Software testing , validation and verification (CSE 338)

Lab Task 3

Submitted to:

Dr. Islam Ahmed Mahmoud elmaddah

Engineer Omar talaat

Made By:

Anas Salah

19P9033

Group 1 Section 1

INTEGRATION TESTING

COFFEE MACHINE

I implemented a simple function that the user uses to order a certain type of coffee.

```
public class CoffeeMachine {

//    function which lets user order coffee
    public String orderCoffee(String type){

        // call stub to check if type is available
        coffeeMachineStub mystub  = new coffeeMachineStub();
        String check = mystub.checkAvailabilityStub(type);
        if (check.equals("Available")){
            return "COFFEE IS BEING PREPARED...";
        }
        else {
            return "COFFEE NOT AVAILABLE";
        }


    }

}
```

Inside this function it calls another function (stub for now) that checks if this type of coffee is available

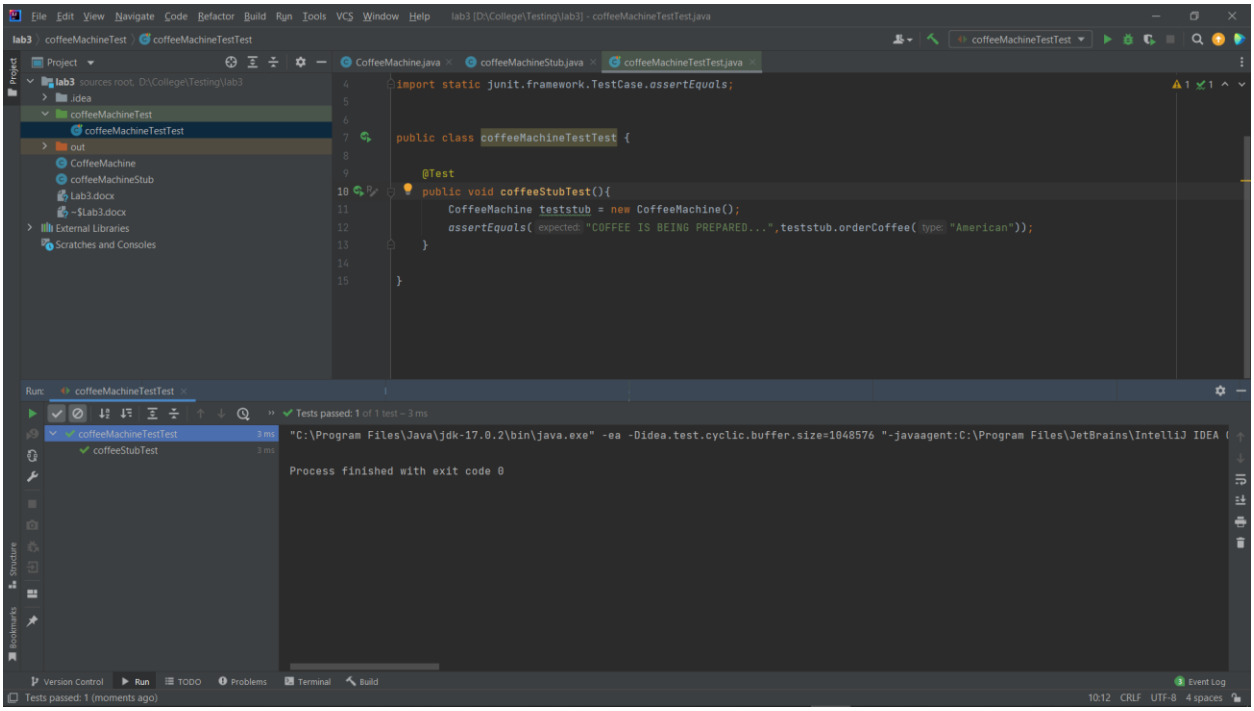
Since the called function is just a stub for now so its implementation would be:

```
public class coffeeMachineStub {


    String[] Available = {"American", "Black", "Turkish"};
    // check for coffee type availability
    public String checkAvailabilityStub(String type){

        return "Available";

    }
```

which simply return Available to the calling function.

I made a test case to insure that this happens

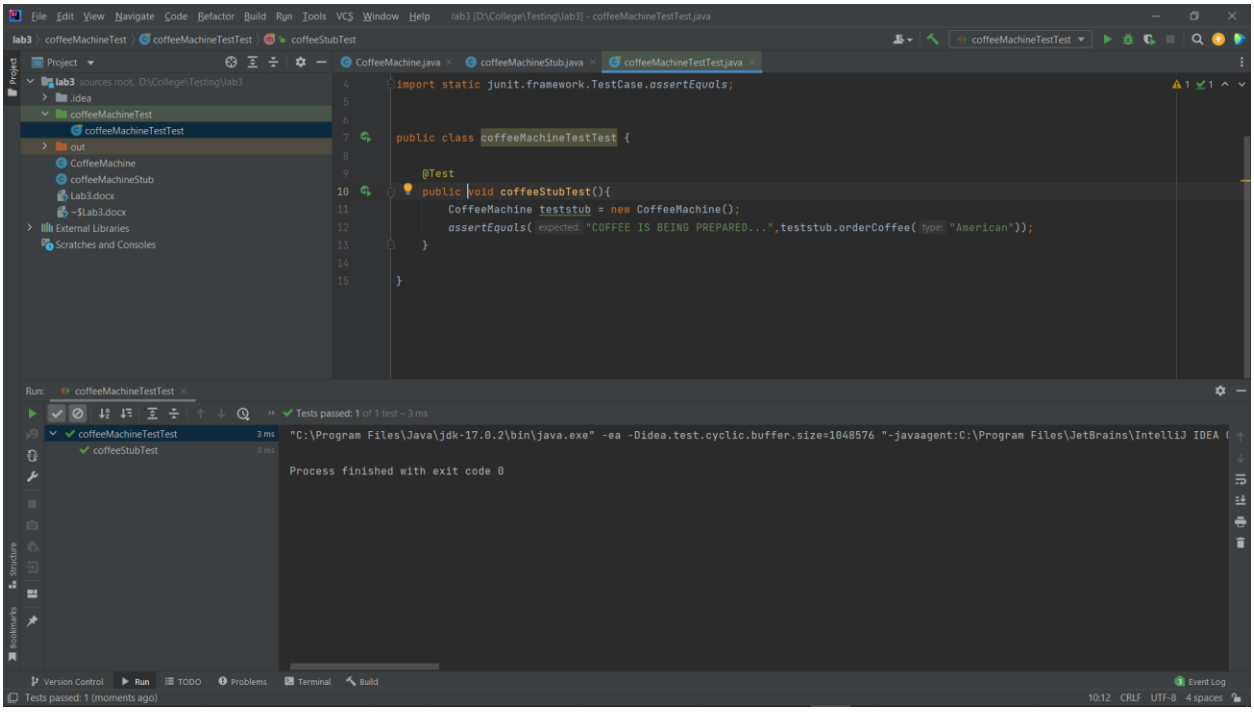
TC:



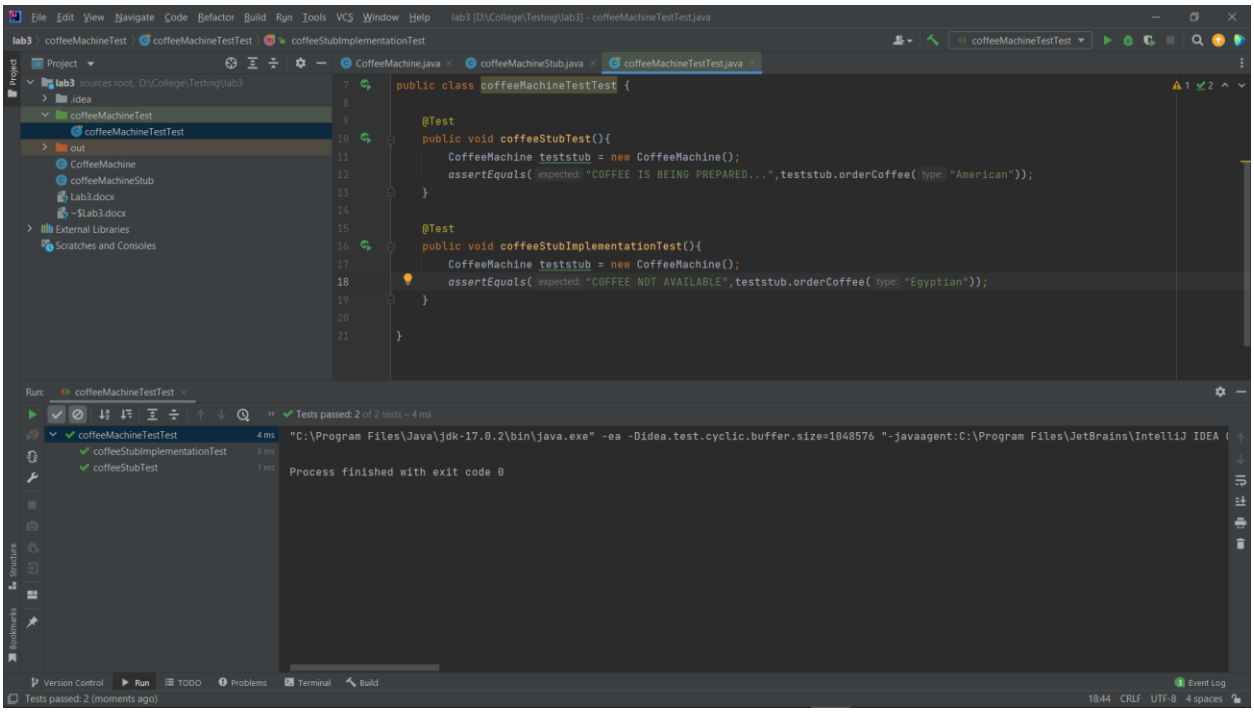
After that we implement the Stub function :

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class coffeeMachineStub {


    String[] Available = new String[]{"American" , "Black" , "Turkish"};
    List<String> Availablelist = new ArrayList<>(Arrays.asList(Available));
    // check for coffee type availability
    public String checkAvailabilityStub(String type){

        if (Availablelist.contains(type)){
         return "Available";
        }
        else{
            return "Not Available";
        }
    }
}
```

we run the test again to check if it works:



We make another test case to check when coffee is not available:



ATM Machine

I implemented a simple withdraw function that tells the user if the requested amount is accepted or not and updates his balance

```
public class ATM {



//    function that lets the user withdraw money
    public String withdraw(int requested){

        // call stub to check if requested amount is available in balance
        ATMStub mystub  = new ATMStub();
        String acceptance = mystub.checkbalance(requested);
        if (acceptance == "Accepted"){
            int newbalance = mystub.balance - requested;
            mystub.balance = mystub.balance - requested;
            return "Withdraw accepted , new balance is " + newbalance;
        }
        else{
            return "invalid amount requested";
        }
```

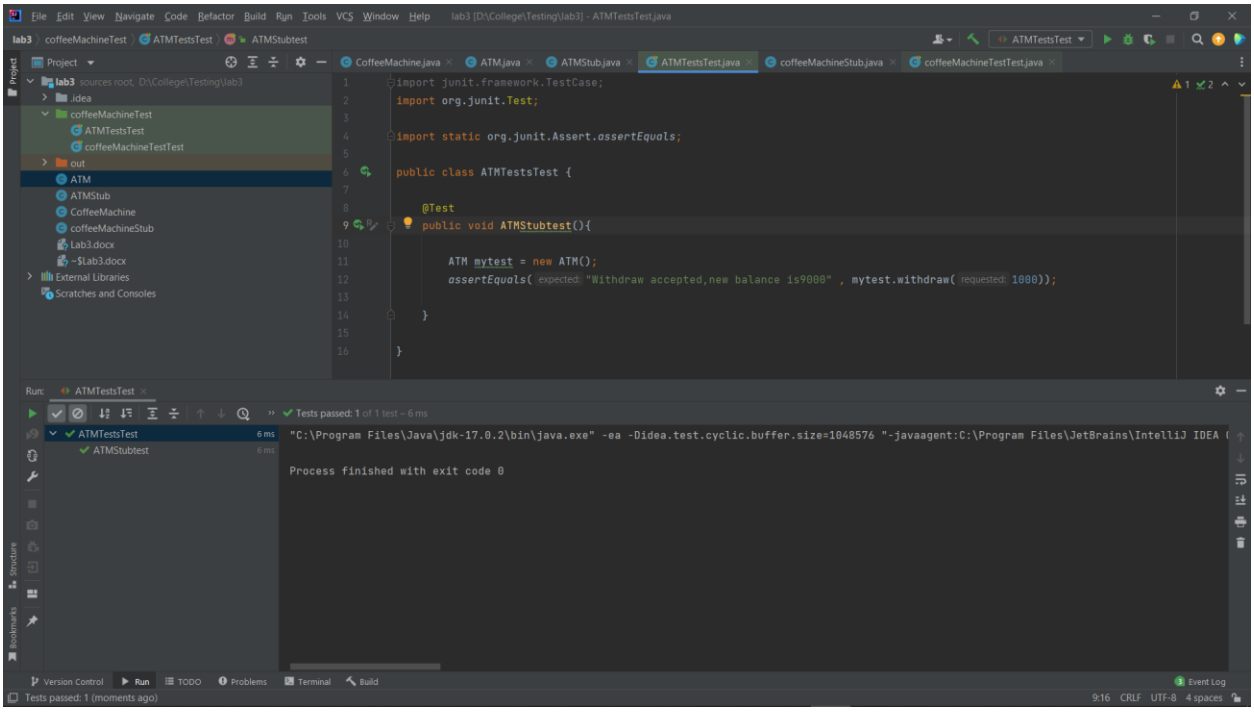
which calls a stub that checks balance

this stub simply return “Accepted” for now

```
public class ATMStub {

    public int balance = 10000;

    public String checkbalance(int requested){

        return "Accepted";

