Case Tools

NAME: KOWSYA MUTKUNDU

ROLL NO: 25

Junit

Program 1: Swapping 2 numbers

public class Swapping {

public double[] swap(double num1, double num2) {

double temp = num1;

num1 = num2;

num2 = temp;

return new double[]{num1, num2};

}

};

**Junit test code**:

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class SwappingTest {

@Test

void test() {

// fail("Not yet implemented");

Swapping s = new Swapping();

double result[] = s.swap(5, 3);

// int[]result = new int[]{90,20};

assertArrayEquals(new double[] {3, 5}, result);

// assertArrayEquals(s.swap(20,90),result);

}

}

Program 2: Concatenate and Multiplication

public class Jan04 {

public String concatenateStrings(String str1, String str2) {

if (str1 == null) str1 = "";

if (str2 == null) str2 = "";

return str1.concat(str2);

};

public double multiply(double num1, double num2) {

return num1 \* num2;

}

};

**Junit test code:**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class concat\_Jan04Test {

@Test

void test() {

// fail("Not yet implemented");

Jan04 j = new Jan04();

String result = j.concatenateStrings("one", "two");

assertEquals("onetwo", result);

}

}

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class calc\_Jan04TestTest {

@Test

void test() {

// fail("Not yet implemented");

Jan04 j = new Jan04();

double result = j.multiply(4, 8);

assertEquals(32, result);

}

}

Program 3: Sorting array

import java.util.Arrays;

public class ArraySorting {

int[] sort(int[] arr) {

if (arr == null) return null;

Arrays.sort(arr);

return arr;

}

}

**Junit test code:**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class ArraySortingTest {

@Test

void test() {

// fail("Not yet implemented");

ArraySorting ar = new ArraySorting();

assertArrayEquals(new int[]{1, 2, 3, 4, 5}, ar.sort(new int[]{2, 3, 1, 5, 4}));

}

}

Program 4: Factorial of a number

public class Factorial {

int factorial(int n) {

if (n == 0) return 1;

else return n \* factorial(n - 1);

}

}

Junit test code:

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class FactorialTest {

@Test

void test() {

// fail("Not yet implemented");

Factorial f = new Factorial();

assertEquals(1, f.factorial(1));

// assertEquals(f.factorial(5),120));

}

}

Program 5: Fibonacci series

public class Fibonacci {

int fibonacci(int n) {

// if (n < 2) { return n;

// else return fibonacci(n - 1) + fibonacci(n - 2);

int num1 = 0, num2 = 1, k = 0, counter = 0;

while (counter < n) {

int num3 = num1 + num2;

num1 = num2;

num2 = num3;

counter++;

k = Integer.valueOf(String.valueOf(k) + String.valueOf(num3));

}

System.out.println(k);

return k;

}

}

**Junit test code:**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class FibonnaciTest {

@Test

void test() {

// fail("Not yet implemented");

Fibonacci f = new Fibonacci();

// assertEquals(5, f.fibonacci(5));

assertEquals(12358, f.fibonacci(5));

}

}

Program 6: Reverse n digit number

public class Reverse {

int reverse (int number ) {

int reversedNumber = 0;

while (number != 0) {

reversedNumber = reversedNumber \* 10 + number % 10;

number /= 10;

};

return reversedNumber;

}

};

**Junit test code:**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

class ReverseTest {

@Test

void test() {

// fail("Not yet implemented");

Reverse r = new Reverse();

assertEquals(r.reverse(12345), 54321);

}

}

Program 7: Calculater with basic operations

Public class Calculator {

Public static int add(int numl,int num2)

{

Return num1 + num2;

}

Public static int sub(int numl, int num2)

{

Return numl- num2;

}

public static int mul(int numl, int num2)

{

Return numl\* num2;

}

Public static int div(int numl, int num2)

{

Return numl / num2;

}

}

**Junit test code:**

Import static org.junit.Assert.\*;

Public class CalculatorTest (

@Test

Public void test() {

Int numl4;

Int num2 21

Int resultl Calculator.add(numl, num2);

Int result2= Calculator.sub(numl, num2);

Int result3 = Calculator.mul(num1, num2); int result4 Calculator.div(num1, num2);

assertEquals(6,resultl); assertEquals(2,result2);

assertEquals(8,result3);

assertEquals(2,result4);

}

}

GITHUB

* Configuration – username, useremail, list

Student@DBC-00 MINGW64 ~ (master)

$ git config --global user.name"kowsya"

Student@DBC-00 MINGW64 ~ (master)

