Day – 9 Java

class ----->  class ======> extends  ====> no multiple inheritance  
interface ----> class =====> Implements  
interface ---> interface ====> extends

Exceptions:   --> unexpected situations

  2 categoried  
    - Checked  ---> compile time  
    - Unchecked  ---> Runtime time..

  pre defined exception classes  
    FileNotFound  
    ArrayIndexoutofBounds exception  
    NullPointer Exception  
    Arthematic Exception ...

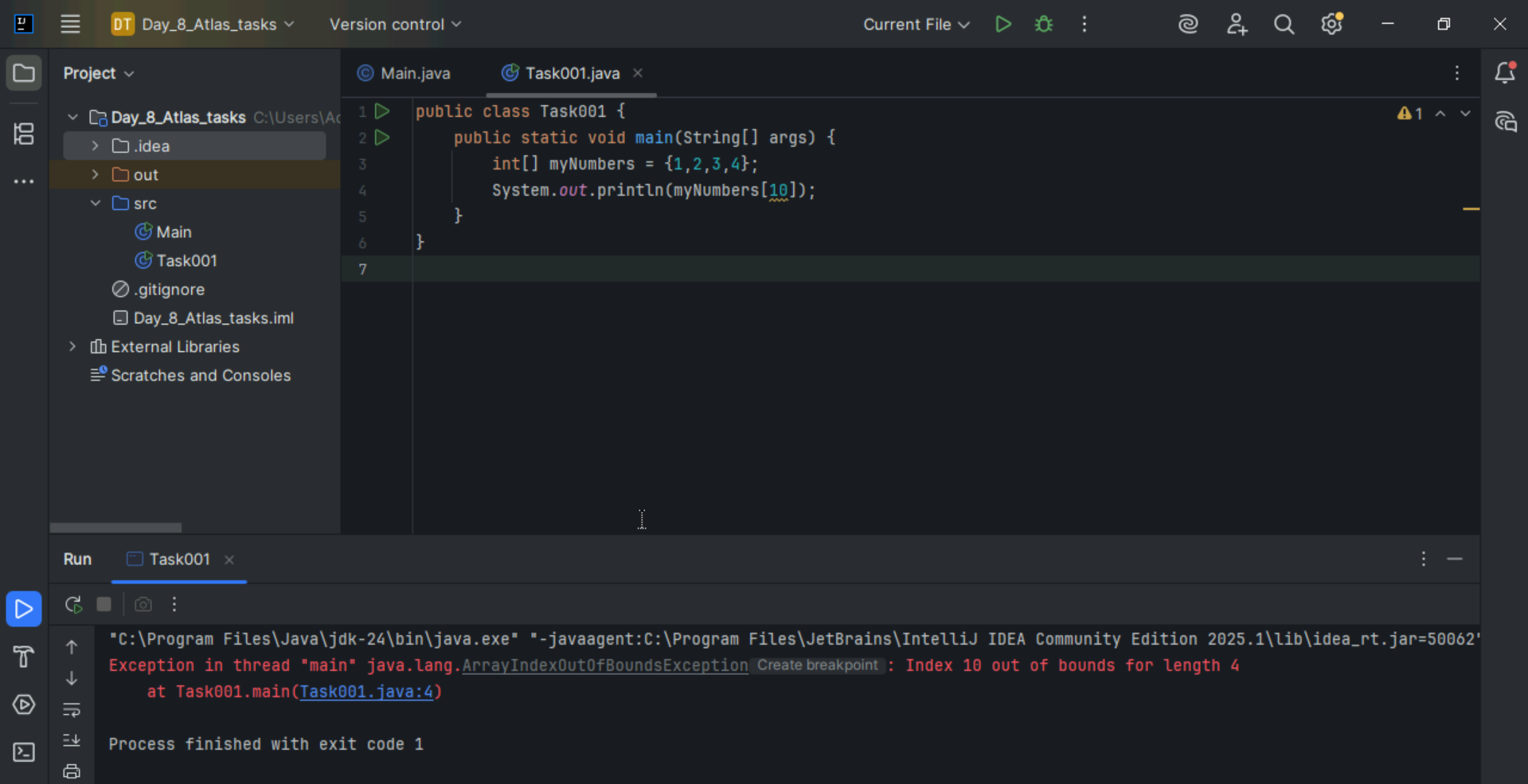
  user defined Exceptions  
try catch blocks  
  try{

  }catch(Exception ex){

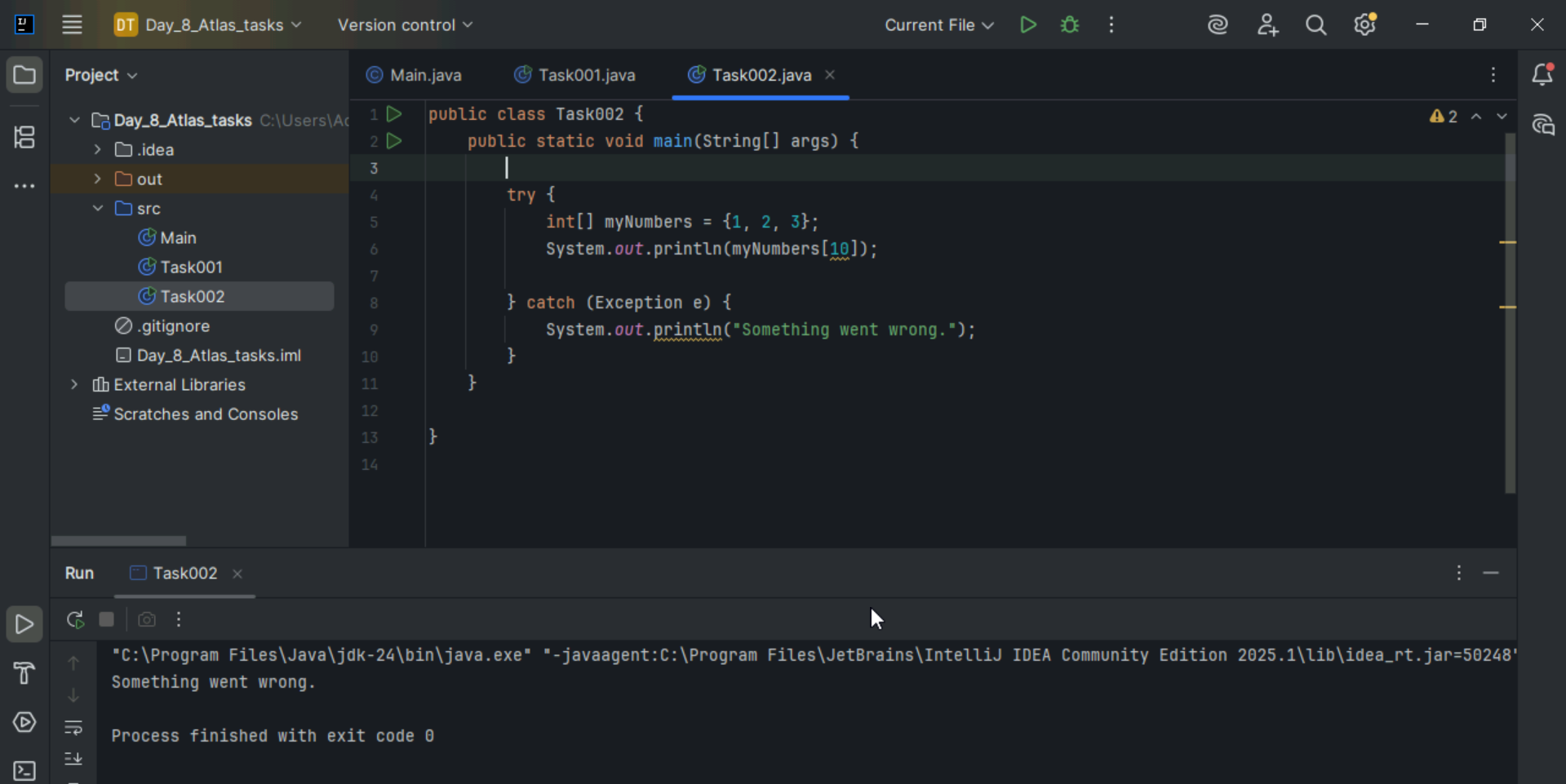
  }catch(ArthematicException ex1{  
  }  
  finally{