    }
}
```

I made a test case to make sure this connection happens

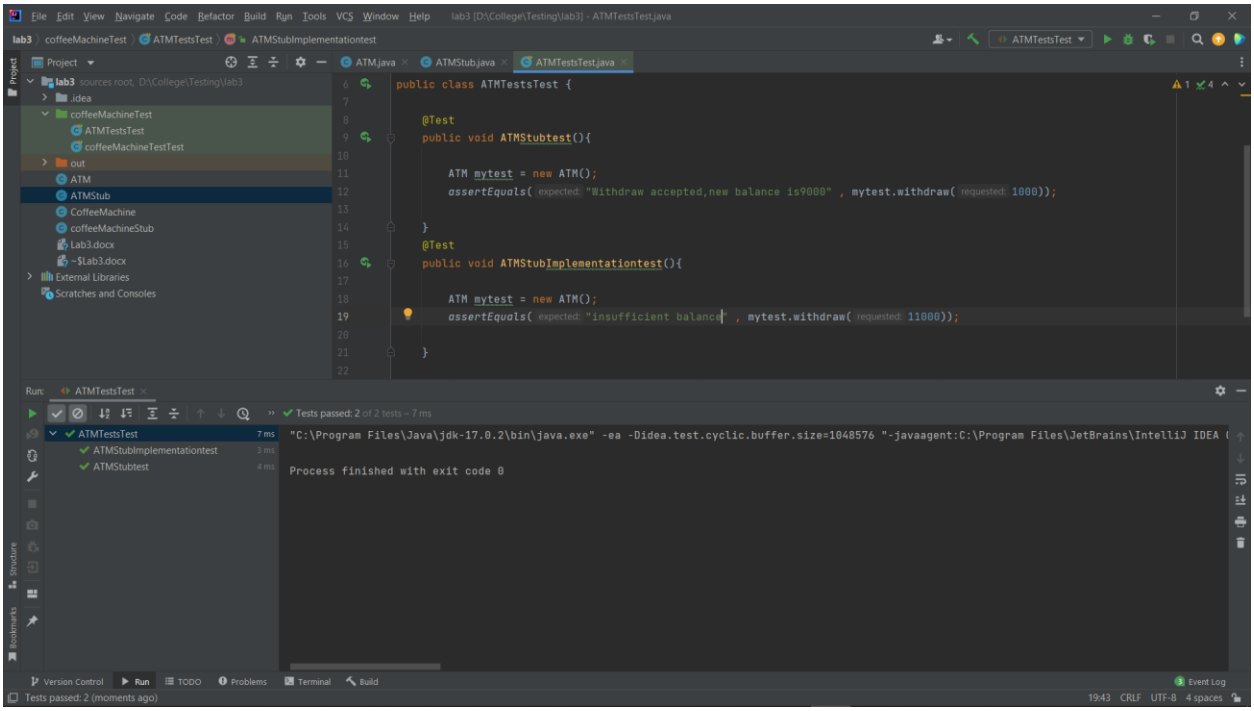
TC:



After that I implemented the stub function:

```
public class ATMStub {

    public int balance = 10000;

    public String checkbalance(int requested){

        if (requested <= balance){
            return "Accepted";

        }
        else{
            return "Insufficient Balance";
        }


    }
}
```

we ran 2 test case to make sure implementation works:



Digital Watch

I will implement a seconds increment function which increment seconds, but when seconds reach 60 then it has to increment minutes and if minutes reached 60 then it has to increment hours and when hours reach 24 it resets watch

```
public class DigitalWatch {

    public int seconds = 0 ;
    public int minutes = 0 ;
    public int hours = 0 ;

    public String incrementSeconds(){

        seconds = seconds + 1;
        // call stub to check if seconds is 60
        boolean secondcheck = checkSecondsStub();
        if (secondcheck == false){
            return "Time is : " + "Hours : " + hours + " " + "Minutes : " +
minutes + " " + "Seconds : " + seconds;
        }
        else{
            seconds = 0;
            return "Time is : " + "Hours : " + hours + " " + "Minutes : " +
minutes + " " + "Seconds : " + seconds;
        }



