

**NATIONAL TECHNICAL UNIVERSITY OF UKRAINE  
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”**

Faculty of Informatics and Computer Engineering

Department of Computer Engineering

## **Distributed Information Systems**

### **Lab №3 Implement Runnable interface**

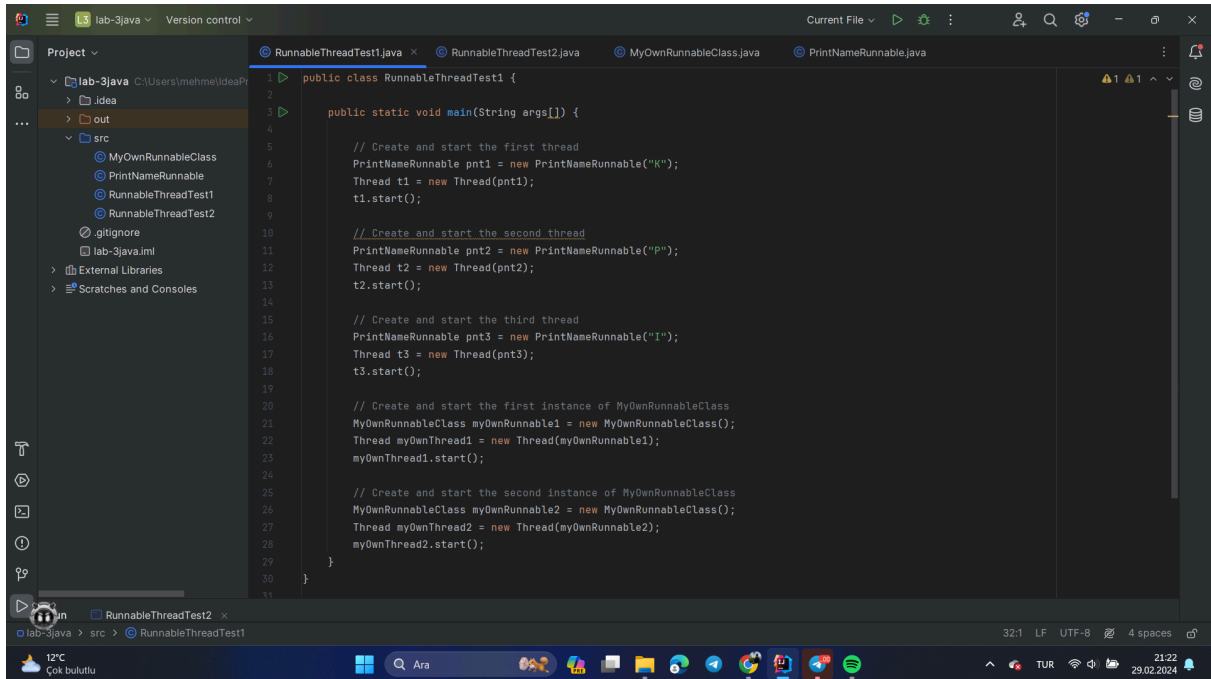
Student, group IM-14 FIOT

MEHMET KULUBEÇİOĞLU

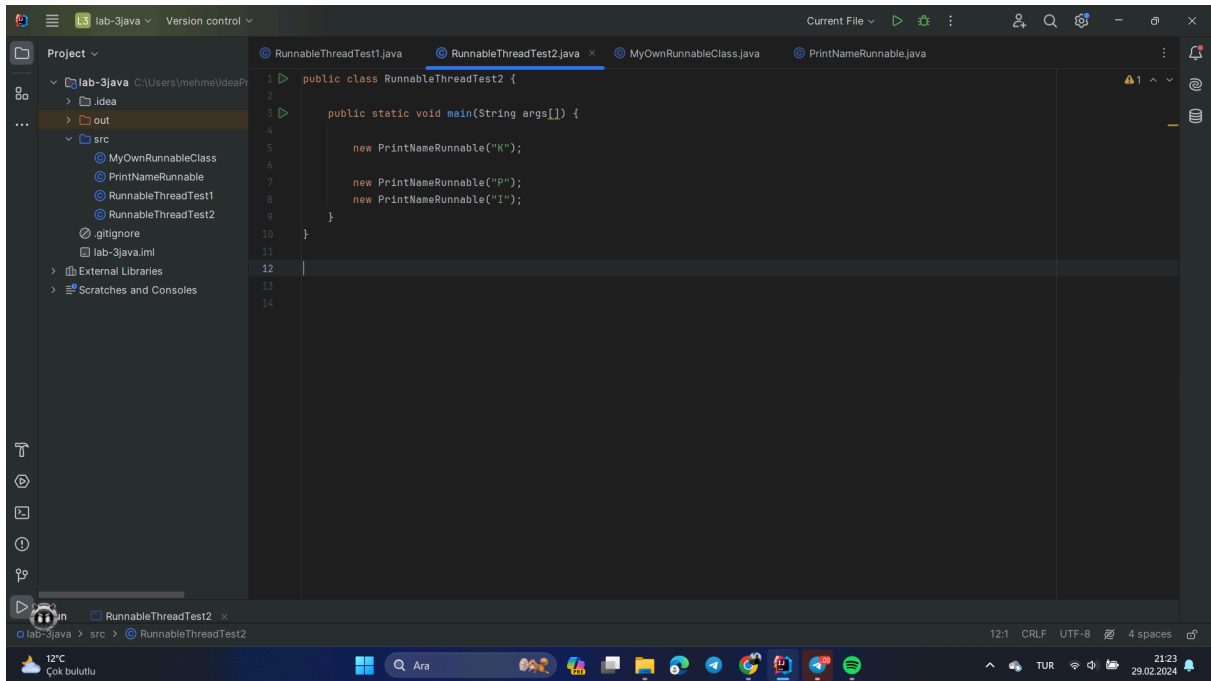
Reviewer YULIA TIMOFEEVA

Kyiv – 2024

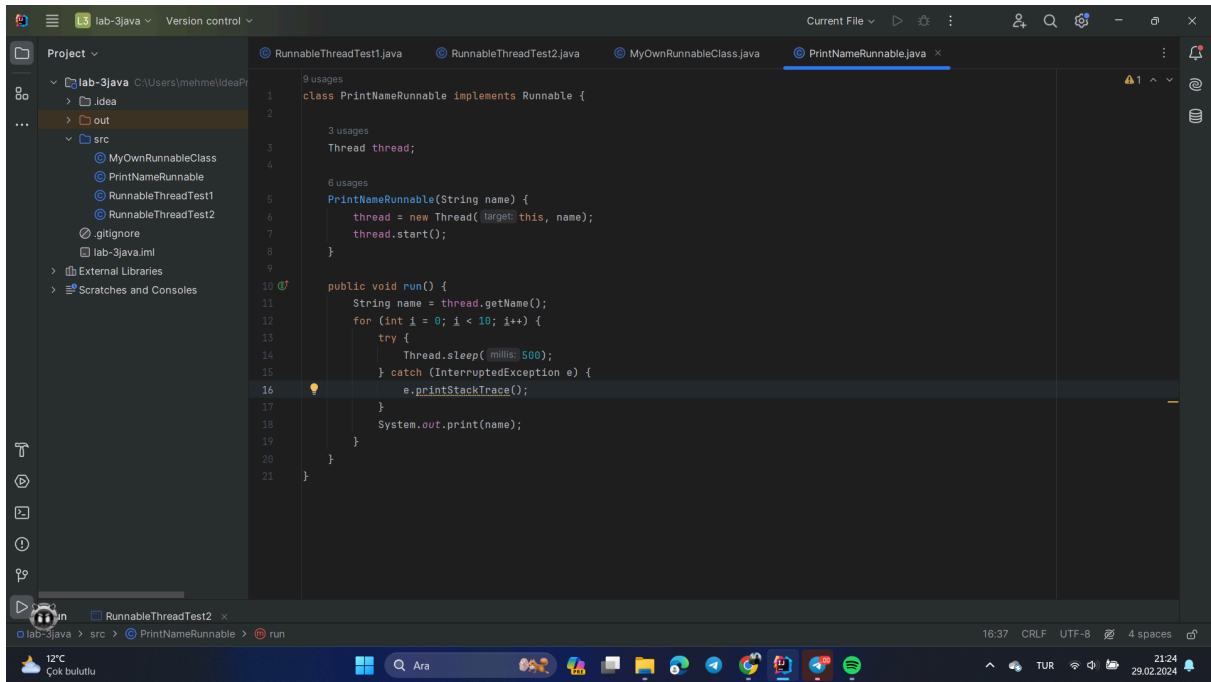
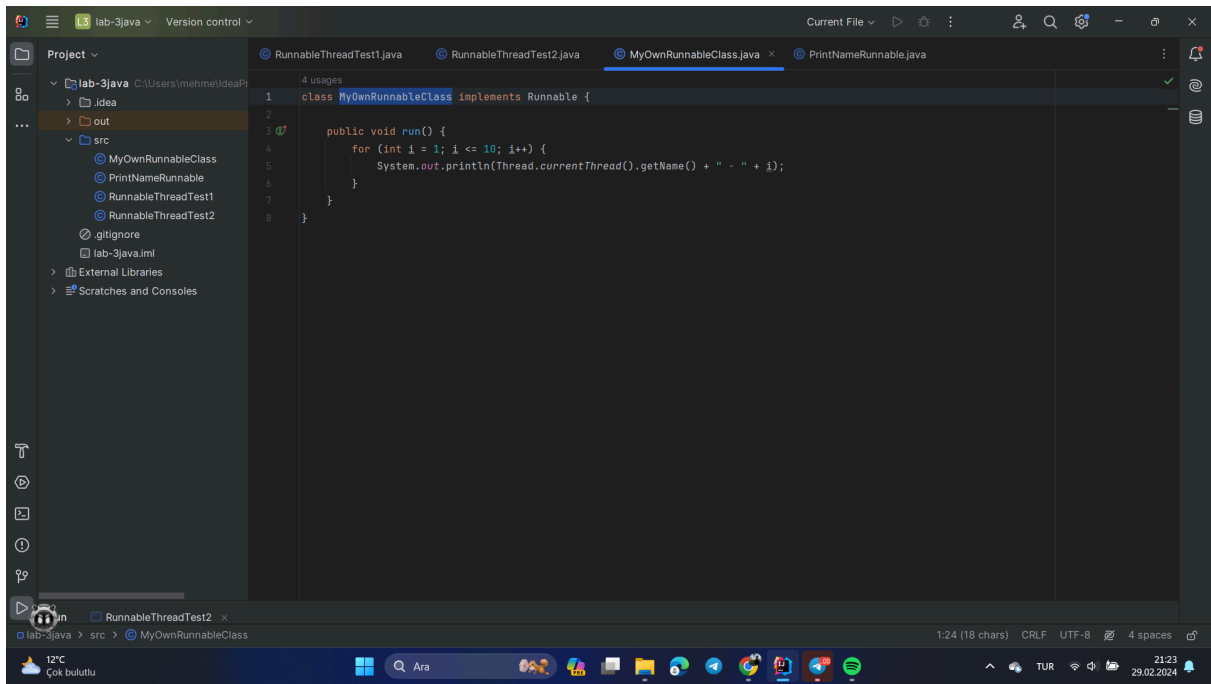
## MY FULL KODES:



```
1 public class RunnableThreadTest1 {
2
3     public static void main(String args[]) {
4
5         // Create and start the first thread
6         PrintNameRunnable pnt1 = new PrintNameRunnable("K");
7         Thread t1 = new Thread(pnt1);
8         t1.start();
9
10        // Create and start the second thread
11        PrintNameRunnable pnt2 = new PrintNameRunnable("P");
12        Thread t2 = new Thread(pnt2);
13        t2.start();
14
15        // Create and start the third thread
16        PrintNameRunnable pnt3 = new PrintNameRunnable("I");
17        Thread t3 = new Thread(pnt3);
18        t3.start();
19
20        // Create and start the first instance of MyOwnRunnableClass
21        MyOwnRunnableClass myOwnRunnable1 = new MyOwnRunnableClass();
22        Thread myOwnThread1 = new Thread(myOwnRunnable1);
23        myOwnThread1.start();
24
25        // Create and start the second instance of MyOwnRunnableClass
26        MyOwnRunnableClass myOwnRunnable2 = new MyOwnRunnableClass();
27        Thread myOwnThread2 = new Thread(myOwnRunnable2);
28        myOwnThread2.start();
29    }
30 }
31
```



