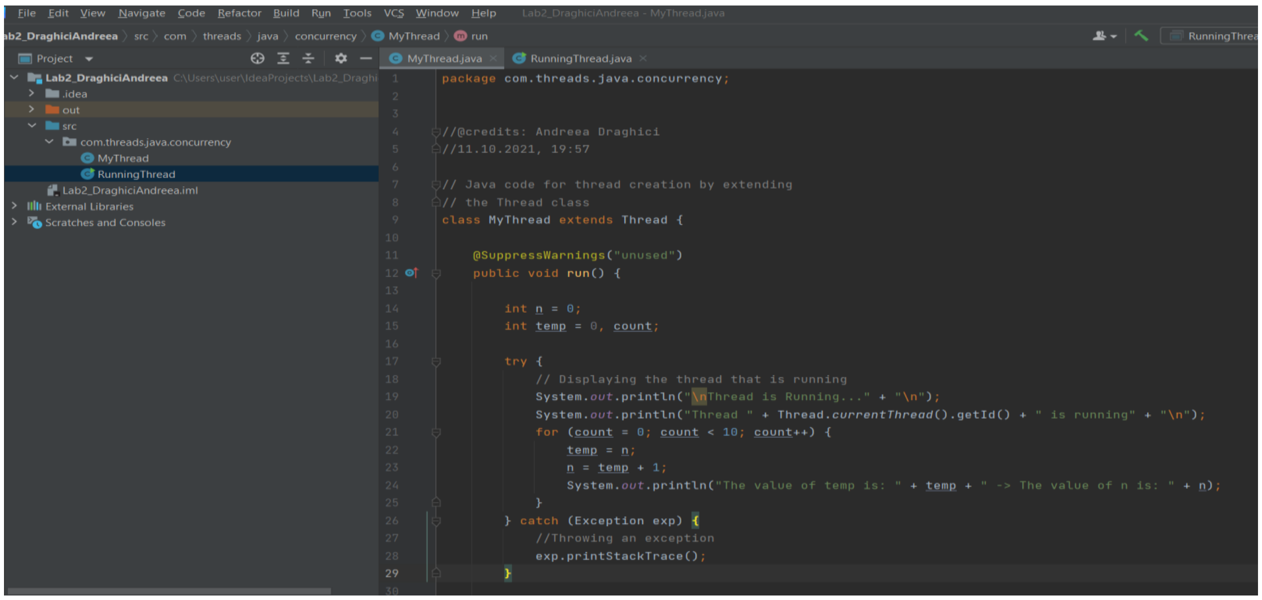
Draghici Andreea-Maria
CR 3.1B

Raport Laborator 2

Rezolvare cerinta 1:

Am creat un proiect nou in Intellij IDEA, apoi am creat un package denumit com.threads.java.concurrency unde am creat doua clase, MyThread si RunningThread, iar ca prim program, am implementat algoritmul de la cerinta 2, dar doar pentru un singur fir. Observatiile le-am adaugat mai jos, atasand si un screenshot cu implementarea si rezultatul.

Clasa MyThread suprascrie metoda run() in clasa Thread, iar aici un fir "isi incepe viata".