$ git config --global user.email kowsyamutkundu25@gmail.com

Student@DBC-00 MINGW64 ~ (master)

$ git config --list

diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=openssl

http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager-core

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=master

user.name=Aaron

user.email=kowsyamutkundu25@gmail.com

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

remote.origin.url=https://github.com/aarxn74/ap3.git

remote.origin.fetch=+refs/heads/\*:refs/remotes/origin/\*

Student@DBC-00 MINGW64 ~ (master)

$ pwd

/c/Users/Student

Student@DBC-00 MINGW64 ~ (master)

$ mkdir secondrep

* Navigation to working directory

Student@DBC-00 MINGW64 ~ (master)

$ cd secondrep

Student@DBC-00 MINGW64 ~/secondrep (master)

$ pwd

/c/Users/Student/secondrep

Student@DBC-00 MINGW64 ~/secondrep (master)

$ mkdir list

Student@DBC-00 MINGW64 ~/secondrep (master)

$ cd list

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ pwd

/c/Users/Student/secondrep/list

* Create files using git commands

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ touch data.txt

* Initialization to GIT

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git init

Initialized empty Git repository in C:/Users/Student/secondrep/list/.git/

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ touch numbers.txt

* Editor

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ notepad numbers.txt

* Check status

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

data.txt

numbers.txt

nothing added to commit but untracked files present (use "git add" to track)

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: data.txt

new file: numbers.txt

* To commit

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git commit -m "numbers committed"

[master (root-commit) 691a53d] numbers committed

2 files changed, 1 insertion(+)

create mode 100644 data.txt

create mode 100644 numbers.txt

* To connect to remote github server

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git remote add origin https://github.com/Kowsya/secondrep.git

* To verify connection

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git remote -v

origin https://github.com/Kowsya/secondrep.git (fetch)

origin https://github.com/Kowsya/secondrep.git (push)

* Push command

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git push -u origin master

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (4/4), 264 bytes | 264.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/Kowsya/secondrep.git

\* [new branch] master -> master

branch 'master' set up to track 'origin/master'.

* Create new branch

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git branch labl

* To switch between branches

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git checkout labl

Switched to branch 'labl'

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ touch test3.txt

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ git status

On branch labl

Untracked files:

(use "git add <file>..." to include in what will be committed)

test3.txt

nothing added to commit but untracked files present (use "git add" to track)

* Add file to staging area to commit after modification

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ git add .

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ git commit -m "committed"

[labl bbd455d] committed

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 test3.txt

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ git push -u origin labl

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (2/2), 271 bytes | 271.00 KiB/s, done.

Total 2 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

remote: Create a pull request for 'labl' on GitHub by visiting:

remote: https://github.com/Kowsya/secondrep/pull/new/labl

remote:

To https://github.com/Kowsya/secondrep.git

\* [new branch] labl -> labl

branch 'labl' set up to track 'origin/labl'.

* To switch between branches

Student@DBC-00 MINGW64 ~/secondrep/list (labl)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

* To merge branches (make sure you are in master repository)

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git merge labl

Updating 691a53d..bbd455d

Fast-forward

test3.txt | 0

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 test3.txt

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git push -u origin master

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/Kowsya/secondrep.git

691a53d..bbd455d master -> master

branch 'master' set up to track 'origin/master'.

* To delete branch

Student@DBC-00 MINGW64 ~/secondrep/list (master)

$ git branch -d labl

Deleted branch labl (was bbd455d).