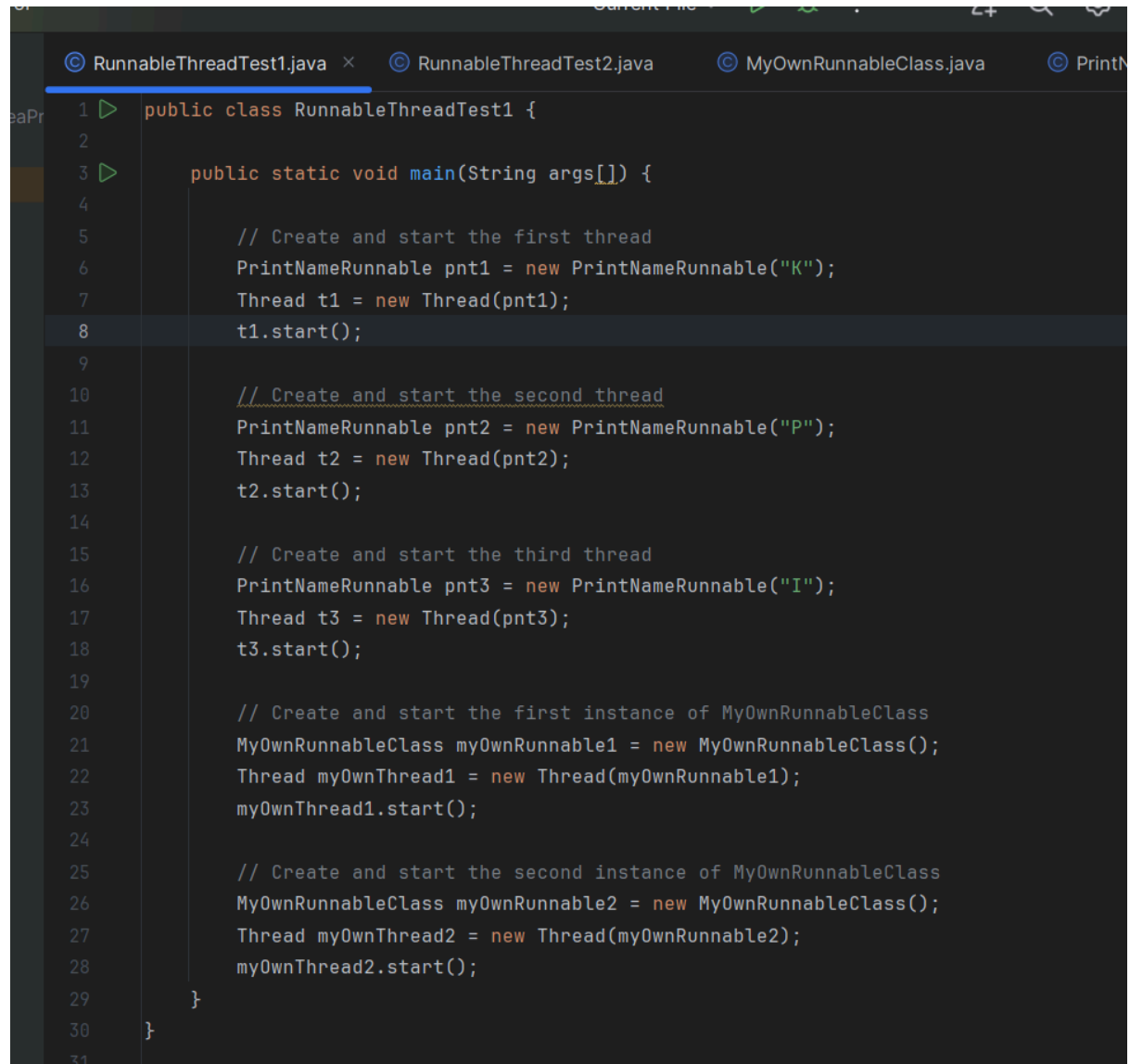
```
1 public class RunnableThreadTest2 {
2
3     public static void main(String args[]) {
4
5         new PrintNameRunnable("K");
6         new PrintNameRunnable("P");
7         new PrintNameRunnable("I");
8     }
9 }
10
11
12
13
14
```