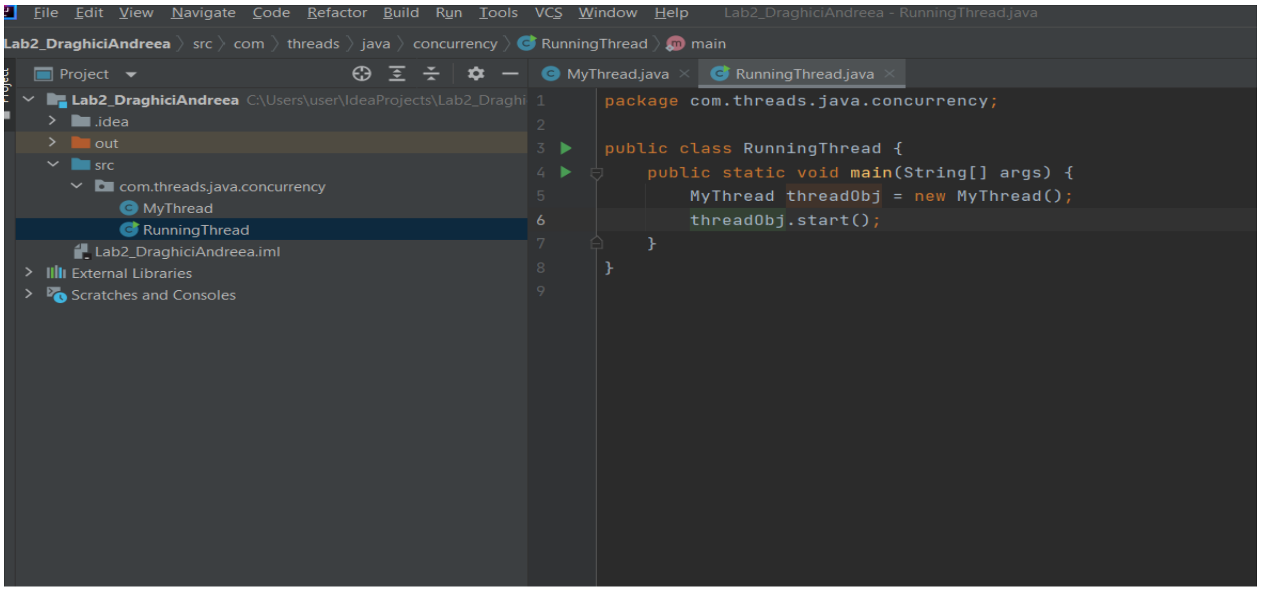

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Lab2_DraghiciAndreea - MyThread.java
ab2_DraghiciAndreea › src › com › threads › java › concurrency › MyThread › run
RunningThrea
Project
Lab2_DraghiciAndreea C:\Users\user\IdeaProjects\Lab2_Draghi
.idea
out
src
com.threads.java.concurrency
MyThread
RunningThread
Lab2_DraghiciAndreea.iml
External Libraries
Scratches and Consoles
MyThread.java
RunningThread.java

```java
package com.threads.java.concurrency;


//@credits: Andreea Draghici
//11.10.2021, 19:57

// Java code for thread creation by extending
// the Thread class
class MyThread extends Thread {

    @SuppressWarnings("unused")
    public void run() {

        int n = 0;
        int temp = 0, count;

        try {
            // Displaying the thread that is running
            System.out.println("\nThread is Running..." + "\n");
            System.out.println("Thread " + Thread.currentThread().getId() + " is running" + "\n");
            for (count = 0; count < 10; count++) {
                temp = n;
                n = temp + 1;
                System.out.println("The value of temp is: " + temp + " -> The value of n is: " + n);
            }
        } catch (Exception exp) {
            //Throwing an exception
            exp.printStackTrace();
        }
```

Programul utilizeaza doua variabile intregi temp si n si ruleaza un fir care va incrementa variabila n pana la 10 si variabila temp, care va fi mereu n-1.

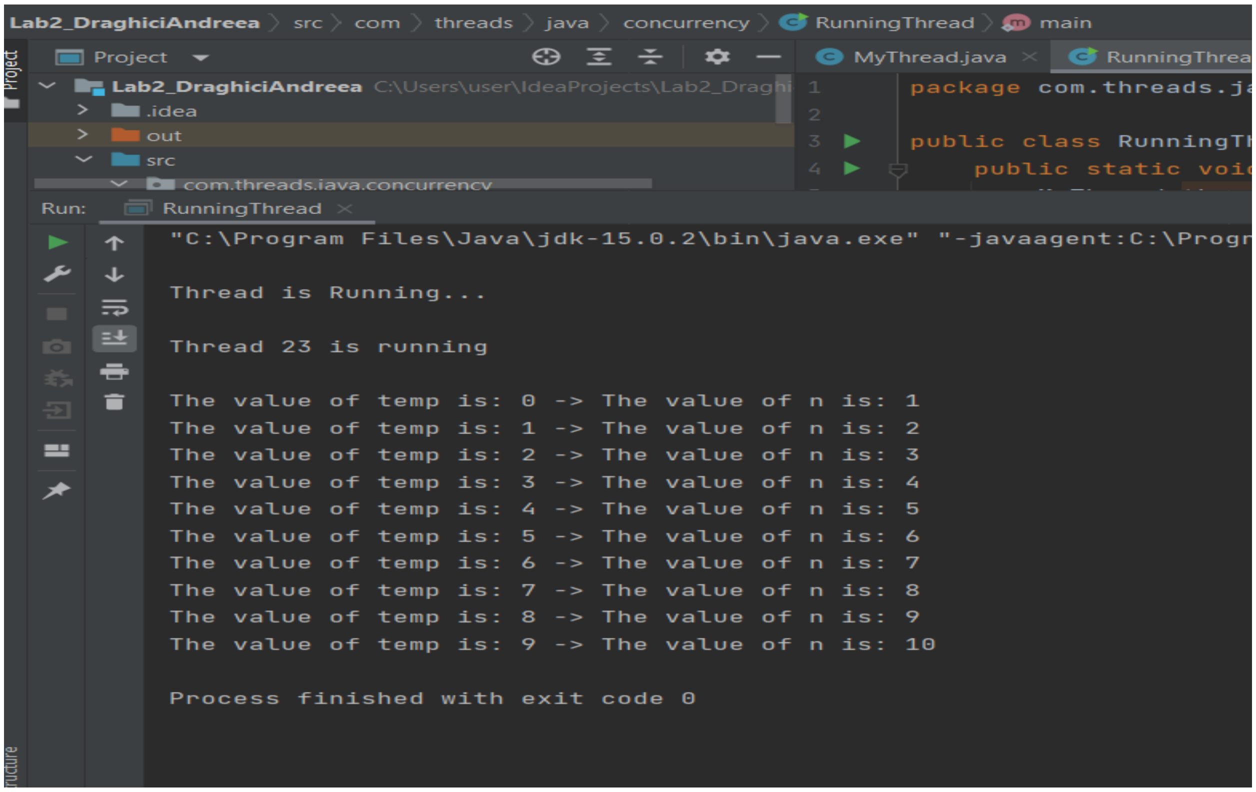
In clasa RunningThread cream un nou obiect denumit threadObj si apelam metoda start() pentru a incepe executarea unui thread.

Draghici Andreea-Maria
CR 3.1B



```
package com.threads.java.concurrency;

public class RunningThread {
    public static void main(String[] args) {
        MyThread threadObj = new MyThread();
        threadObj.start();
    }
}
```

Mai jos se pot vedea rezultatele finale in urma rularii programului de mai sus , unde aici un fir “isi incepe viata”:



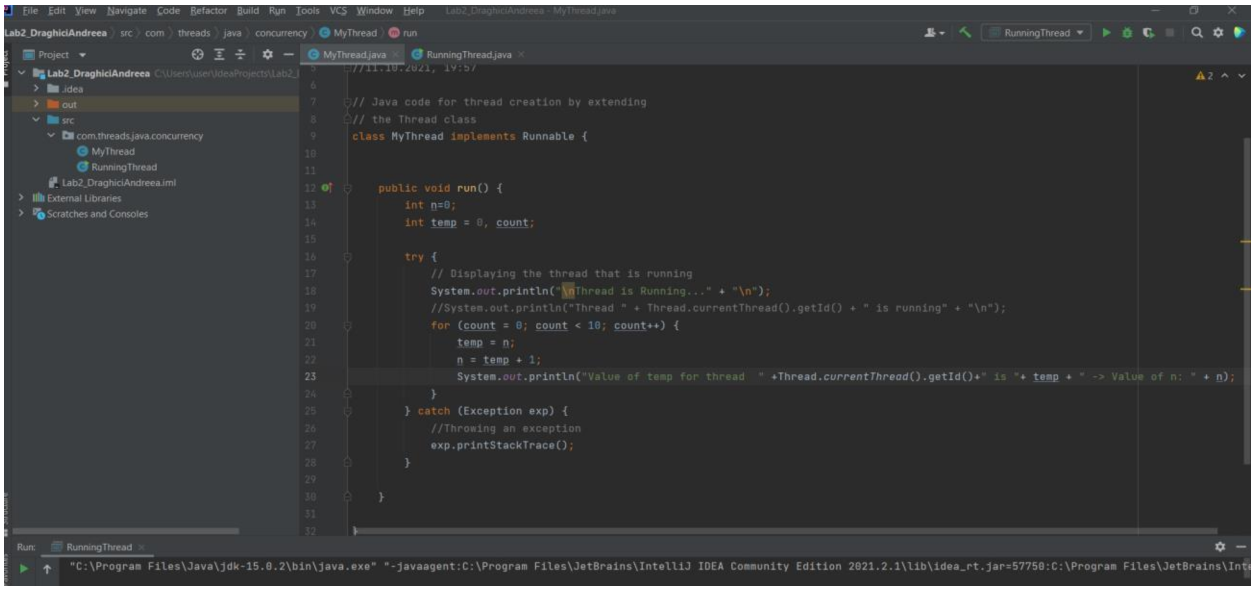
```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Progr

Thread is Running...

Thread 23 is running

The value of temp is: 0 -> The value of n is: 1
The value of temp is: 1 -> The value of n is: 2
The value of temp is: 2 -> The value of n is: 3
The value of temp is: 3 -> The value of n is: 4
The value of temp is: 4 -> The value of n is: 5
The value of temp is: 5 -> The value of n is: 6
The value of temp is: 6 -> The value of n is: 7
The value of temp is: 7 -> The value of n is: 8
The value of temp is: 8 -> The value of n is: 9
The value of temp is: 9 -> The value of n is: 10

Process finished with exit code 0
```

Draghici Andreea-Maria
CR 3.1B

Rezolvare cerinta 2:

Pentru reolvarea cerintei 2, am ales sa folosesc aceeasi clasa MyThred care de data aceasta implementeaza interfata Runnable.

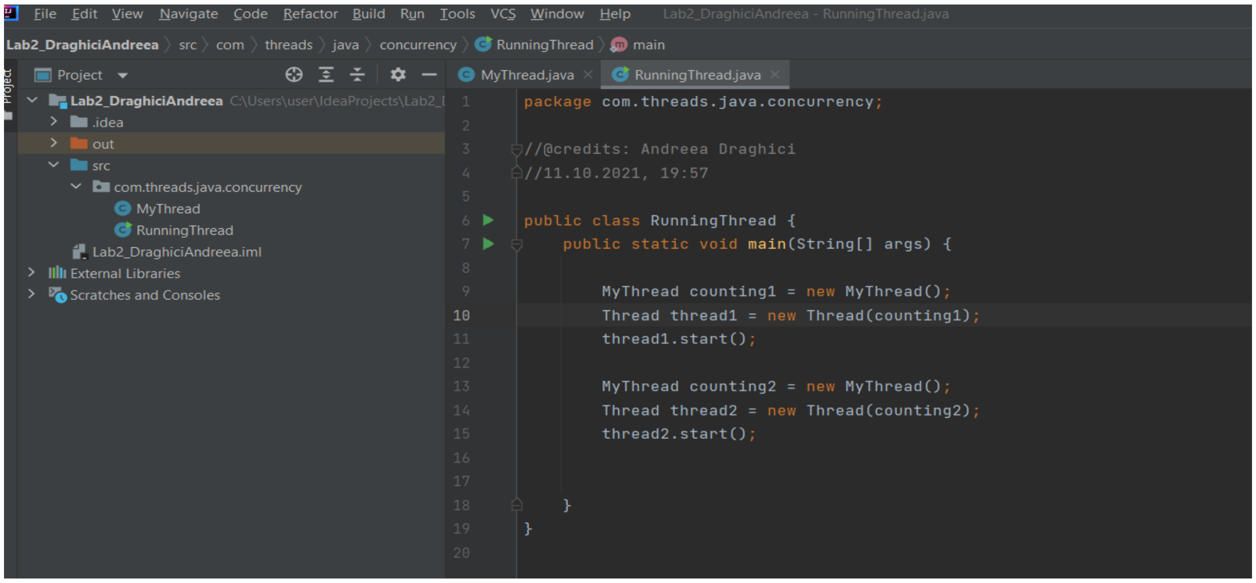
Programul utilizeaza doua variabile intregi temp si n si ruleaza doua fire care vor incrementa variabila n pana la 10 si variabila temp, care va fi mereu n-1.



```java
//11.10.2021, 19:57

// Java code for thread creation by extending
// the Thread class
class MyThread implements Runnable {


    public void run() {
        int n=0;
        int temp = 0, count;

        try {
            // Displaying the thread that is running
            System.out.println("\nThread is Running..." + "\n");
            //System.out.println("Thread " + Thread.currentThread().getId() + " is running" + "\n");
            for (count = 0; count < 10; count++) {
                temp = n;
                n = temp + 1;
                System.out.println("Value of temp for thread  " +Thread.currentThread().getId()+" is "+ temp + " -> Value of n: " + n);
            }
        } catch (Exception exp) {
            //Throwing an exception
            exp.printStackTrace();
        }

    }

}
```

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.1\lib\idea_rt.jar=57750:C:\Program Files\JetBrains\Inte
```

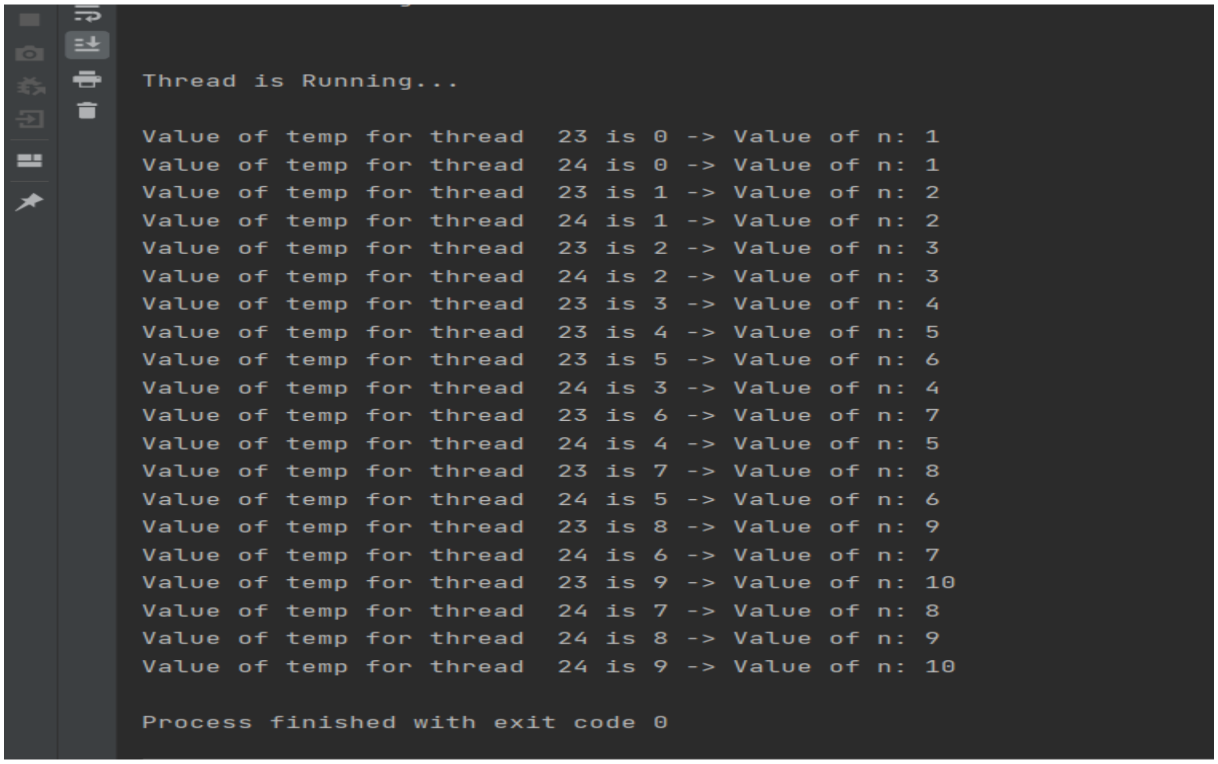
Apoi instantiem doua obiecte thread si apelam metoda start() pe aceste obiecte.



```java
package com.threads.java.concurrency;

//@credits: Andreea Draghici
//11.10.2021, 19:57

public class RunningThread {
    public static void main(String[] args) {

        MyThread counting1 = new MyThread();
        Thread thread1 = new Thread(counting1);
        thread1.start();

        MyThread counting2 = new MyThread();
        Thread thread2 = new Thread(counting2);
        thread2.start();


    }
}
```

Draghici Andreea-Maria
CR 3.1B

Mai jos se pot vedea rezultatele finale in urma rularii programului de mai sus , unde aici doua fire “isi incep viata”:



```
Thread is Running...

Value of temp for thread  23 is 0 -> Value of n: 1
Value of temp for thread  24 is 0 -> Value of n: 1
Value of temp for thread  23 is 1 -> Value of n: 2
Value of temp for thread  24 is 1 -> Value of n: 2
Value of temp for thread  23 is 2 -> Value of n: 3
Value of temp for thread  24 is 2 -> Value of n: 3
Value of temp for thread  23 is 3 -> Value of n: 4
Value of temp for thread  23 is 4 -> Value of n: 5
Value of temp for thread  23 is 5 -> Value of n: 6
Value of temp for thread  24 is 3 -> Value of n: 4
Value of temp for thread  23 is 6 -> Value of n: 7
Value of temp for thread  24 is 4 -> Value of n: 5
Value of temp for thread  23 is 7 -> Value of n: 8
Value of temp for thread  24 is 5 -> Value of n: 6
Value of temp for thread  23 is 8 -> Value of n: 9
Value of temp for thread  24 is 6 -> Value of n: 7
Value of temp for thread  23 is 9 -> Value of n: 10
Value of temp for thread  24 is 7 -> Value of n: 8
Value of temp for thread  24 is 8 -> Value of n: 9
Value of temp for thread  24 is 9 -> Value of n: 10

Process finished with exit code 0
```

Observatie: Daca implementam interfata Runnable, clasa noastra MyThread poate extinde in continuare alte clase de baza, adica putem realiza functionalitatea de baza a unui thread extinzand clasa de baza Thread.