  sout(" ");  
  }

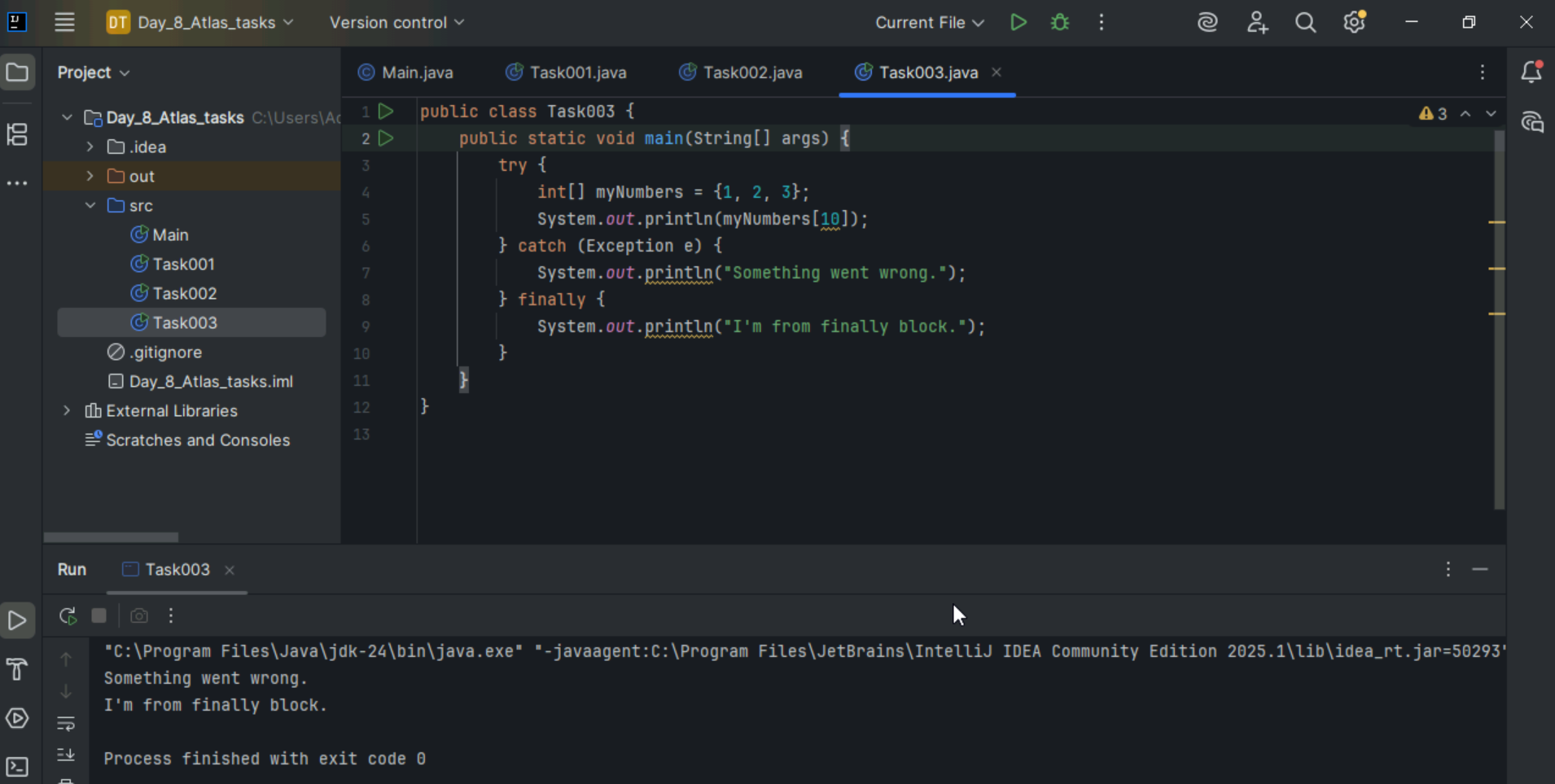
**Task 1:** Check for exception

****

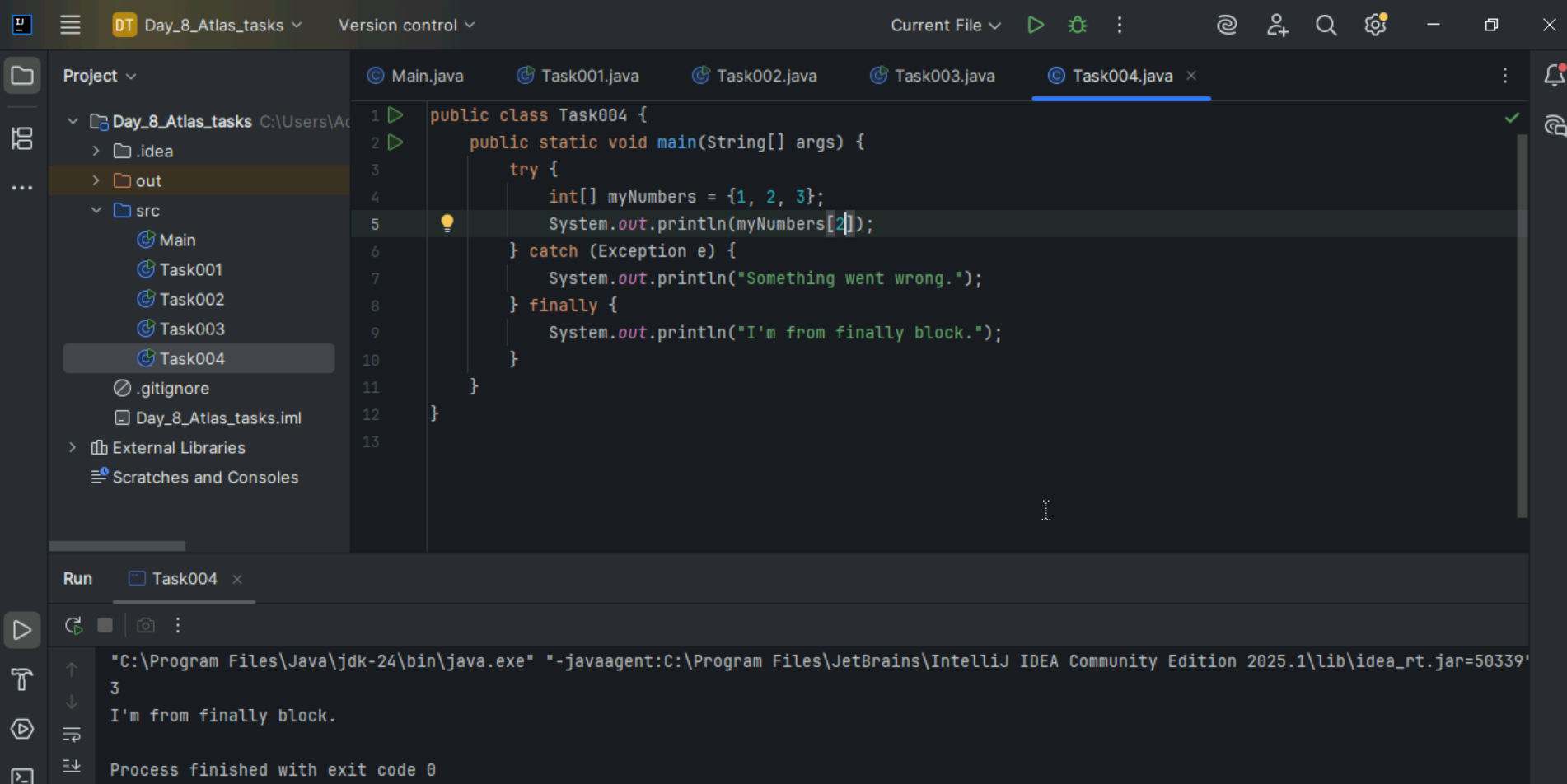
Task 2:



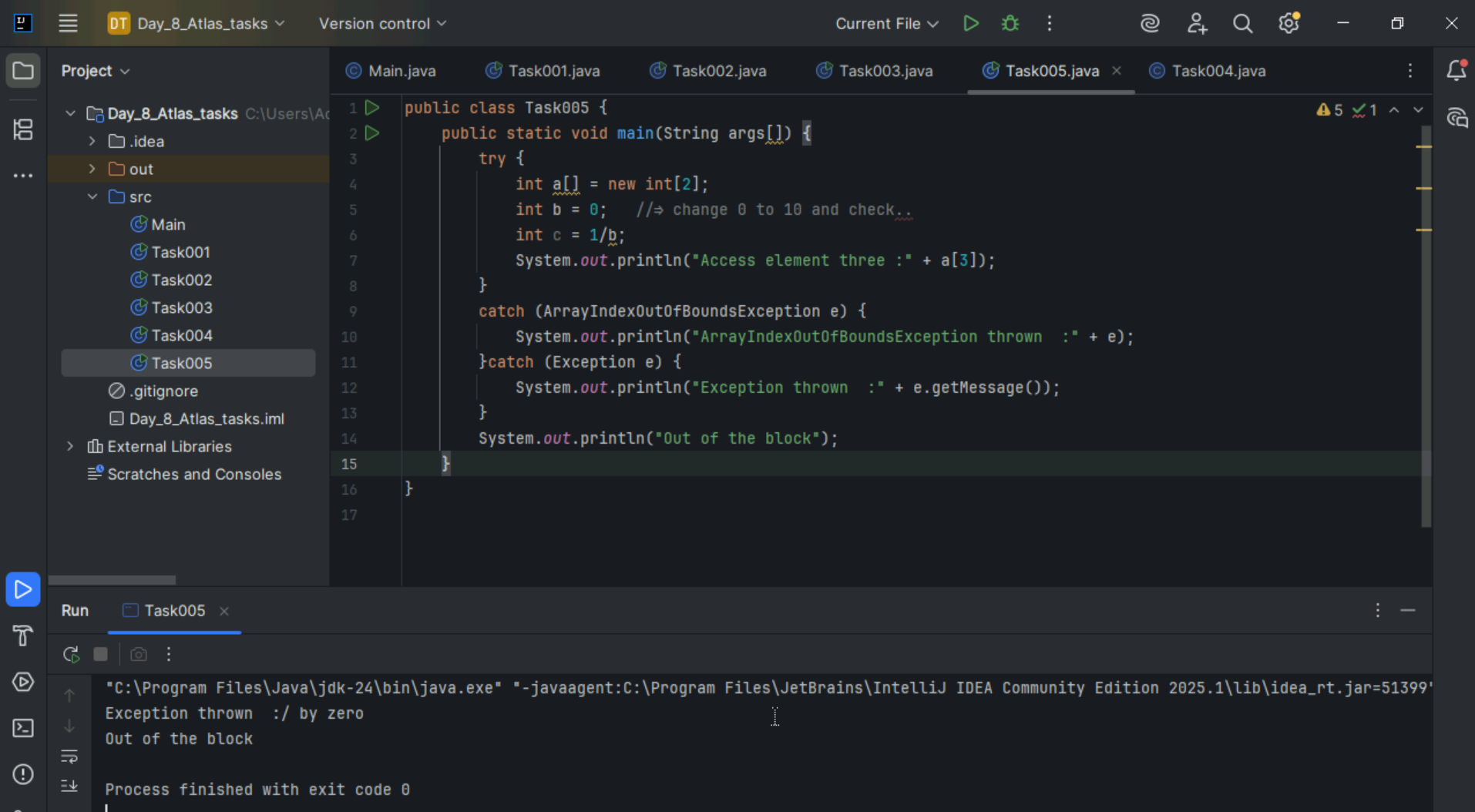
Task 3:



Task 4:

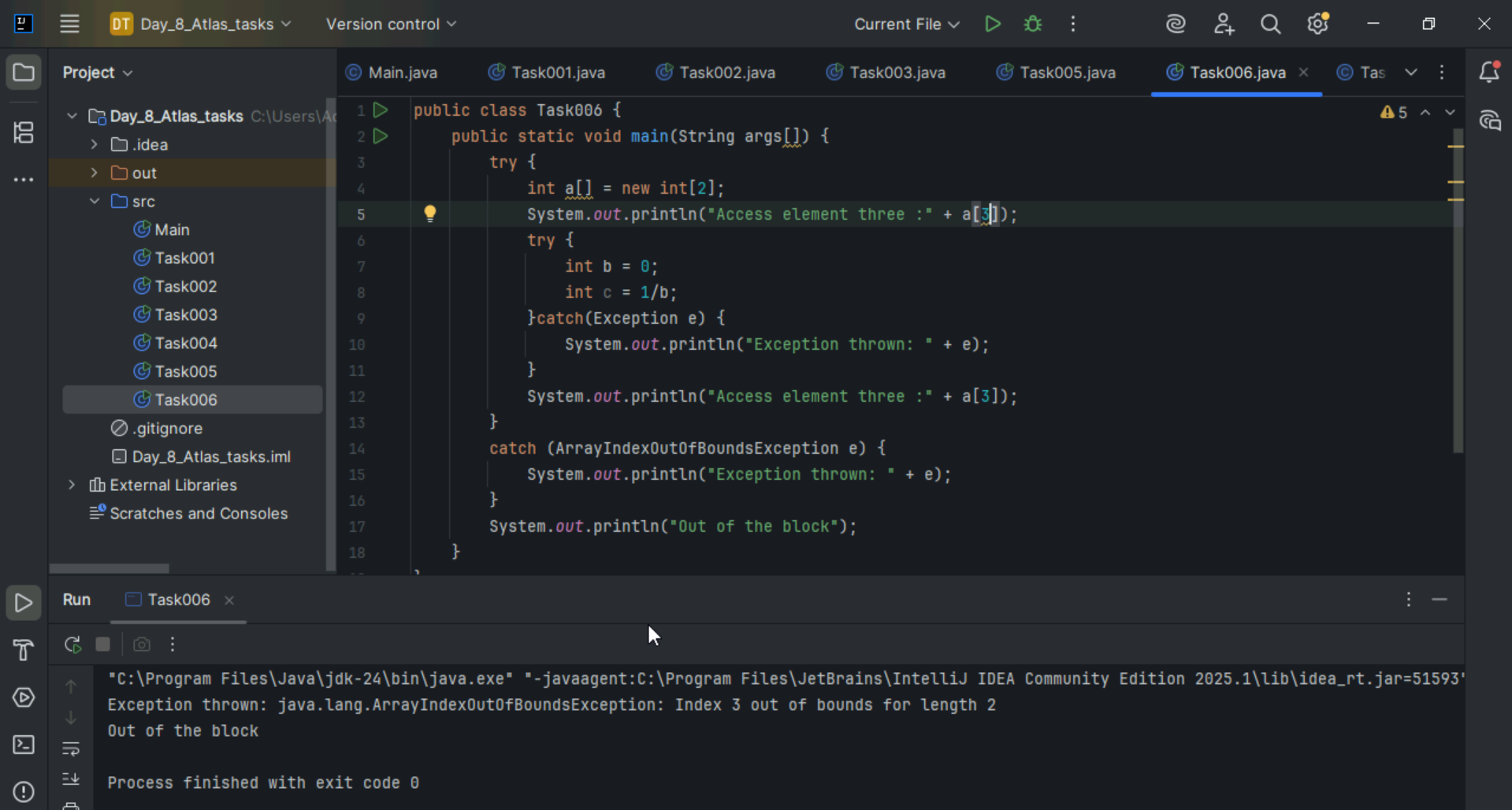


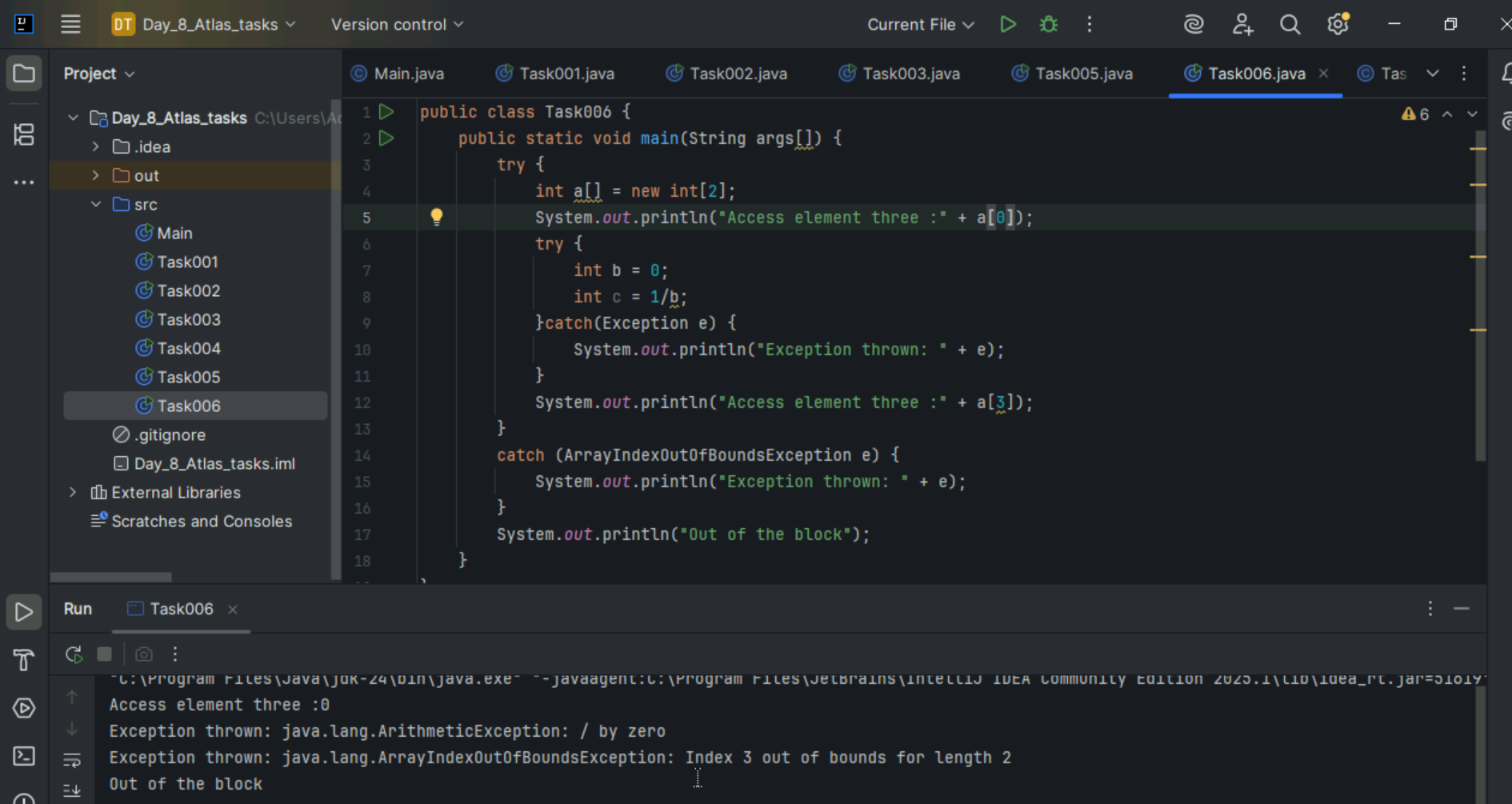
Task 5:



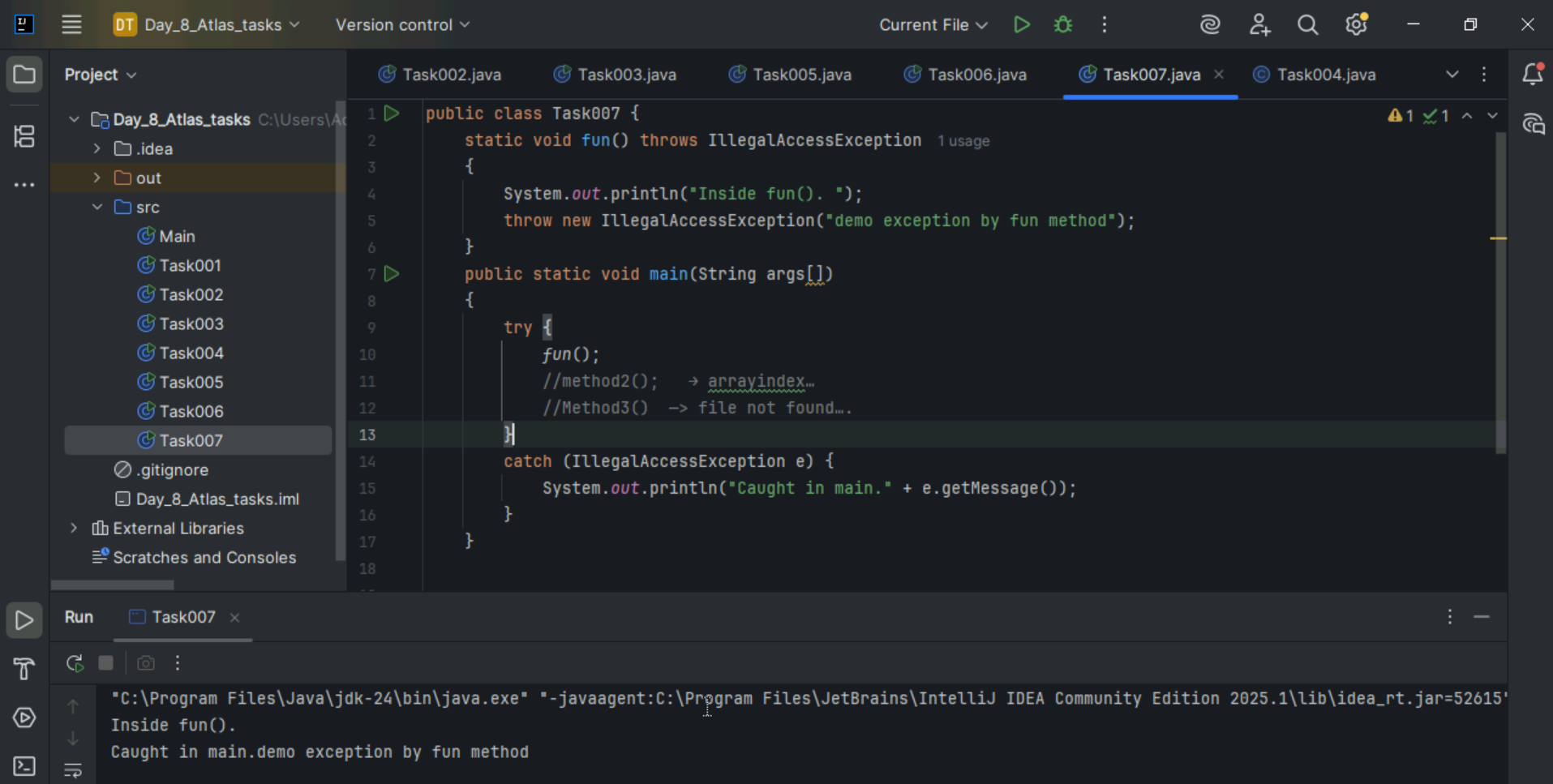


Task 6:





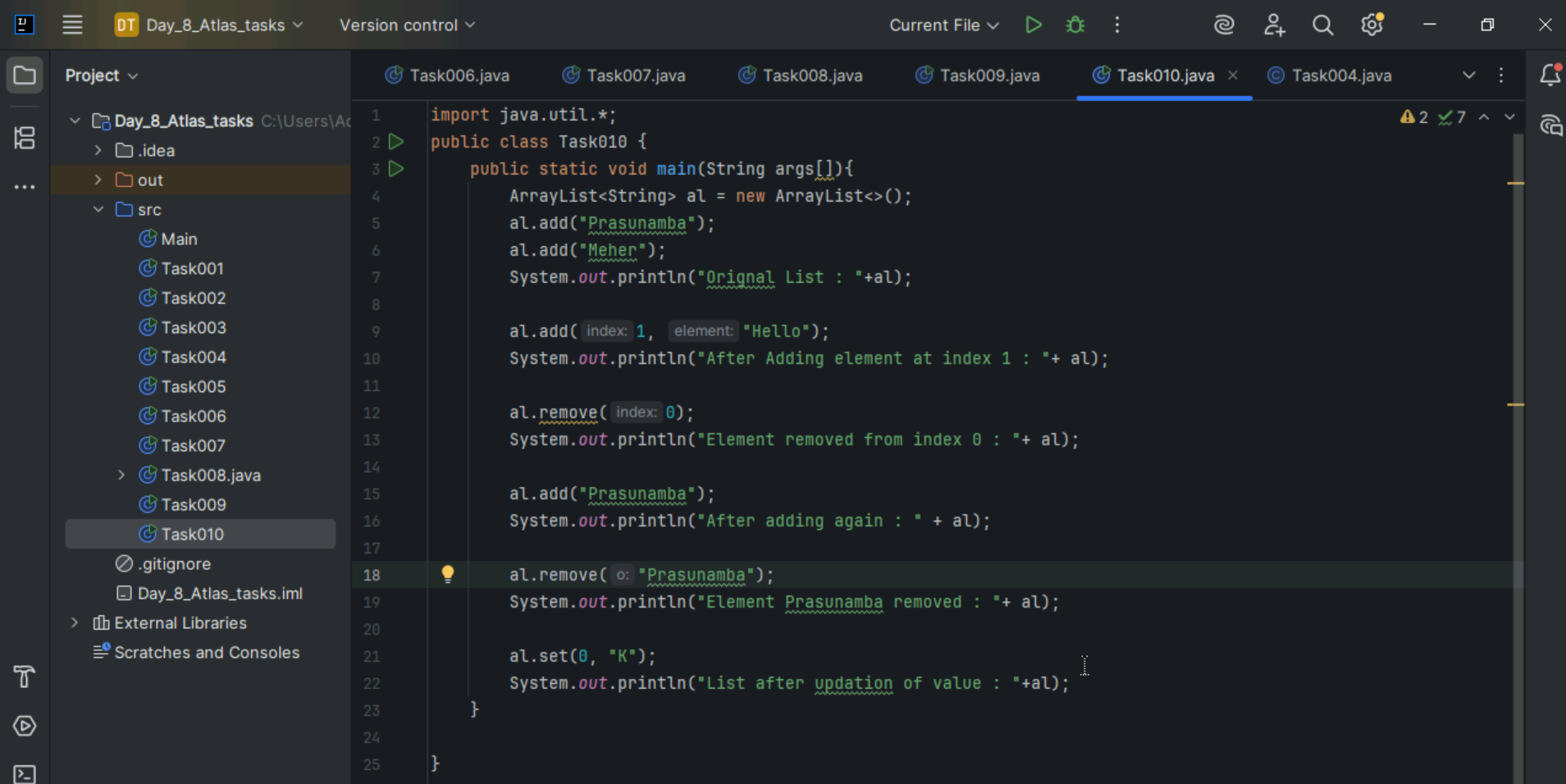
Task 7:

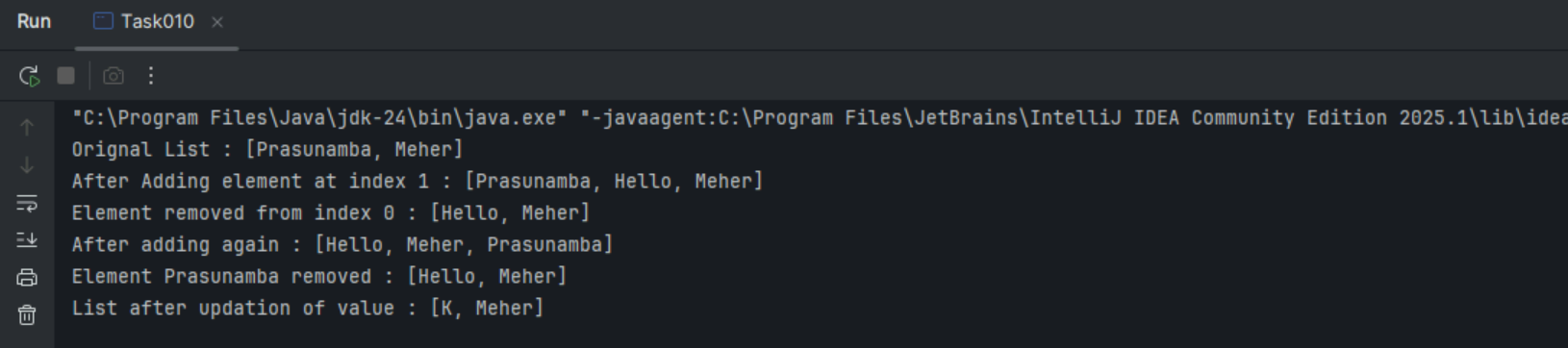


Task 8:

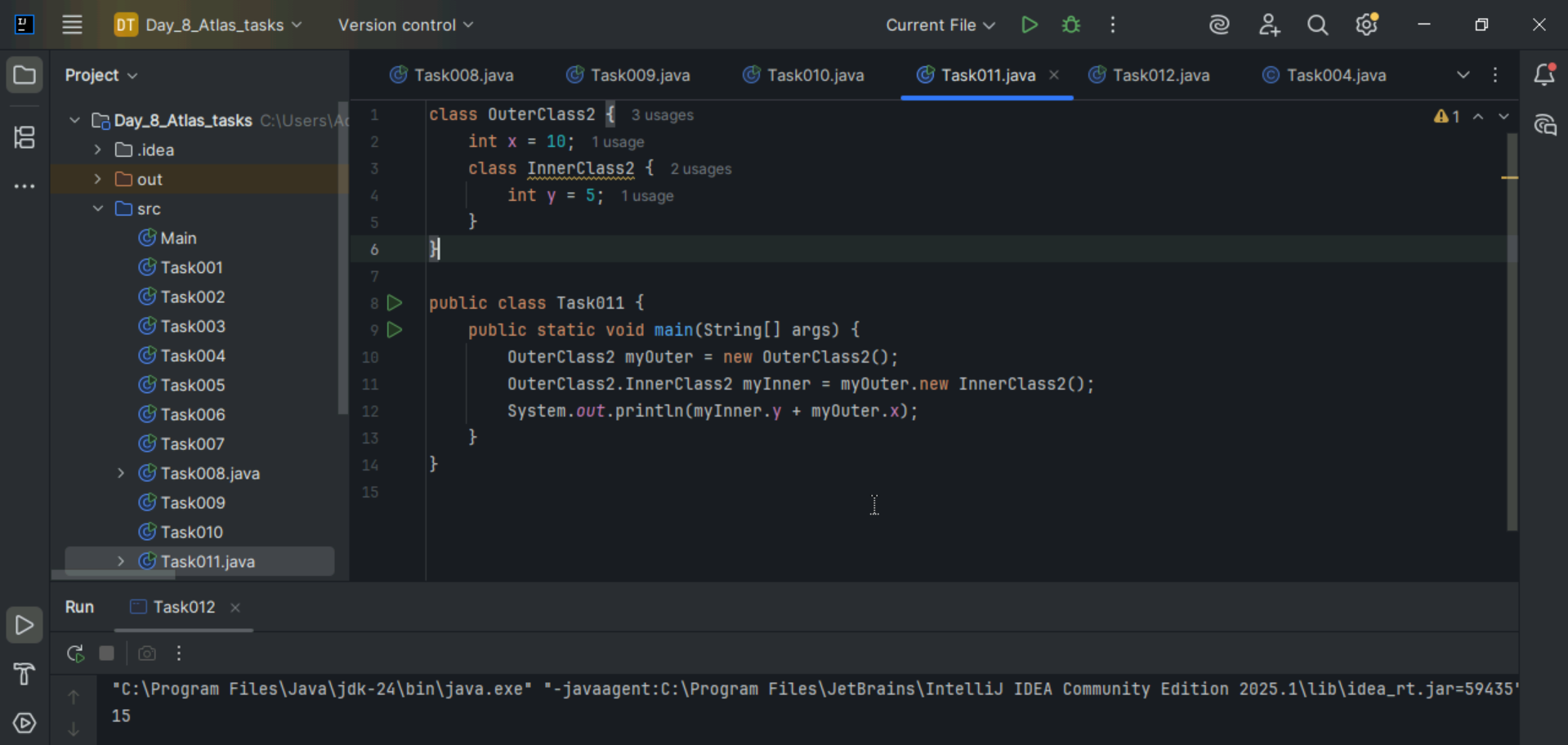


Task 10:

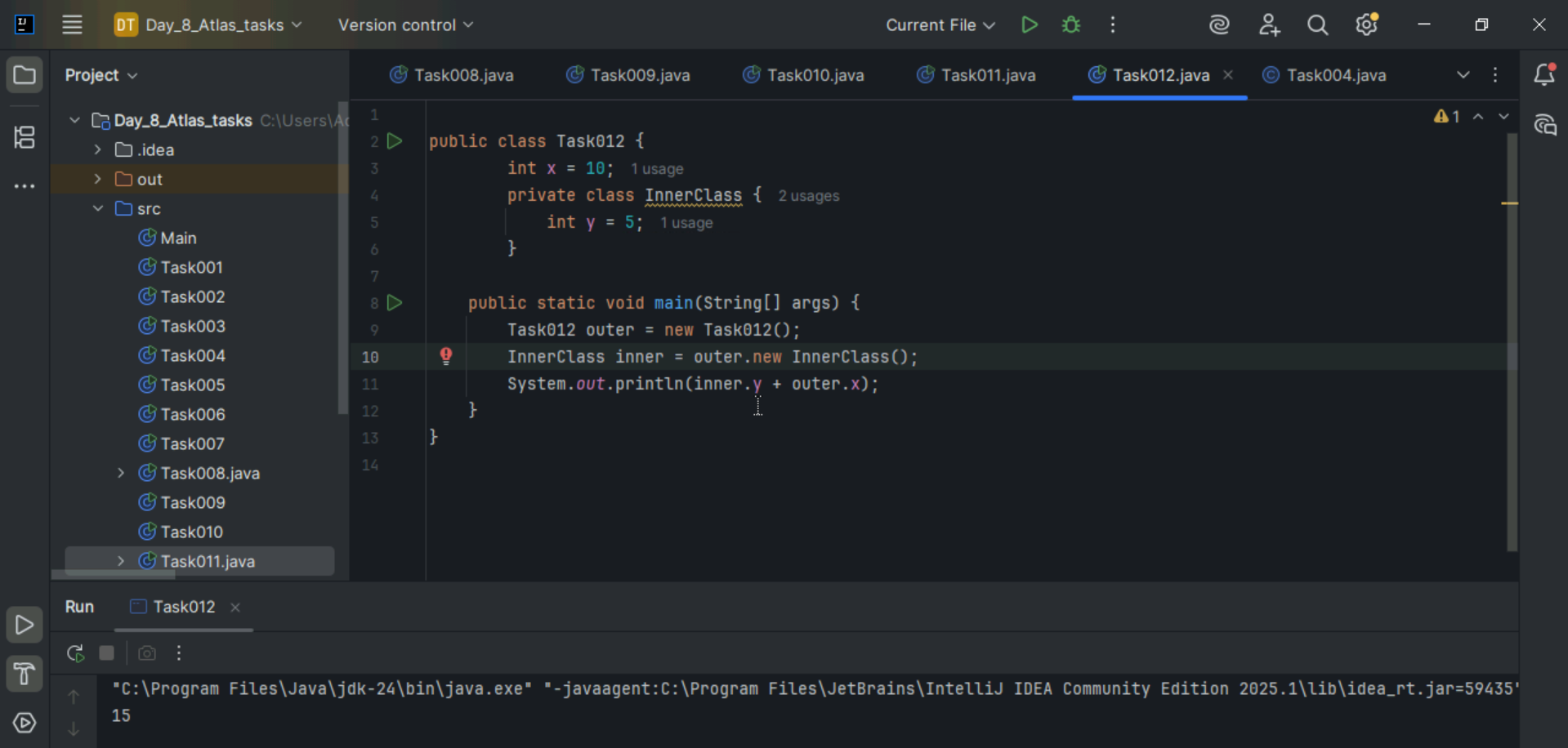




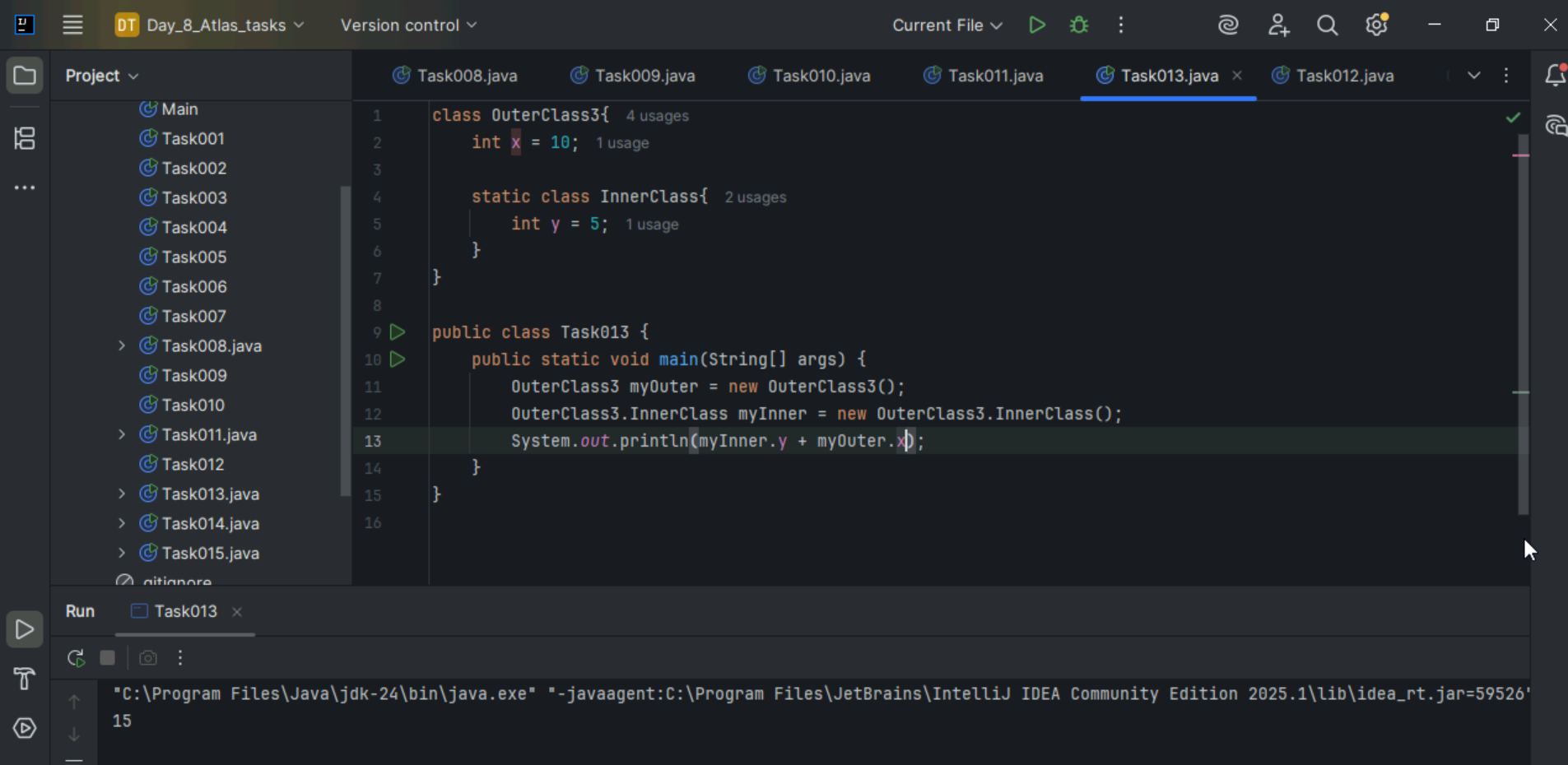
Task 11:



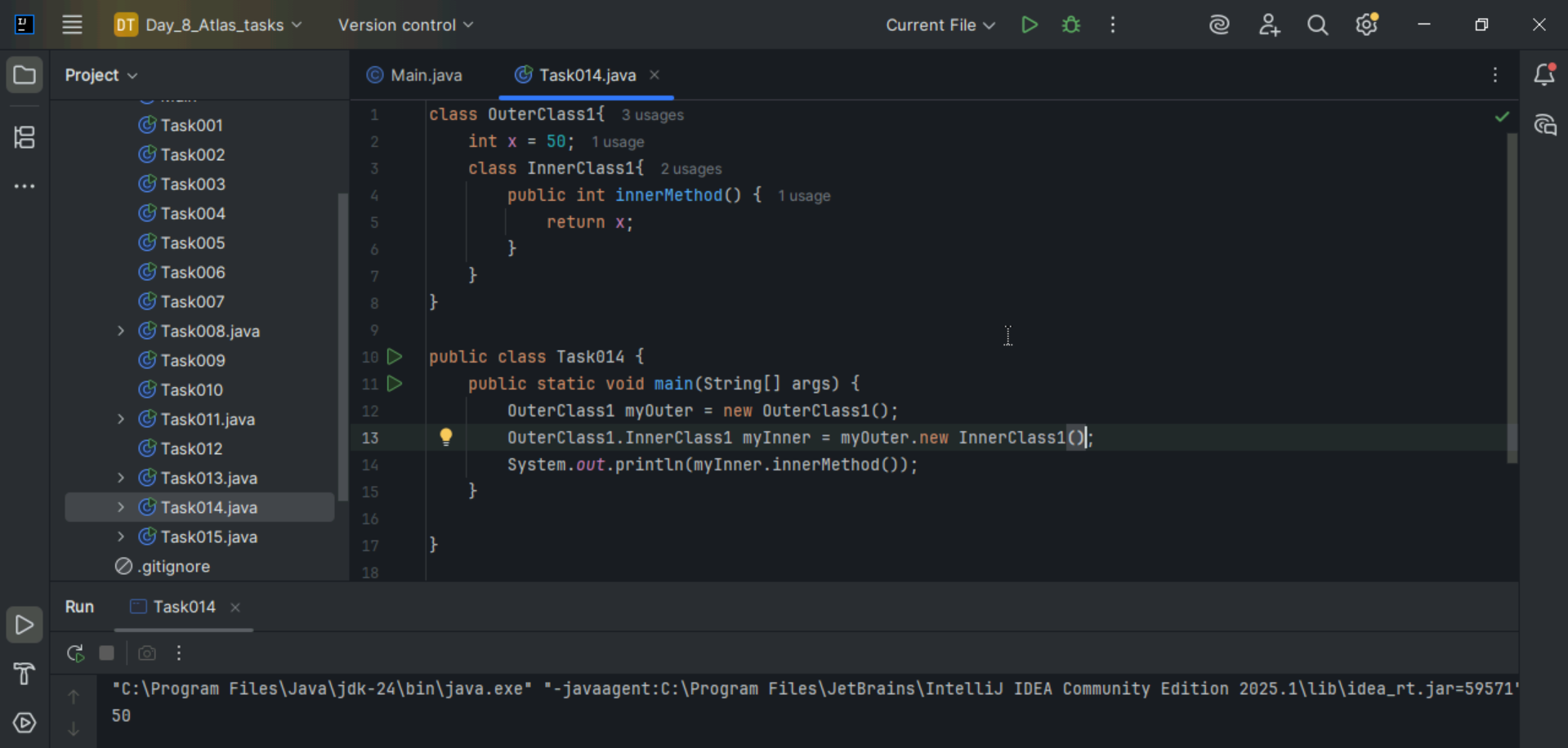
Task 12:



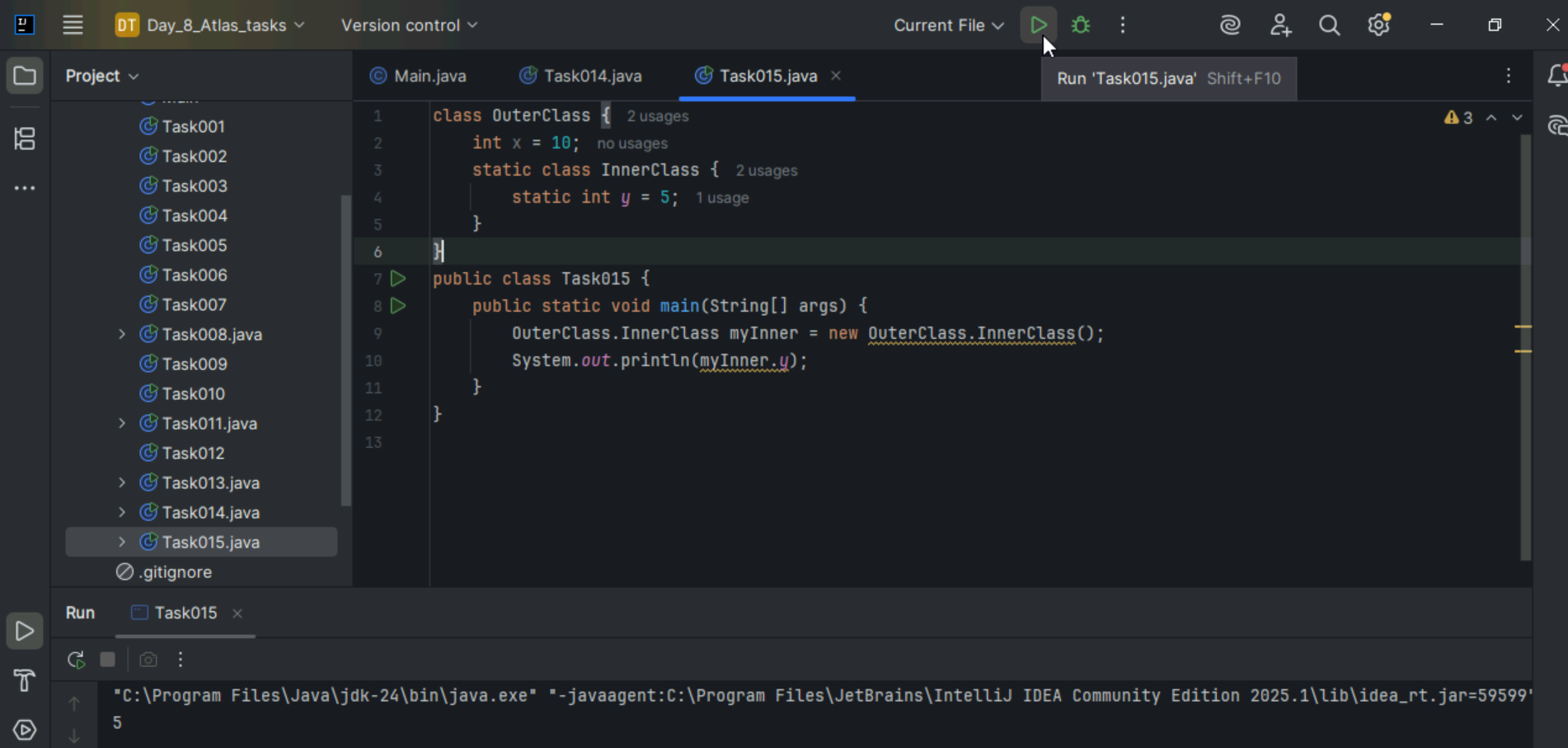
Task 13:



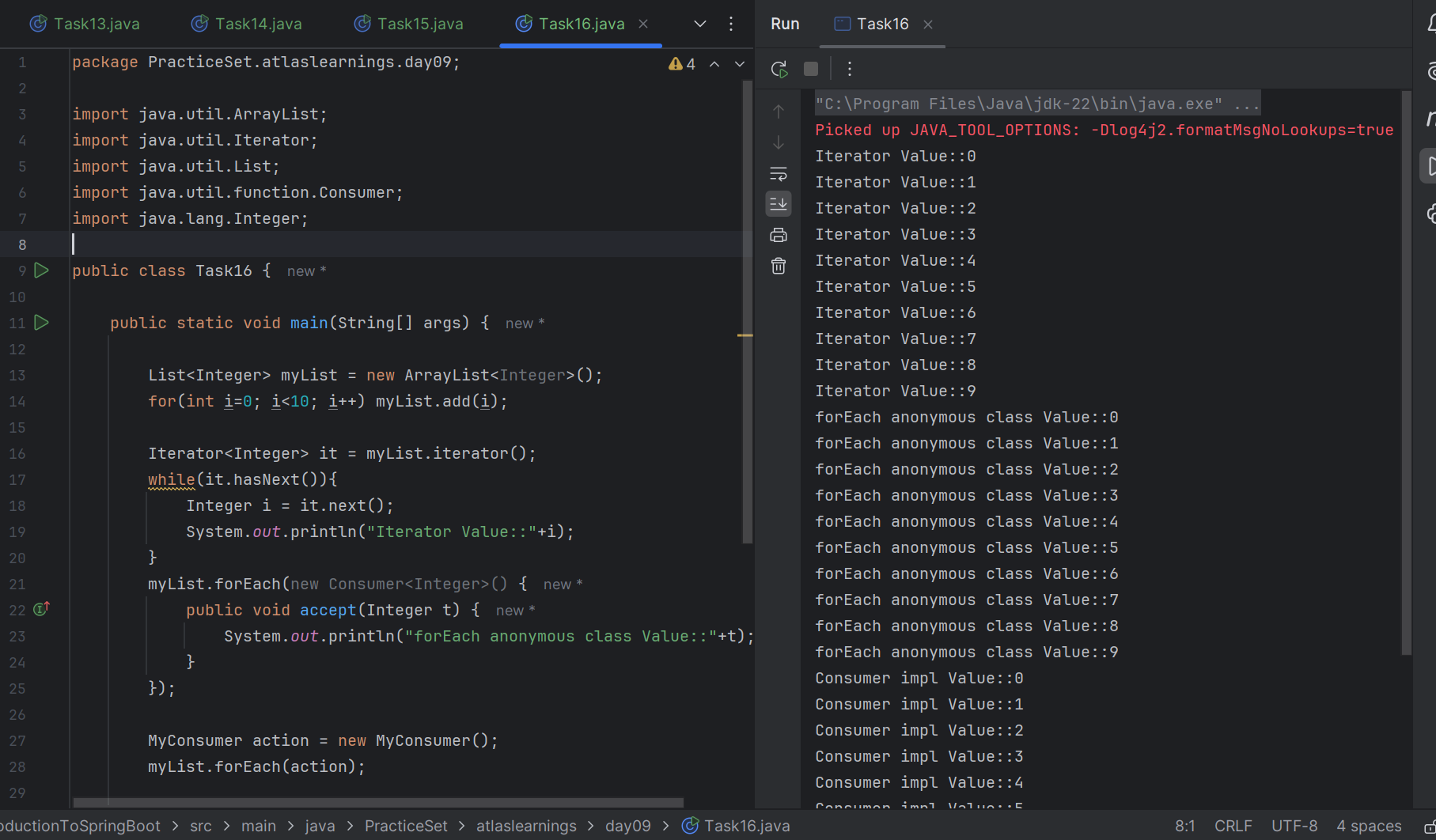
Task 14:

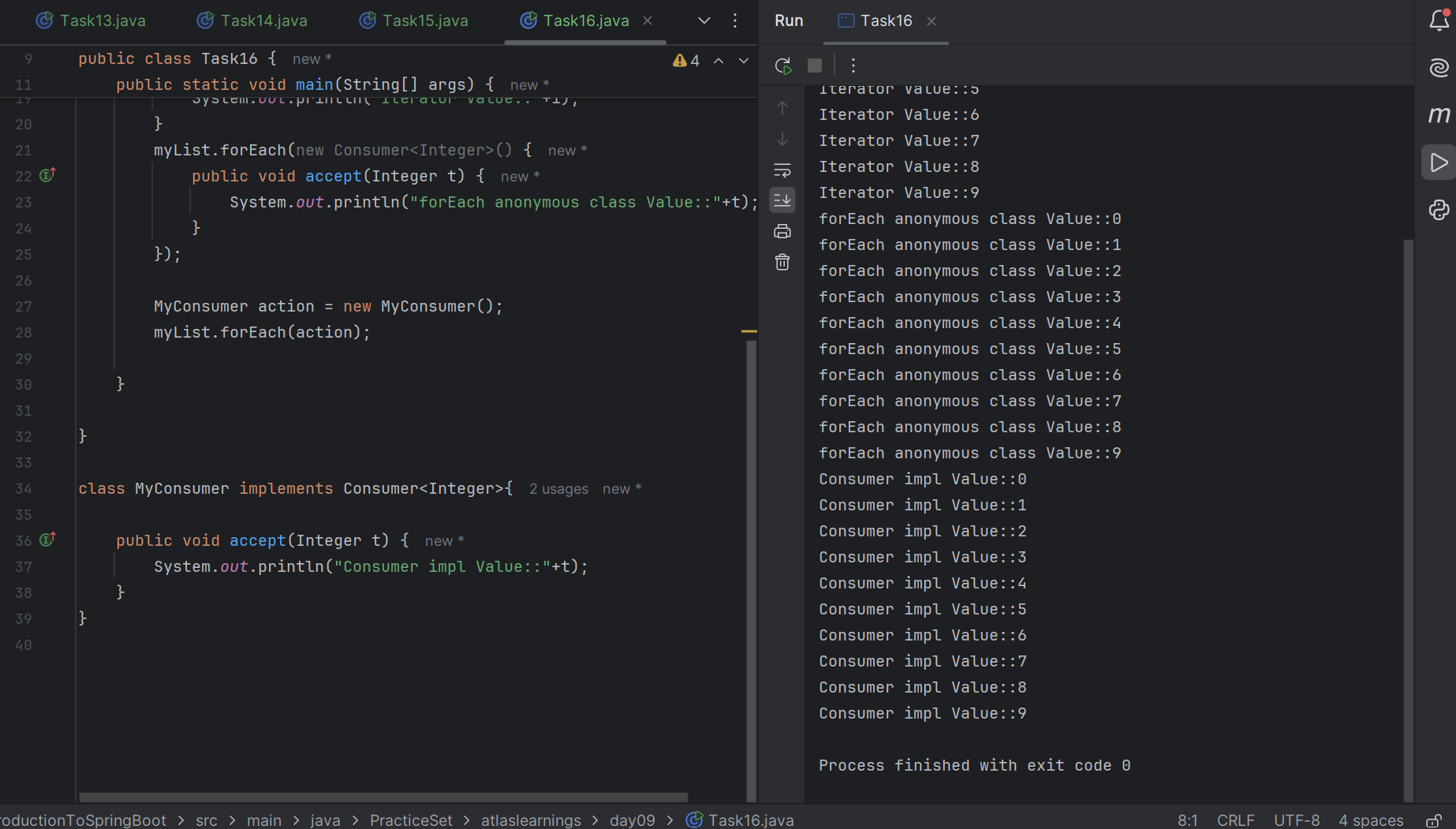


Task 15:



Task 16:





**Task 17**: What are the features of Java 8?

1. Lambda Expressions

Now we can use shorthand for writing some programs

(a, b) -> a + b

1. Default Method in Interface

Interfaces can have method implementations using the default keyword.

interface MyInterface {

default void show() {

System.out.println("Default implementation");

}

}

1. Static method in Interface

interface MyInterface {

static void print() {

System.out.println("Static method in interface");

}

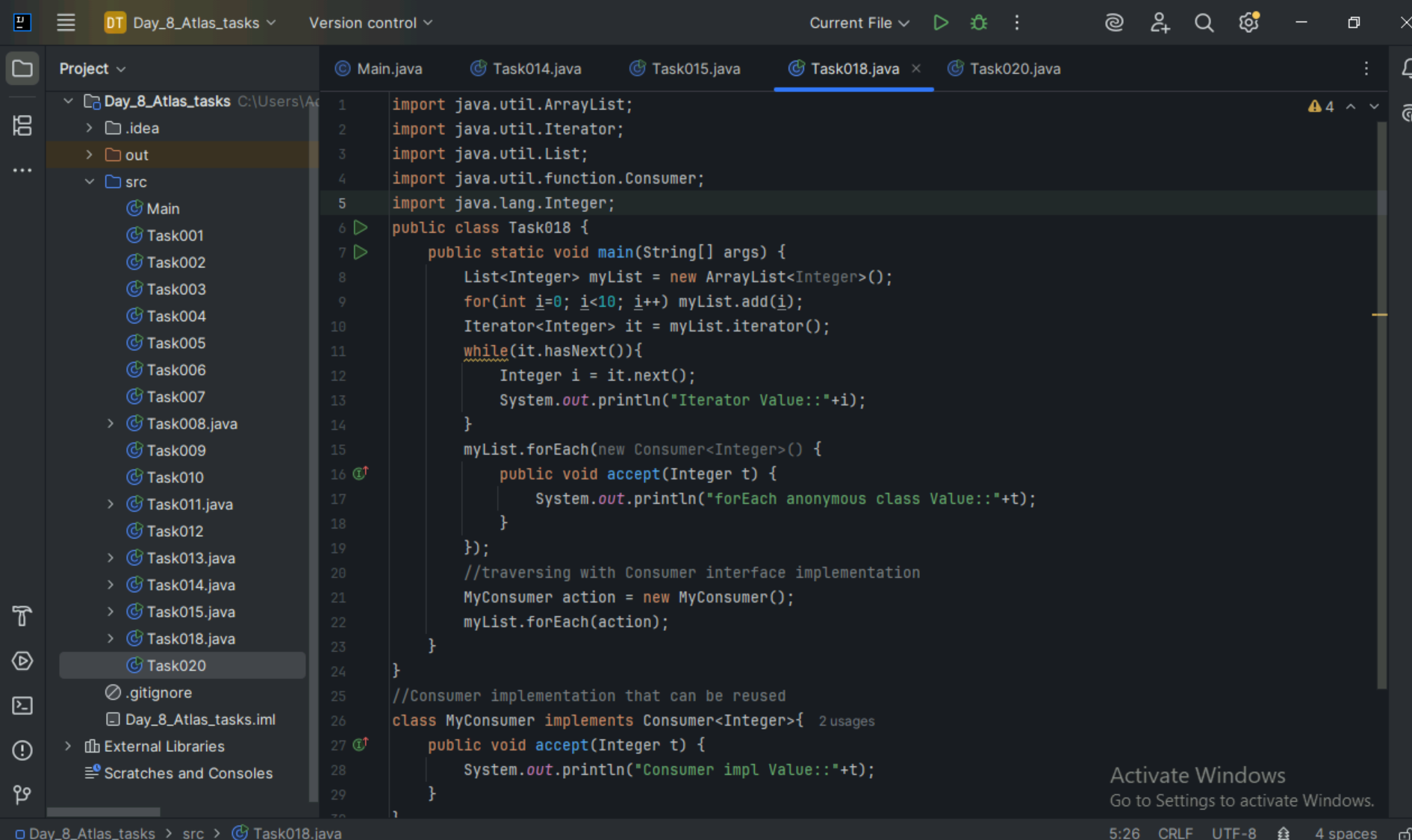
}

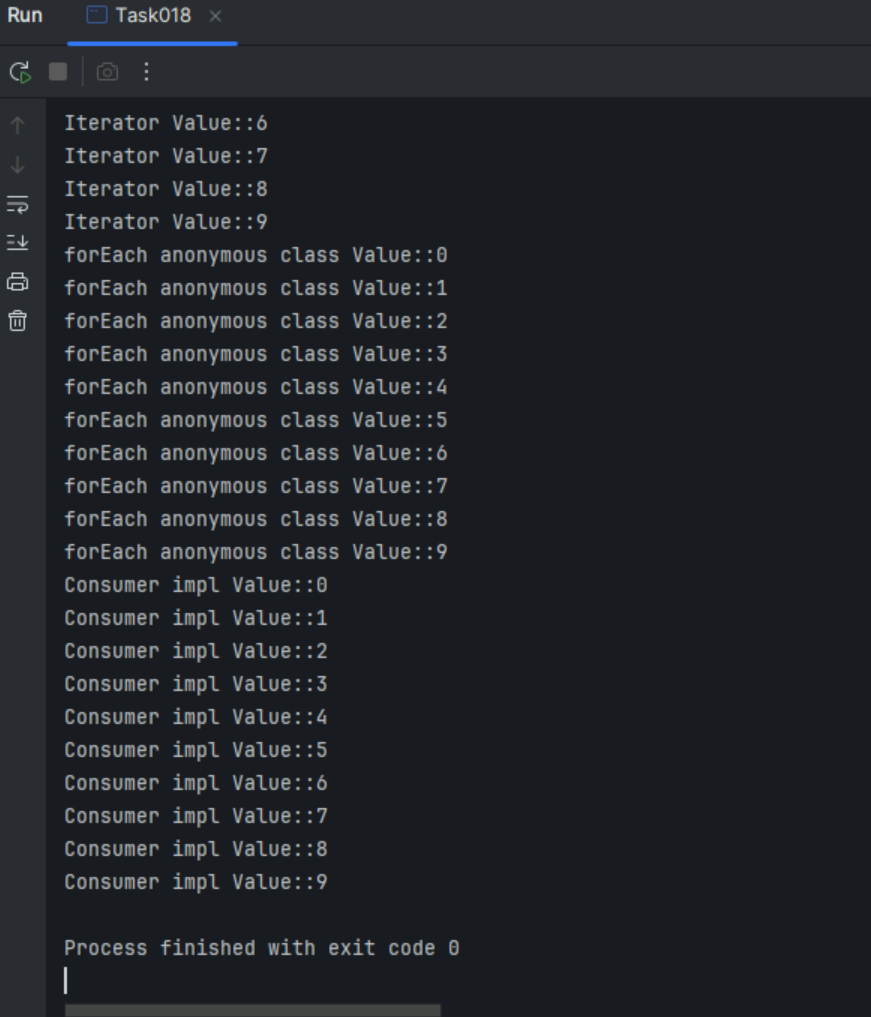
1. Time API
2. Optional Class

Helps avoid NullPointerException by representing optional (nullable) values.

1. Collection API Improvement
2. Concurrency API Improvements
3. forEach() Method in Iterable Interface

**Task 18**:





Task 20:

