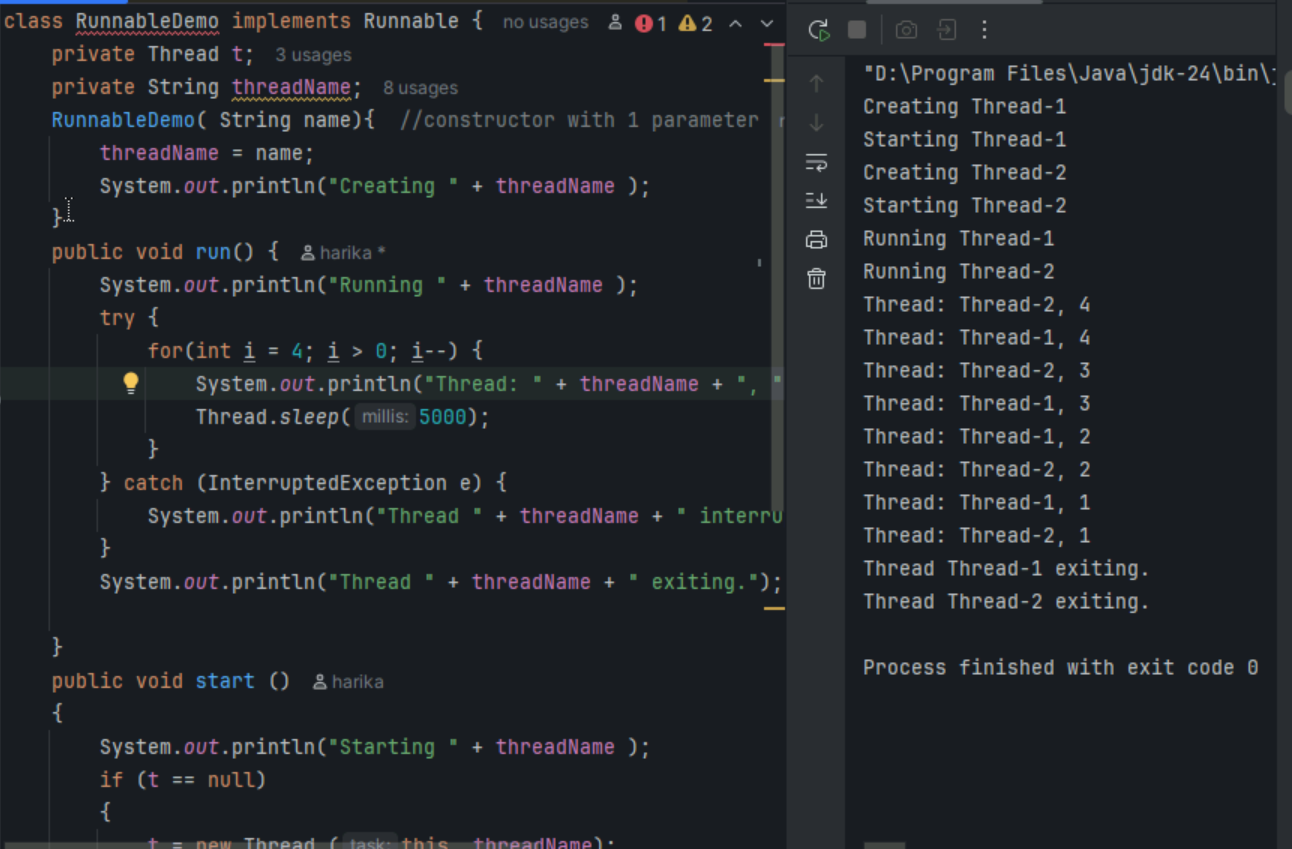
A **process** in Java is an external program or command started and managed from within a Java application.

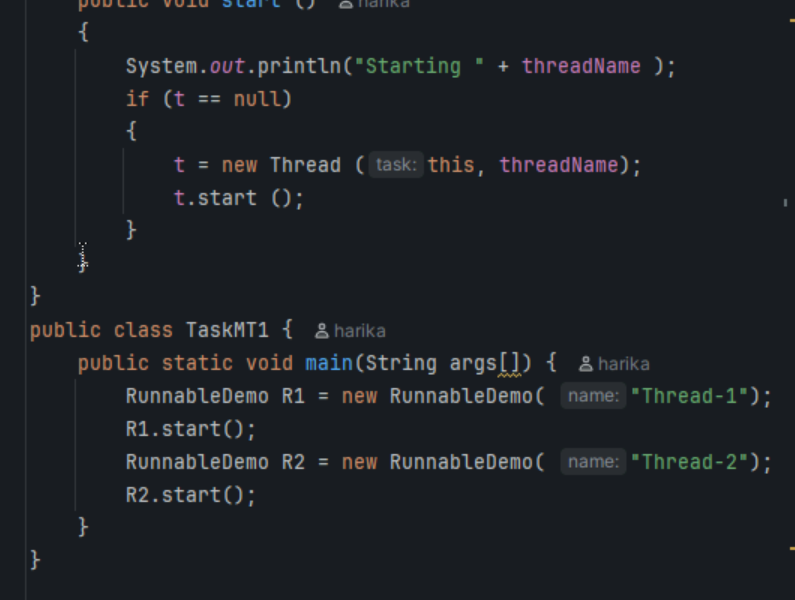
Java provides the Process and ProcessBuilder classes to run system-level commands or external programs.