    }
```

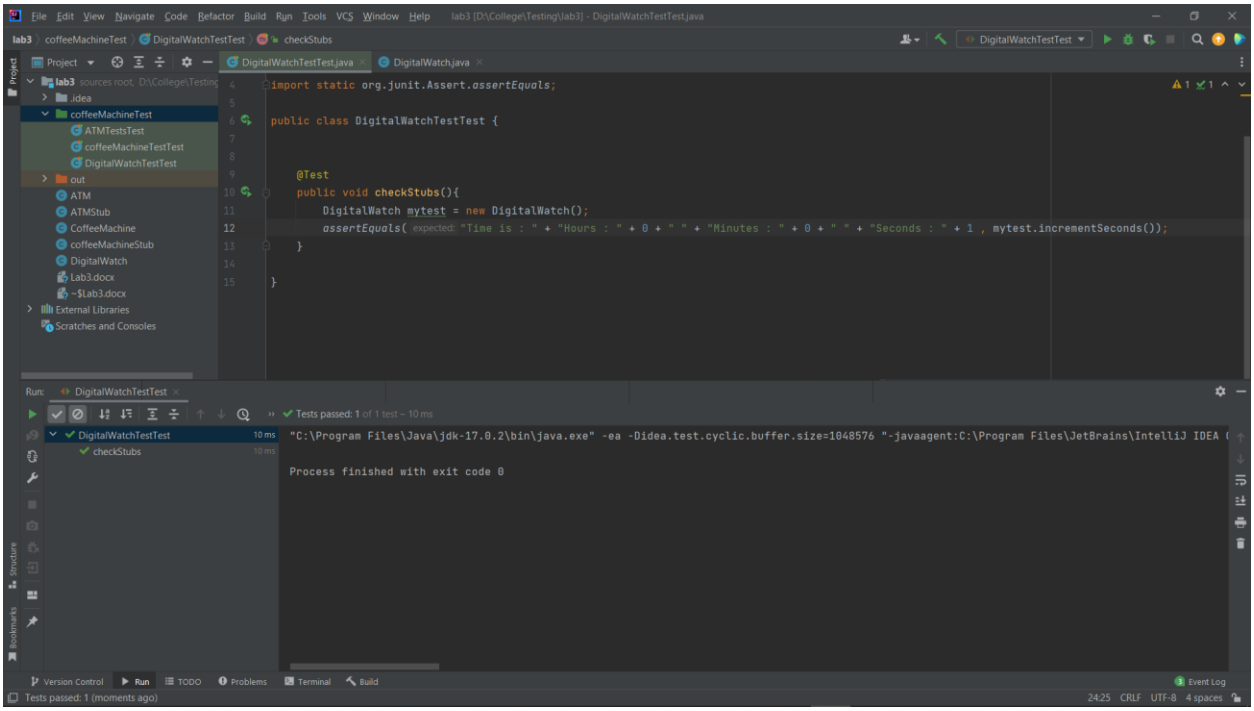
then I make stubs to check if seconds reached 60 or minutes reached 60 or hours reached 24

but for now they return false which means none of these conditions occurred

```
public boolean checkSecondsStub(){
    return checkMinutessStub();