* To delete branch on github

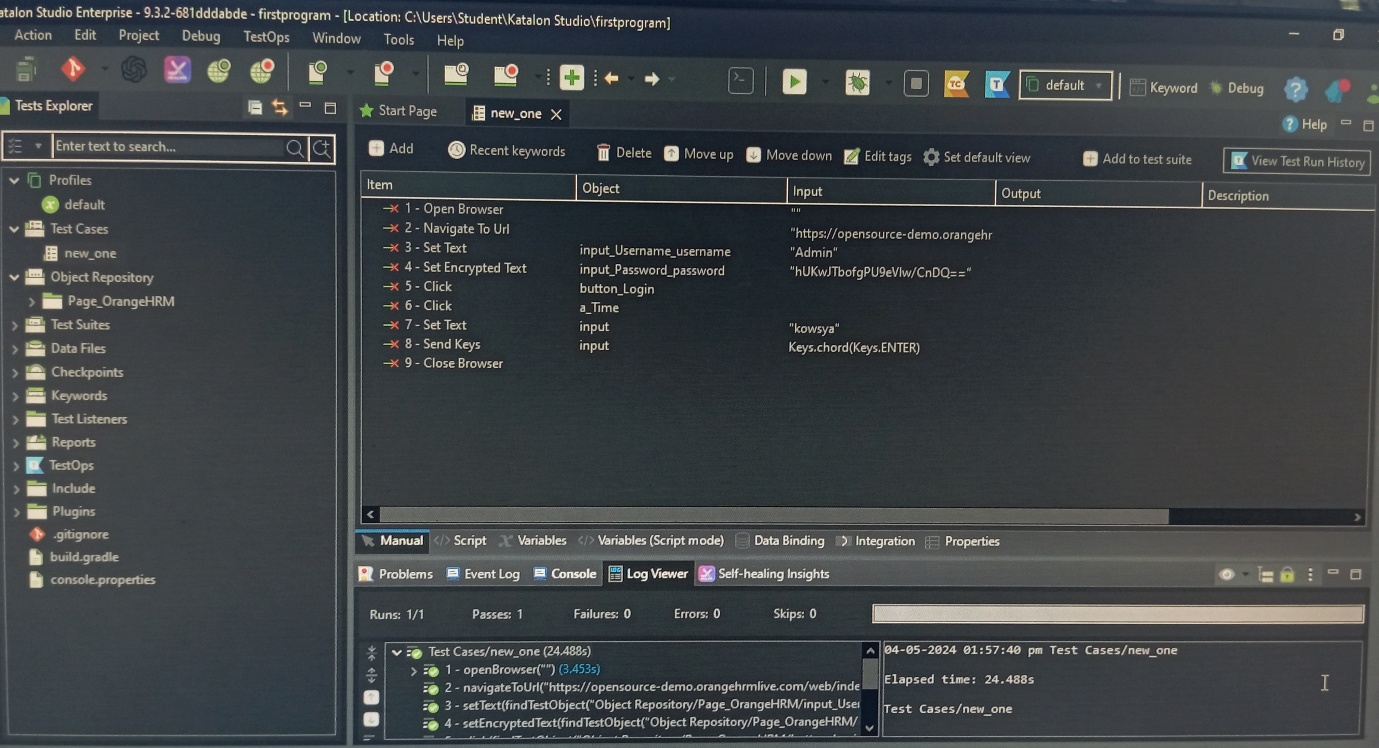
Student@DBC-00 MINGW64 ~/secondrep/list (master)

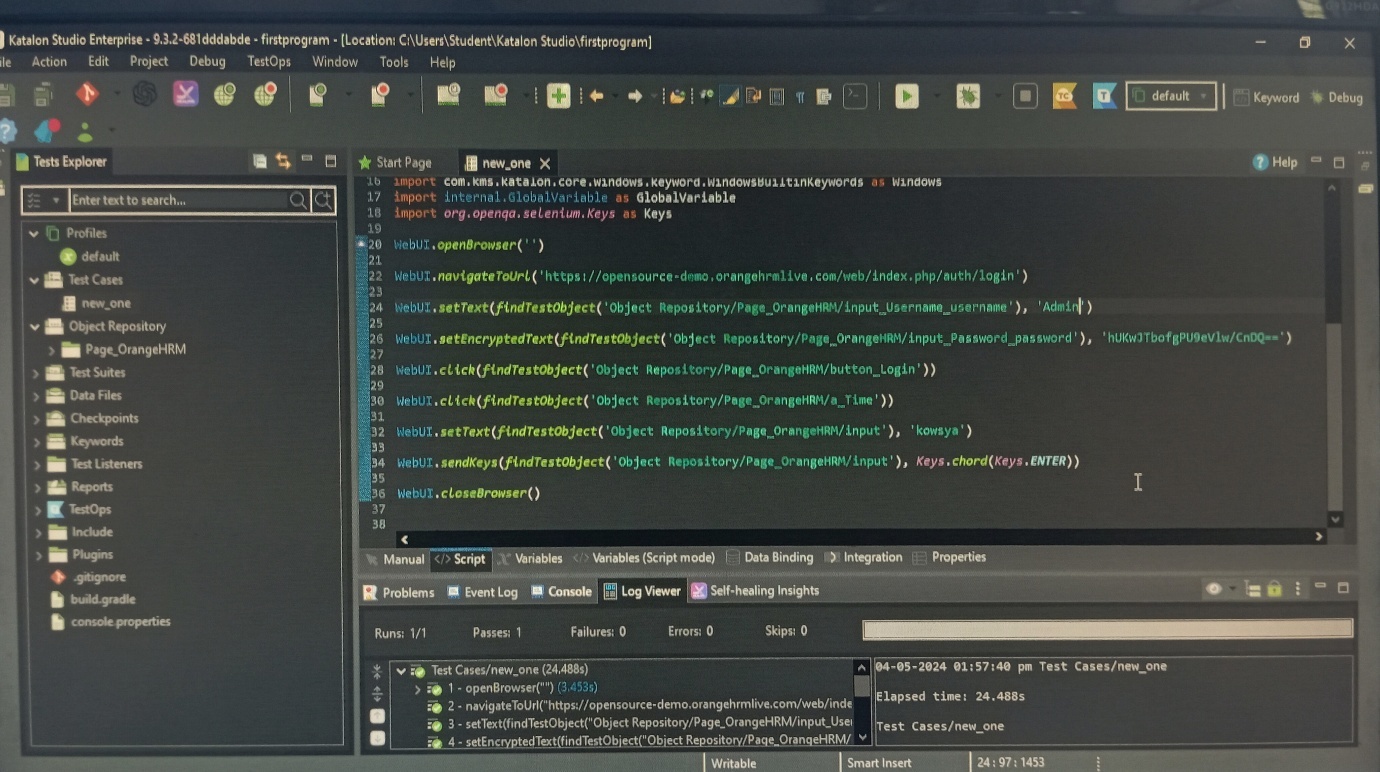
$ git push origin --delete labl

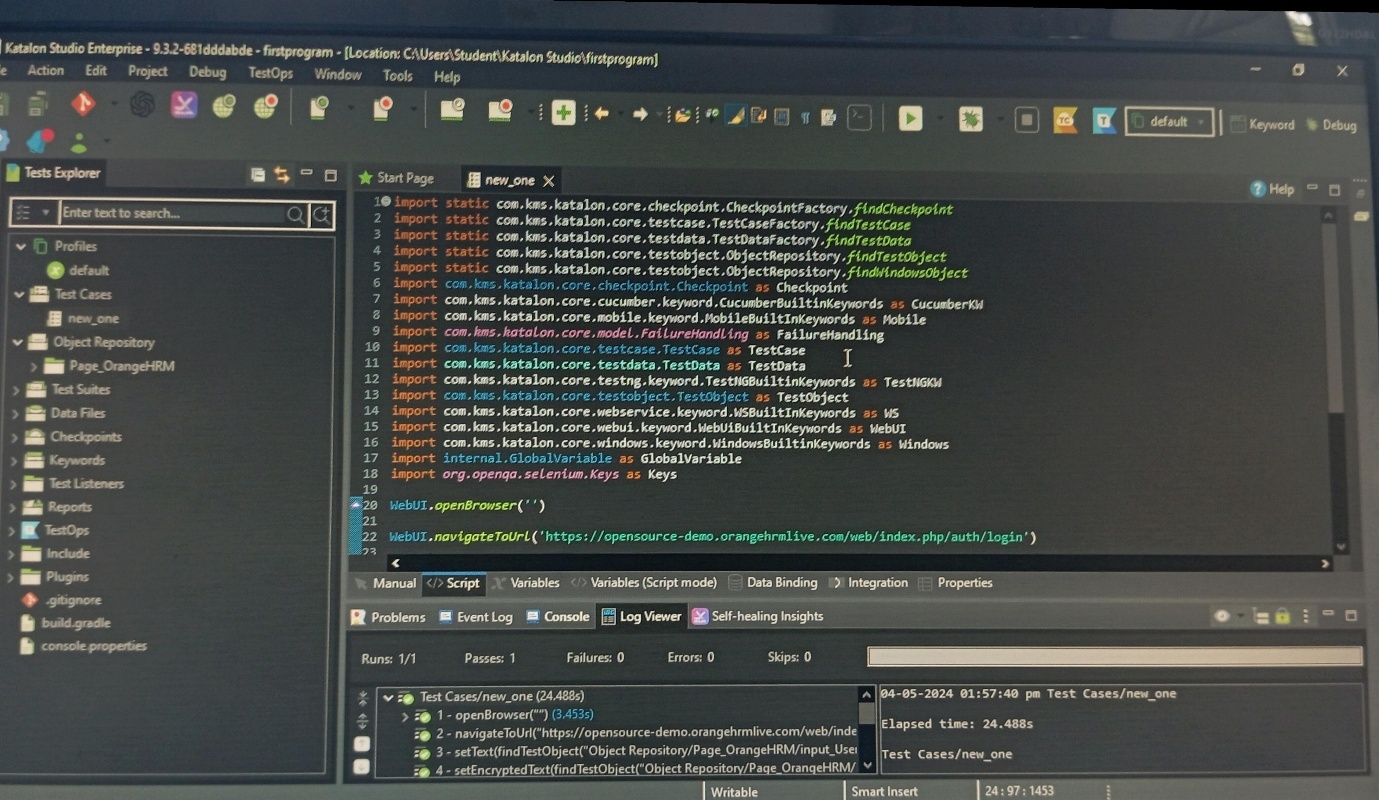
To https://github.com/Kowsya/secondrep.git