**TASK2 - What is a thread?**

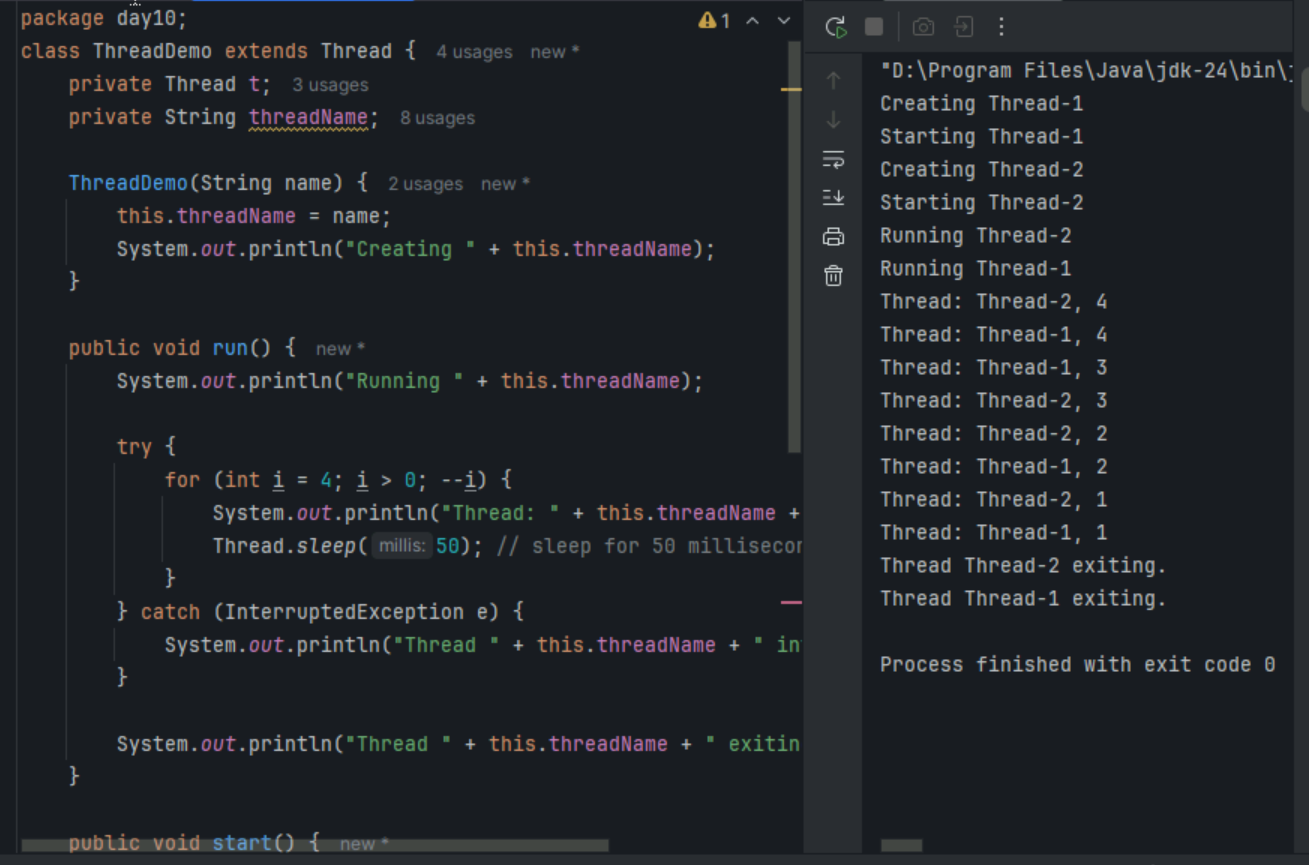
A thread is like a mini-task running inside a Java program (process). Multiple threads can run at the same time to do different things.