}
public boolean checkMinutessStub(){
    return checkHoursStub();
}
public boolean checkHoursStub(){
    return false;
}
```

we make a test case to make sure these connections take place correctly

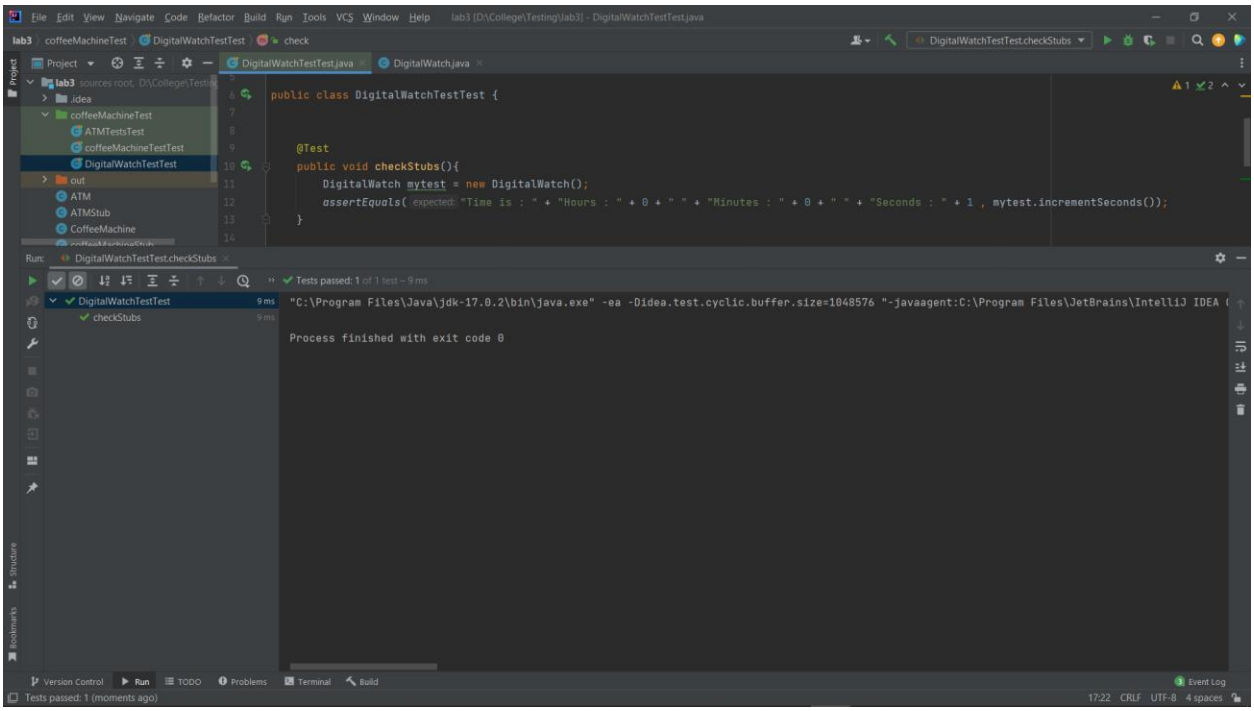
TC :



Then we implement these stubs:

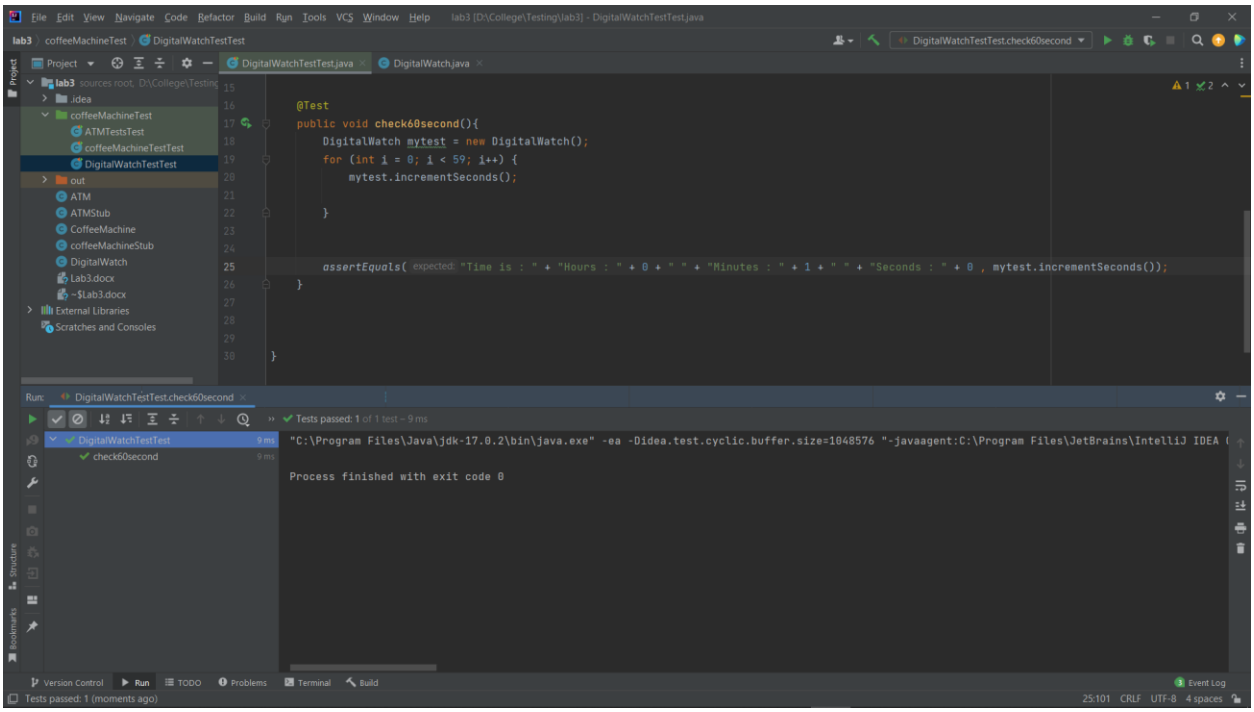
```
public boolean checkSecondsStub(){
    if (seconds == 60){
        minutes++;
        checkMinutessStub();
        return true;
    }
    else {
        return false;
    }

}
public void checkMinutessStub(){
    if (minutes == 60){
        minutes = 0 ;
        hours ++;
        checkHoursStub();
    }


}
public void checkHoursStub(){
    if (hours==24){
        hours=0;
    }
}
```

Then we ran the same test again to make sure same result:



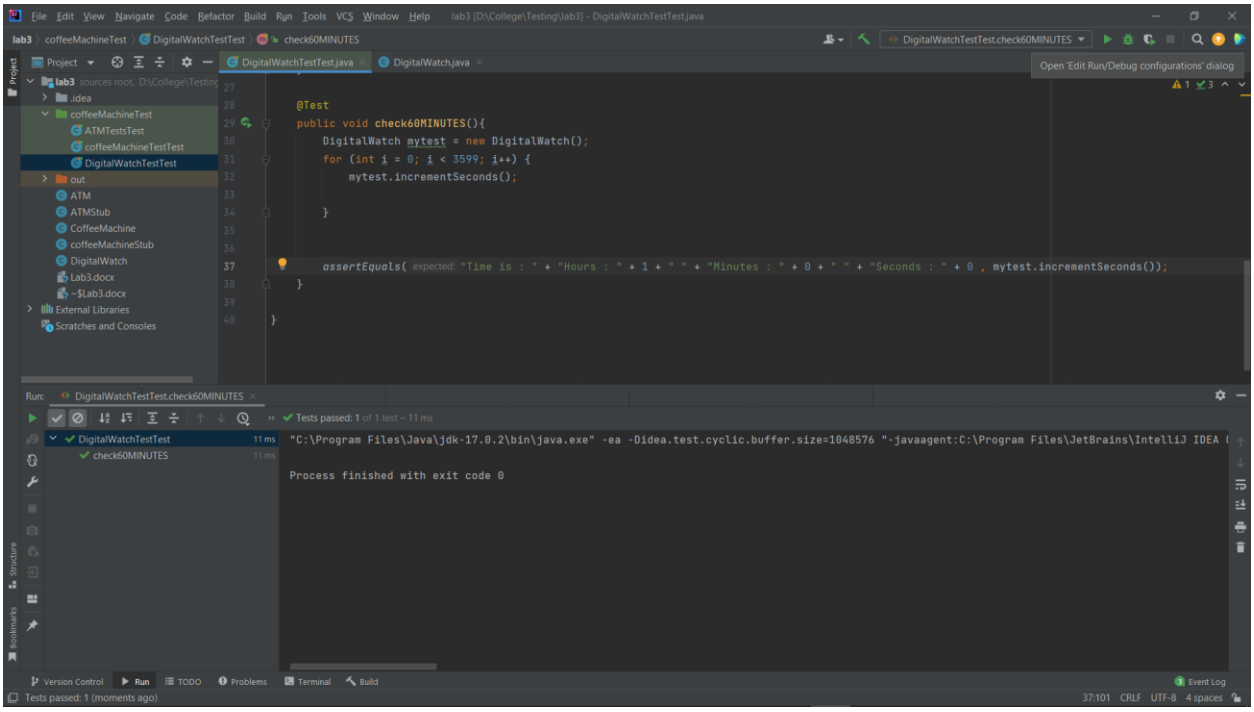
We make another test to check if seconds reached 60 :

TC:



Another test to check when minutes reach 60:

3600 second increments in an hour



Another test to check when hours reach 24 :

86400 second increments in a day

Since it resets a watch after a whole day