## Step by Step Explanation of Codes

### 1. RunnableThreadTest1.java:

**Purpose:** To illustrate threading concepts using MyOwnRunnableClass and PrintNameRunnable examples running in different threads.



```
1 public class RunnableThreadTest1 {
2
3     public static void main(String args[]) {
4
5         // Create and start the first thread
6         PrintNameRunnable pnt1 = new PrintNameRunnable("K");
7         Thread t1 = new Thread(pnt1);
8         t1.start();
9
10        // Create and start the second thread
11        PrintNameRunnable pnt2 = new PrintNameRunnable("P");
12        Thread t2 = new Thread(pnt2);
13        t2.start();
14
15        // Create and start the third thread
16        PrintNameRunnable pnt3 = new PrintNameRunnable("I");
17        Thread t3 = new Thread(pnt3);
18        t3.start();
19
20        // Create and start the first instance of MyOwnRunnableClass
21        MyOwnRunnableClass myOwnRunnable1 = new MyOwnRunnableClass();
22        Thread myOwnThread1 = new Thread(myOwnRunnable1);
23        myOwnThread1.start();
24
25        // Create and start the second instance of MyOwnRunnableClass
26        MyOwnRunnableClass myOwnRunnable2 = new MyOwnRunnableClass();
27        Thread myOwnThread2 = new Thread(myOwnRunnable2);
28        myOwnThread2.start();
29    }
30 }
31
```

### RunnableThreadTest1.java Output:

[illegible]

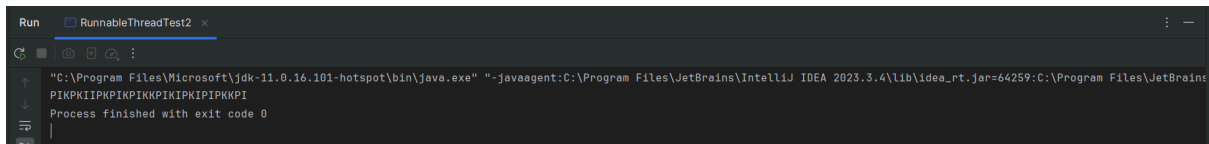
## 2. RunnableThreadTest2.java:

**Purpose:** to demonstrate threading by creating `PrintNameRunnable` instances using the `new` operator.

© RunnableThreadTest1.java      © RunnableThreadTest2.java    ×    © MyOv

```
1  ▶ public class RunnableThreadTest2 {
2
3  ▶      public static void main(String args[]) {
4
5          new PrintNameRunnable("K");
6
7          new PrintNameRunnable("P");
8          new PrintNameRunnable("I");
9      }
10 }
```


## RunnableThreadTest2.java output:



```
Run RunnableThreadTest2 x
"C:\Program Files\Microsoft\jdk-11.0.16-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=64259:C:\Program Files\JetBrains
PIKPKIIPKPIKPIKPIKPIKPIKPI
Process finished with exit code 0
```

## MyOwnRunnableClass:

- This class implements the Runnable interface and defines the run() method. The run() method prints the name of the thread and the numbers 1 to 10.



```
RunnableThreadTest1.java RunnableThreadTest2.java MyOwnRunnableClass.java PrintNameR
4 usages
1 class MyOwnRunnableClass implements Runnable {
2
3     public void run() {
4         for (int i = 1; i <= 10; i++) {
5             System.out.println(Thread.currentThread().getName() + " - " + i);
6         }
7     }
8 }
```

## PrintNameRunnable:

- This class also implements the Runnable interface and defines the run() method. The run() method prints the name of the thread every 500 milliseconds.
- Five PrintNameRunnable instances ("K", "P", "I") and two MyOwnRunnableClass instances are created and run. Each PrintNameRunnable instance prints a different letter ("K", "P", "I"), while MyOwnRunnableClass instances print thread names and numbers 1 through 10.

```
readTest1.java  RunnableThreadTest2.java  MyOwnRunnableClass.java  PrintNameRunnable.java x
9 usages
1  class PrintNameRunnable implements Runnable {
2
3      3 usages
4      Thread thread;
5
6      6 usages
7      PrintNameRunnable(String name) {
8          thread = new Thread( target: this, name);
9          thread.start();
10     }
11
12     public void run() {
13         String name = thread.getName();
14         for (int i = 0; i < 10; i++) {
15             try {
16                 Thread.sleep( millis: 500);
17             } catch (InterruptedException e) {
18                 e.printStackTrace();
19             }
20             System.out.print(name);
21         }
22     }
23 }
```

**Both classes(PrintNameRunnable and MyOwnRunnableClass) :**

- It implements the Runnable interface.
- It defines a method called run().

**Differences:**

- MyOwnRunnableClass also prints numbers, while PrintNameRunnable just prints the name of the thread.
- PrintNameRunnable delays 500 milliseconds before each print, while MyOwnRunnableClass does not.