**Task3**

****

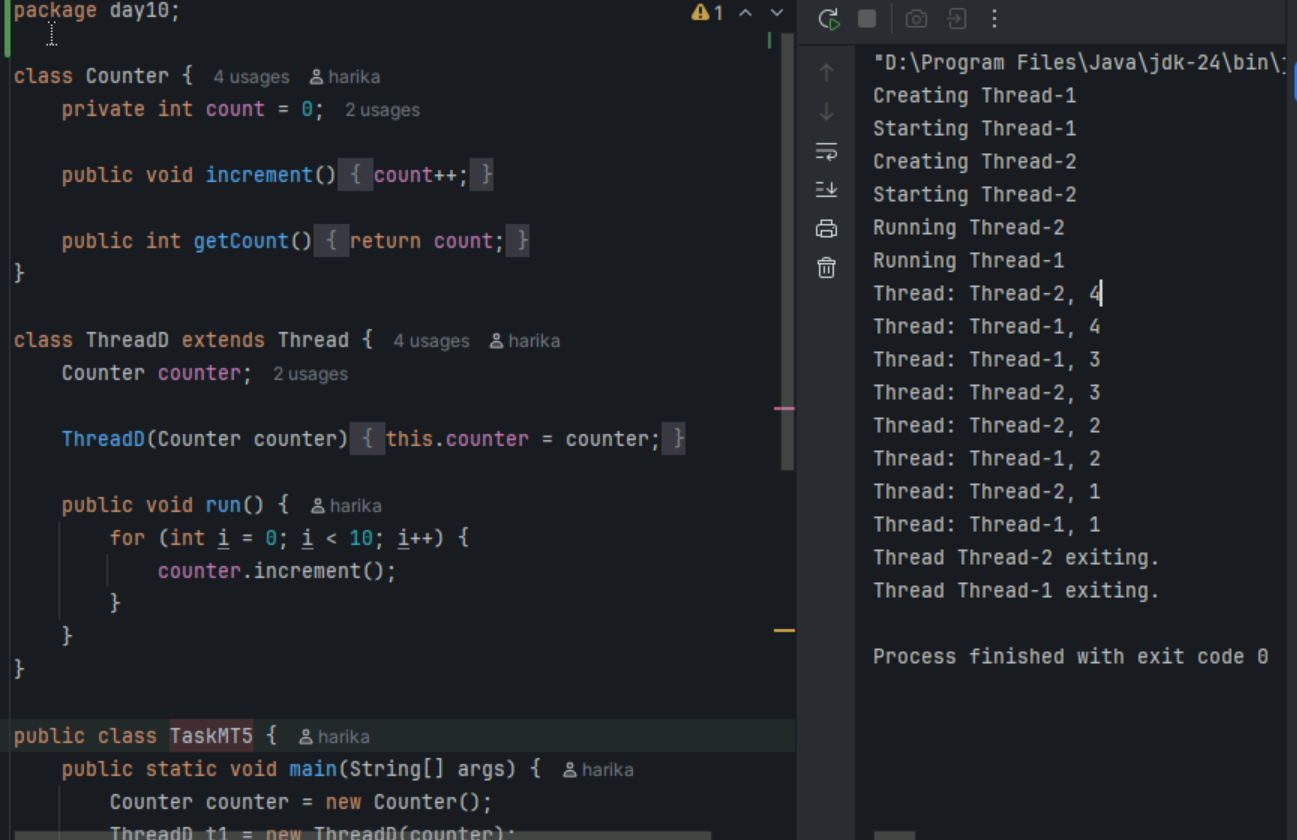
****

**Task4**

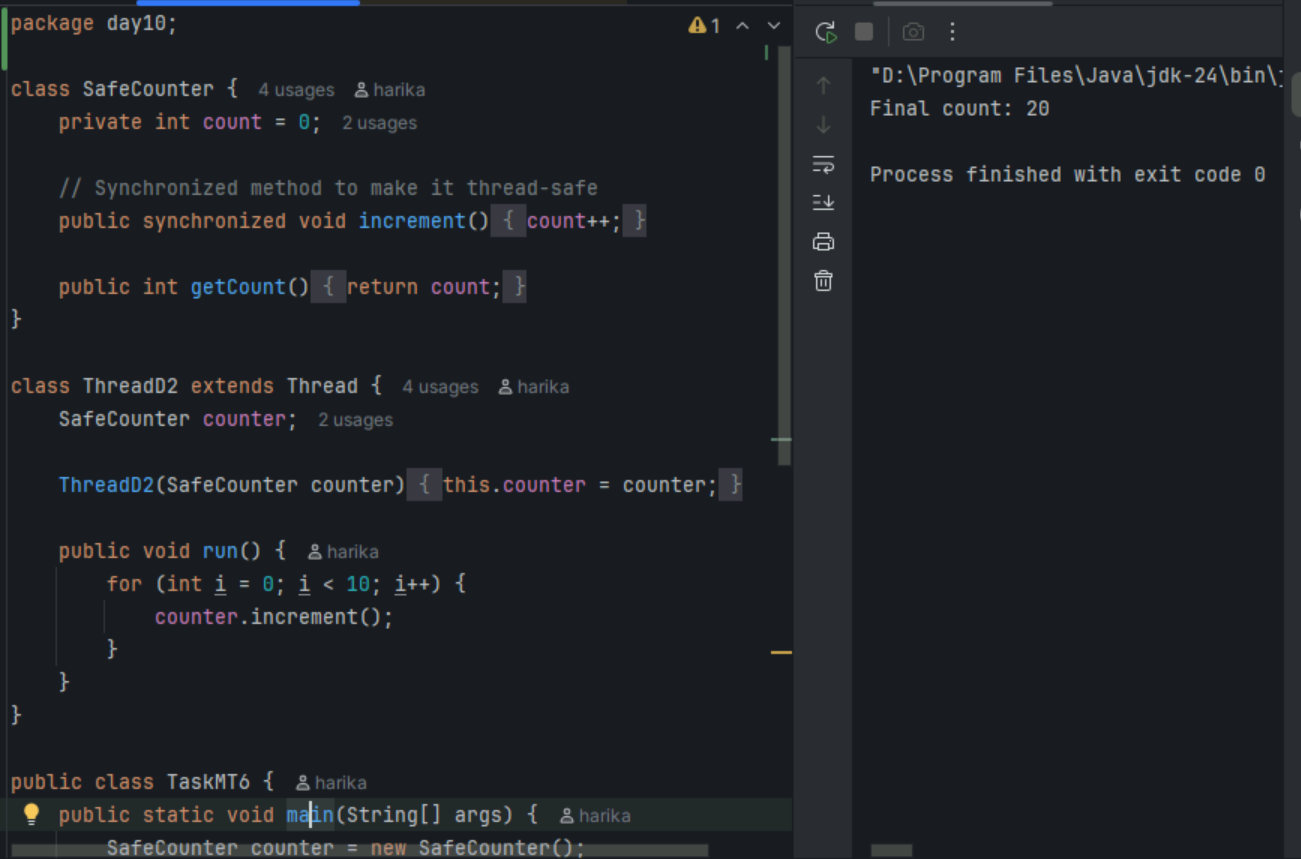
****

****

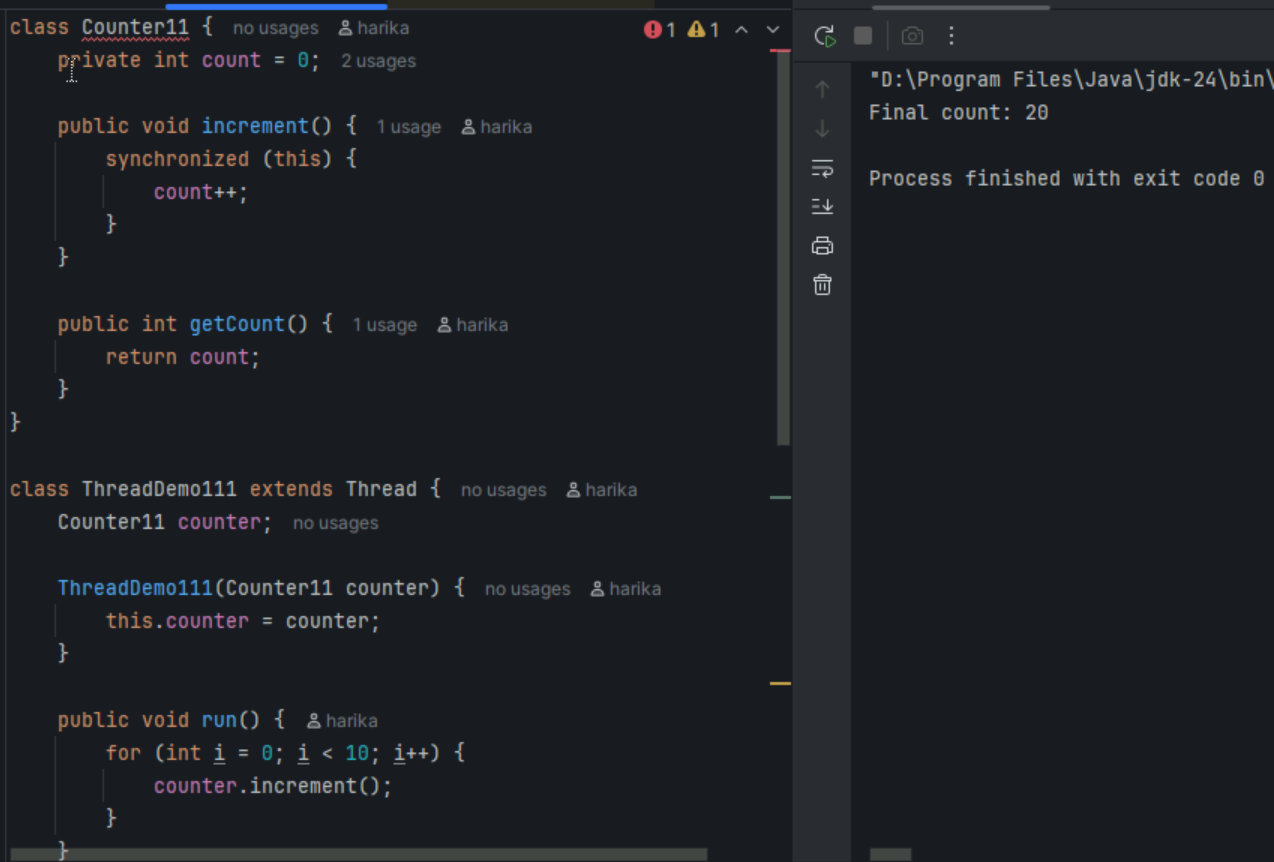
**Task5**

****

**Task6**

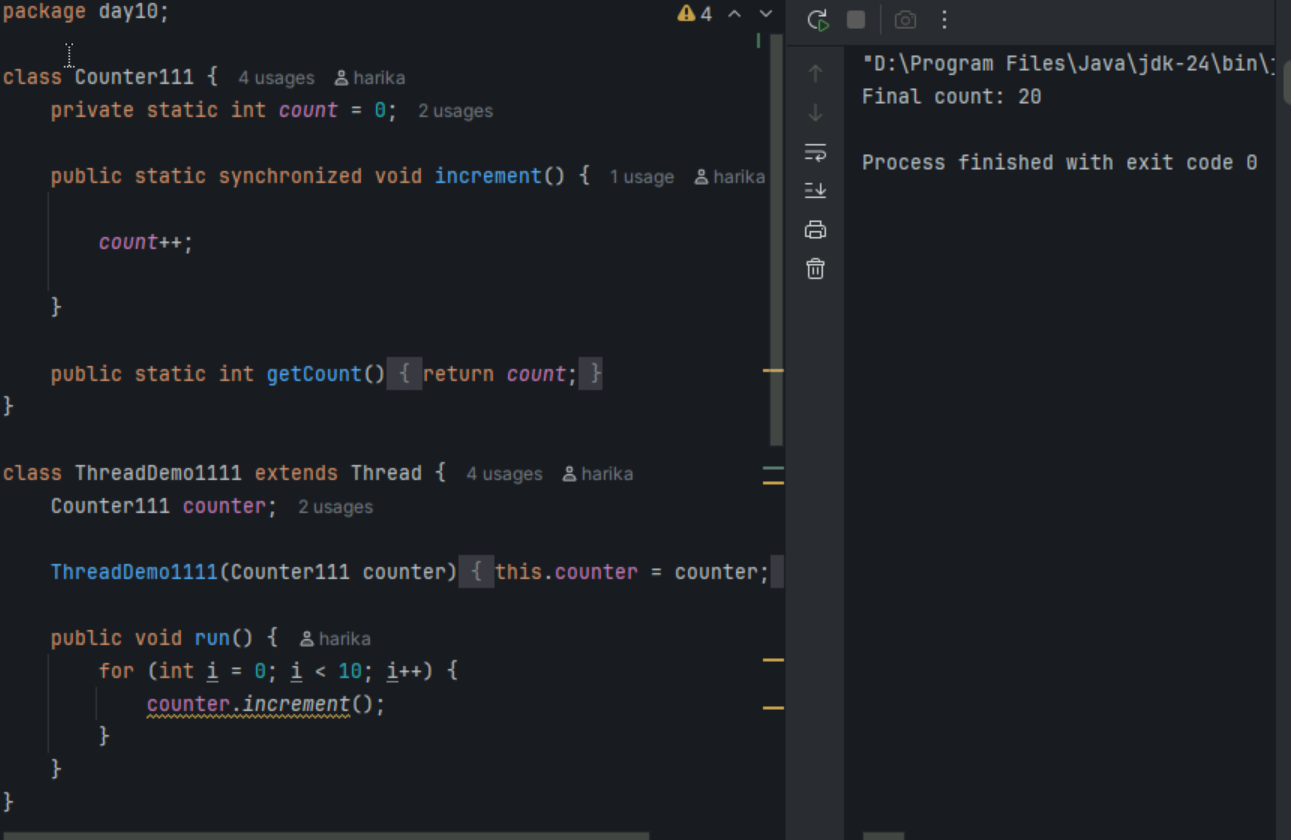
****

**Task7**

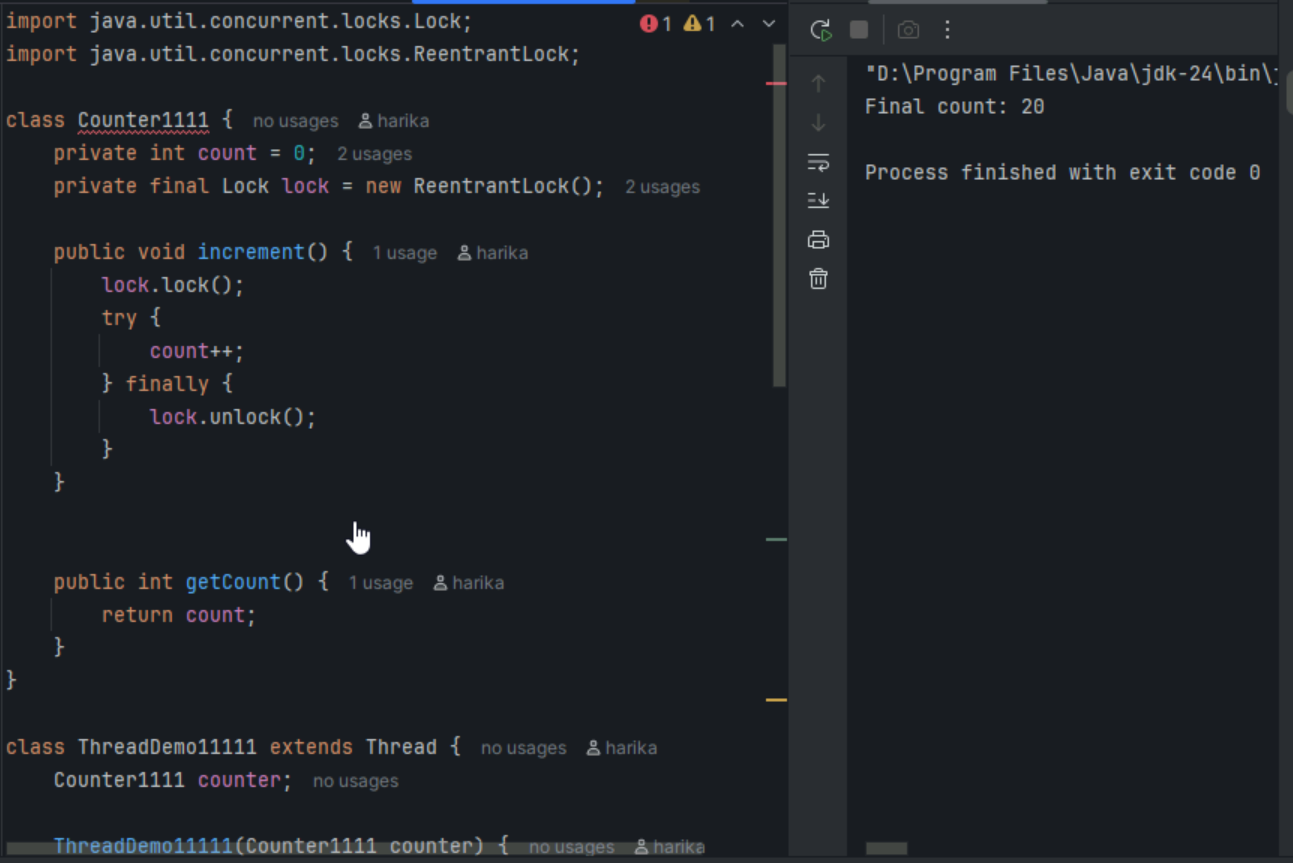
****

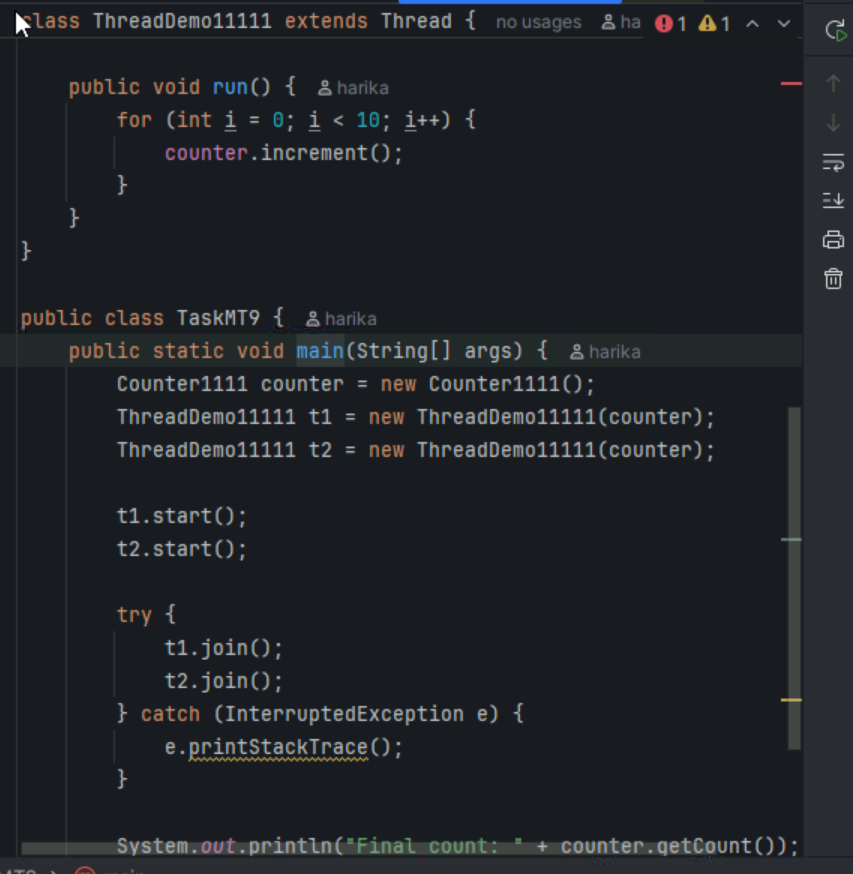
****

**Task8**

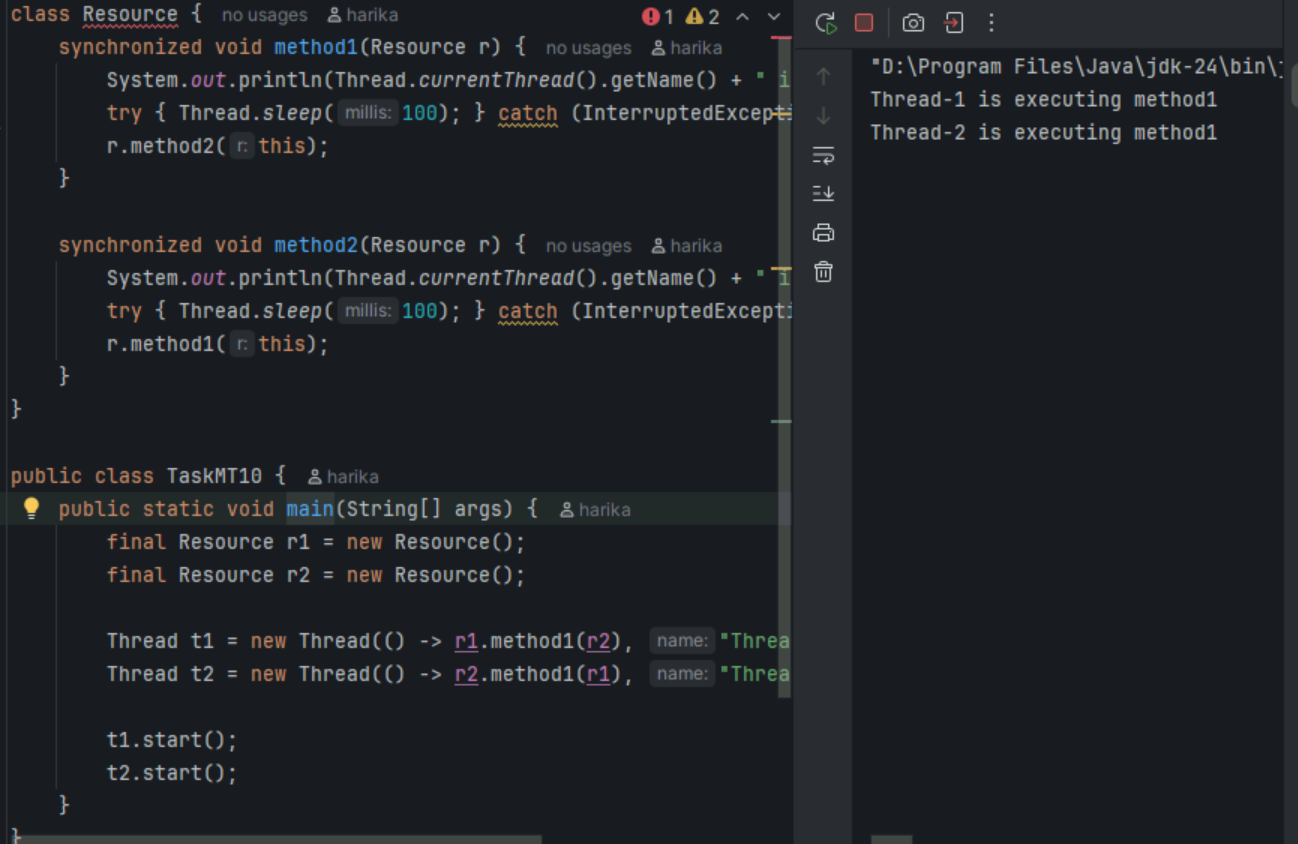
****

**Task9**

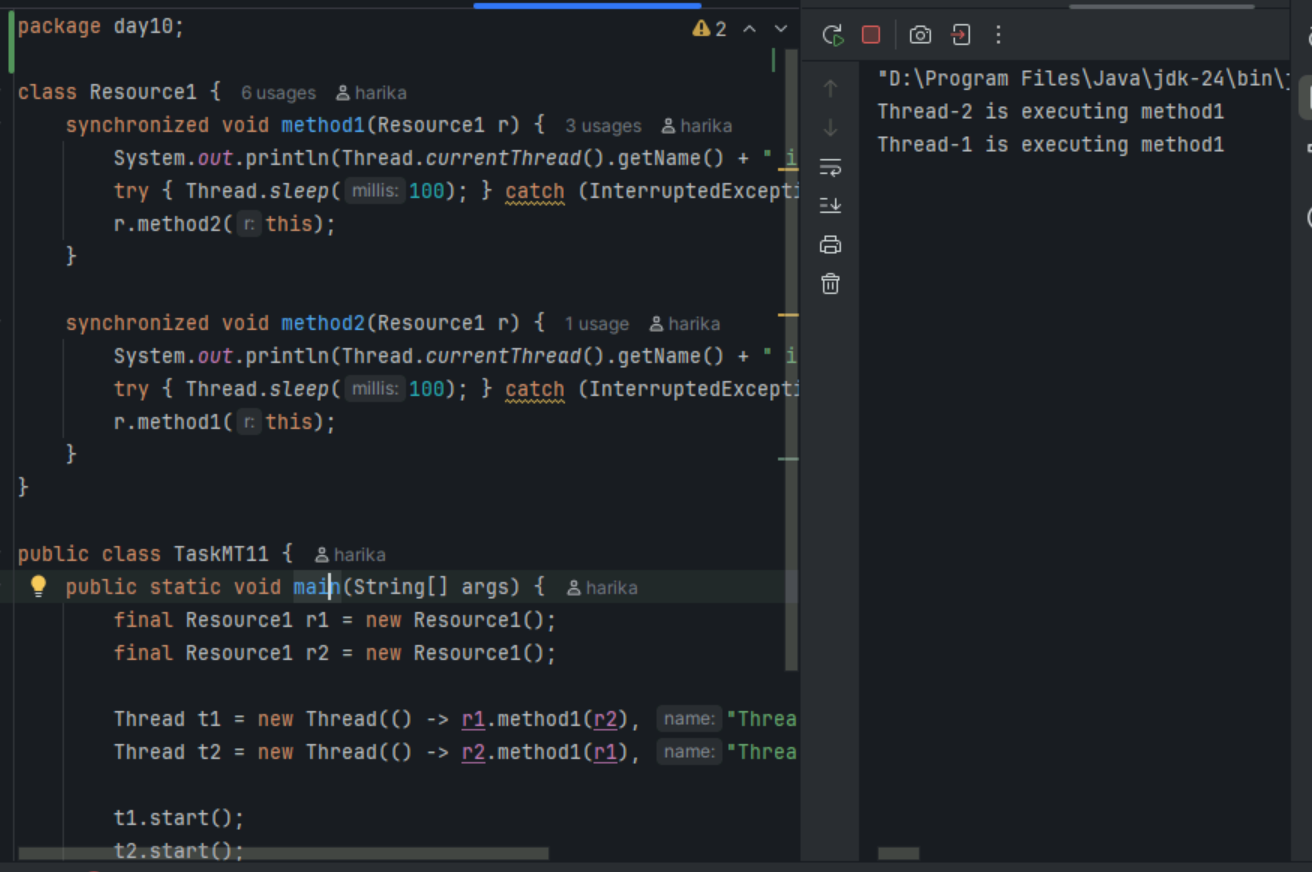
****

****

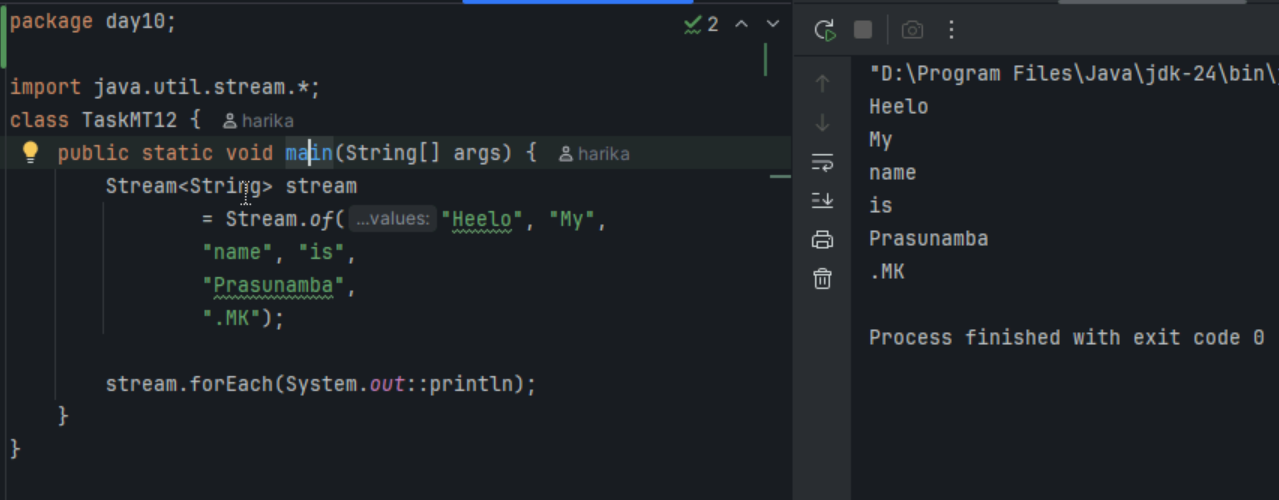
**Tak10**

****

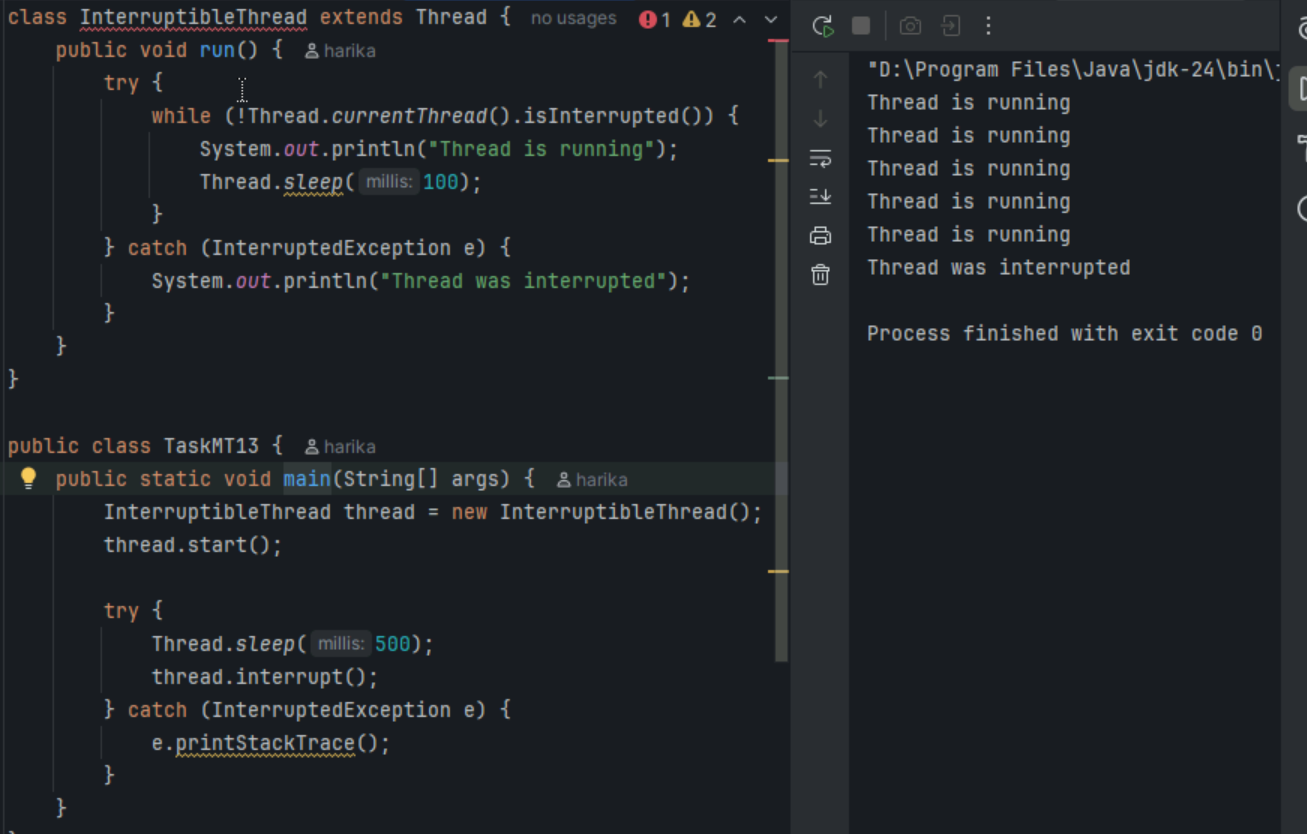
**Task11**

****

**Task12**

****

**Task13**

****

**Task14 - What are Daemon Threads in Java?**

A Daemon thread is a background thread in Java that runs in the background and provides services to user (non-daemon) threads.

🔸 Key Point:  
 When all user threads finish, the JVM automatically exits, and all daemon threads are terminated.

**Task15 - What are the debugging tools in Java.. list down a few..**

### **🔧 Java Debugging Tools:**

1. Eclipse Debugger – Built into Eclipse IDE, supports breakpoints, step execution, and variable watching.
2. IntelliJ IDEA Debugger – Powerful and user-friendly debugger with features like smart step-in and thread control.
3. jdb (Java Debugger) – Command-line tool provided with JDK for low-level debugging.
4. VisualVM – Monitors memory, CPU usage, threads, and garbage collection in real-time.
5. JConsole – GUI tool for monitoring Java applications using JMX; useful for performance and memory tracking.

**Task 15 - What are the debugging tools in Java.. list down a few..**

## **Error Messages in Java**

### **1. Compile-Time Errors**

* Happen before running the program.
* Caused by wrong syntax.

**Examples:**

* Missing ;
* Missing { or }
* Typing mistakes

**Example:**

int a = 10 // ❌ missing semicolon

**Error: ';' expected**

### **2. Run-Time Errors (Exceptions)**

* Happen when the program is running.
* Code compiles fine but crashes while running.

**Examples:**

| Error Name | When It Happens |
| --- | --- |
| NullPointerException | Using a variable that is null |
| ArrayIndexOutOfBounds | Using wrong index in an array |
| IOException | Problem while reading/writing files |
| StackOverflowError | Infinite method calls (like recursion) |

### **Example:**

String name = null;

System.out.println(name.length()); // ❌ Error

Output: NullPointerException

### **Summary:**

* Compile-time error = grammar mistake in code
* Run-time error = error while program is running

**Task16** - What is Stack trace.. What will it do? Understand the below points.. Identify the error

Locate the code Analyze the code Solution also

A **stack trace** is a **detailed error message** shown when a program crashes at **runtime**.  
 It helps you **find and fix** the problem in your code.

### **What does a Stack Trace do?**

It helps you:

1. **Identify the Error** –  
    It shows **what type of error** happened (e.g., NullPointerException, ArrayIndexOutOfBoundsException).
2. **Locate the Code** –  
    It shows the **exact line number** and the **class/method** where the error occurred.
3. **Analyze the Code** –  
    You can look at the code on that line to see **why** it failed.
4. **Fix the Issue** –  
    Use the info to write the correct code (solution).

**Example:**

public class Demo {

public static void main(String[] args) {

String name = null;

System.out.println(name.length()); // This line causes error

}

}

Output:

Exception in thread "main" java.lang.NullPointerException

at Demo.main(Demo.java:4)

### **Explanation:**

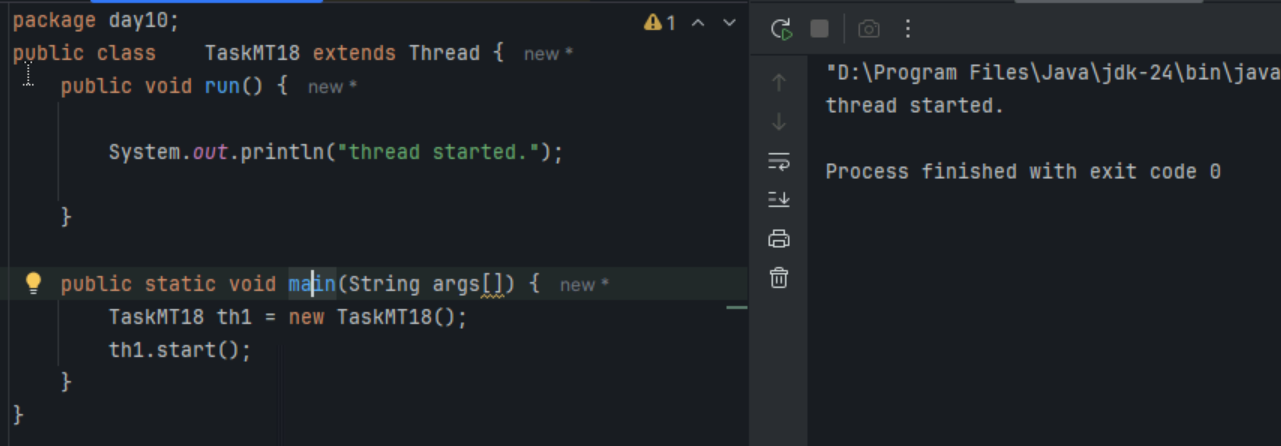
* **Error**: NullPointerException
* **Where**: Demo.java:4 → Line 4 in Demo.java
* **Why**: name is null, and calling .length() on it caused the error
* **Solution**: Make sure name is not null before using it

if (name != null) {

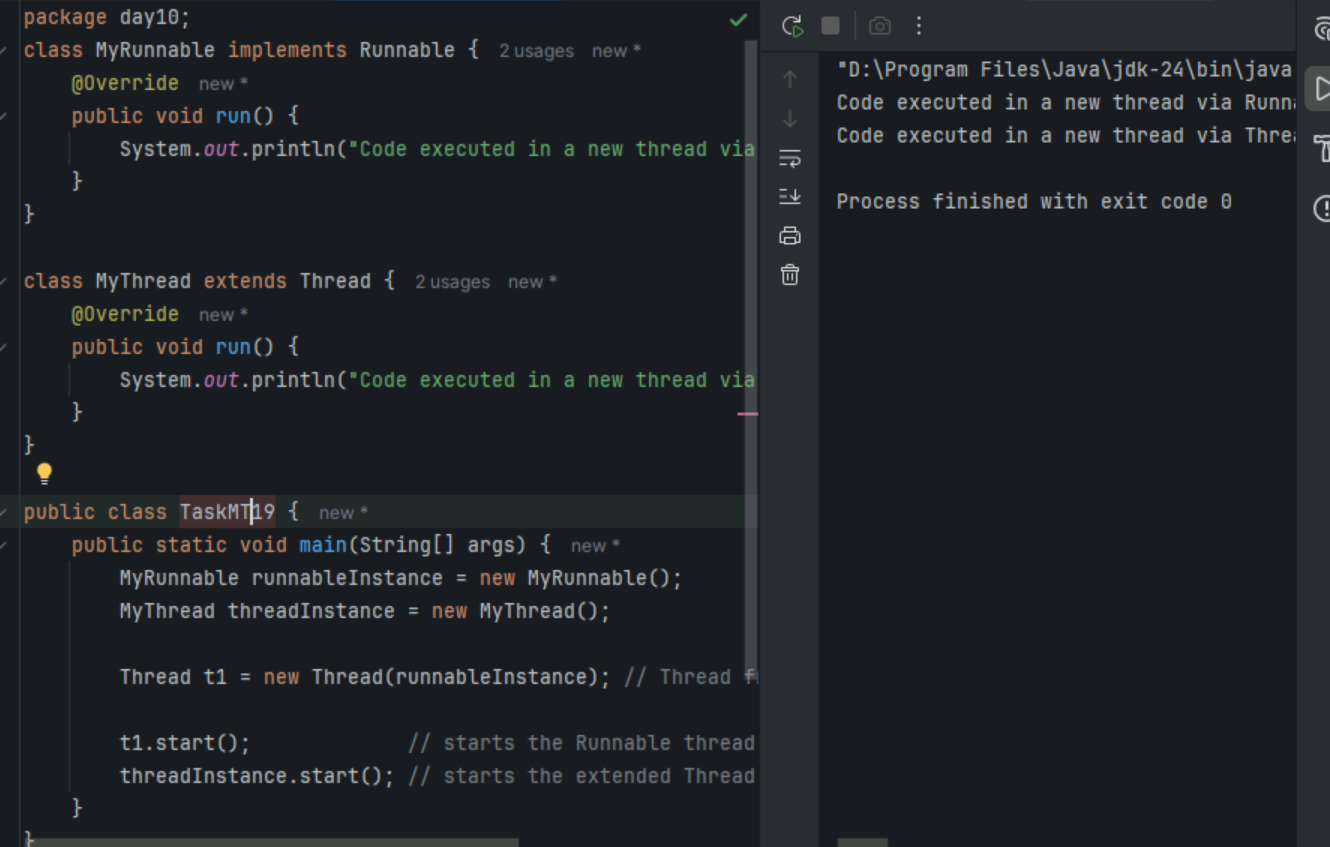
System.out.println(name.length());

}

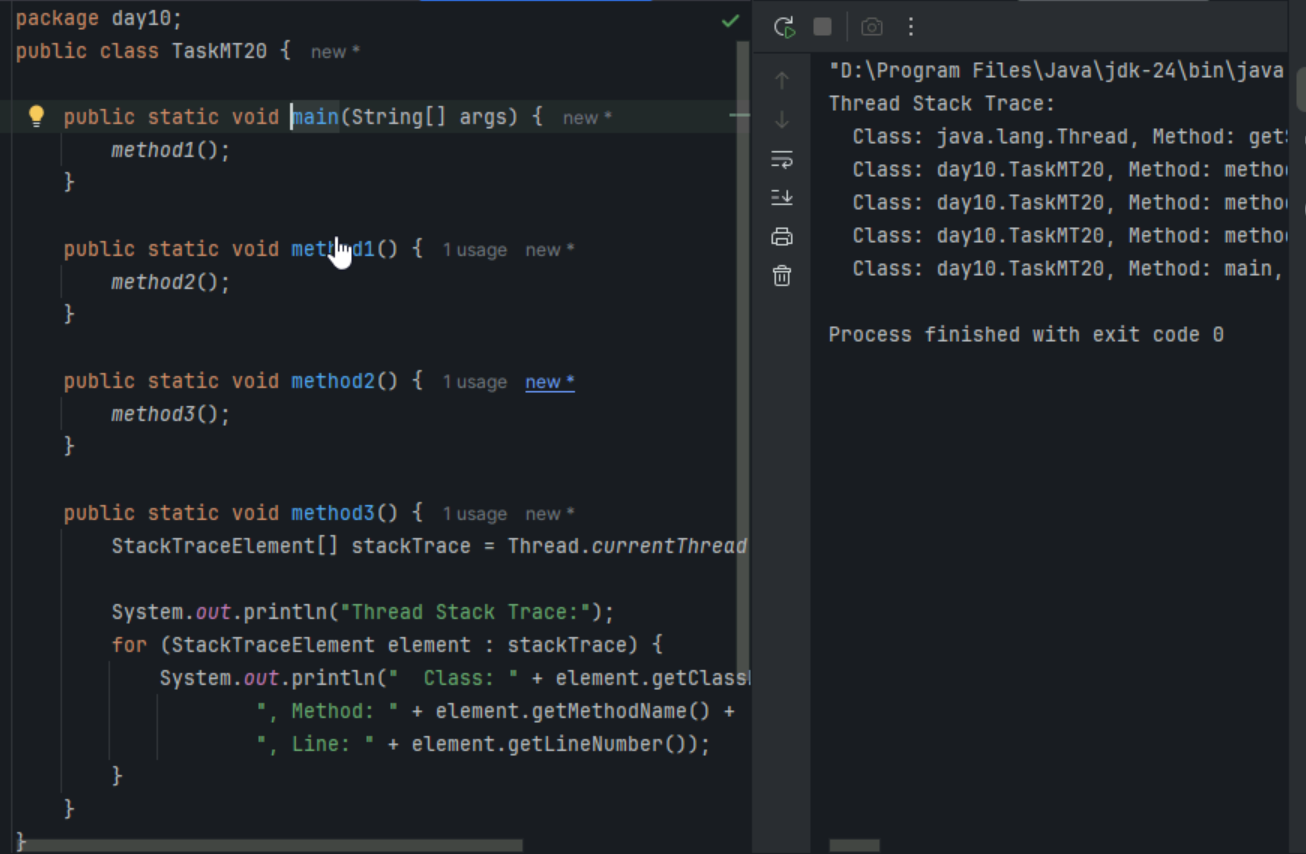
**Task18**

****

**Task19**

****

**Task20**

****