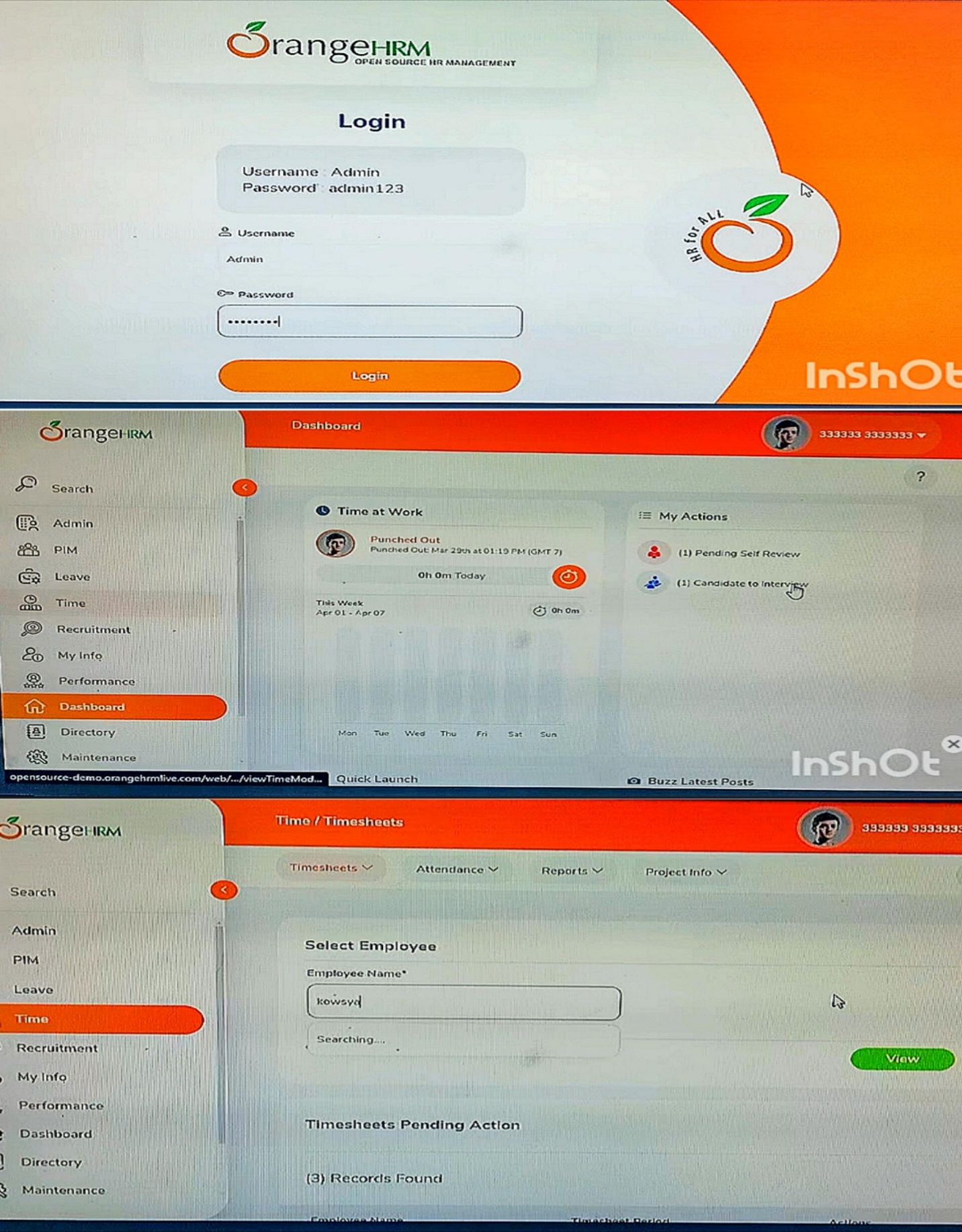
- [deleted] labl

Katalon Studio

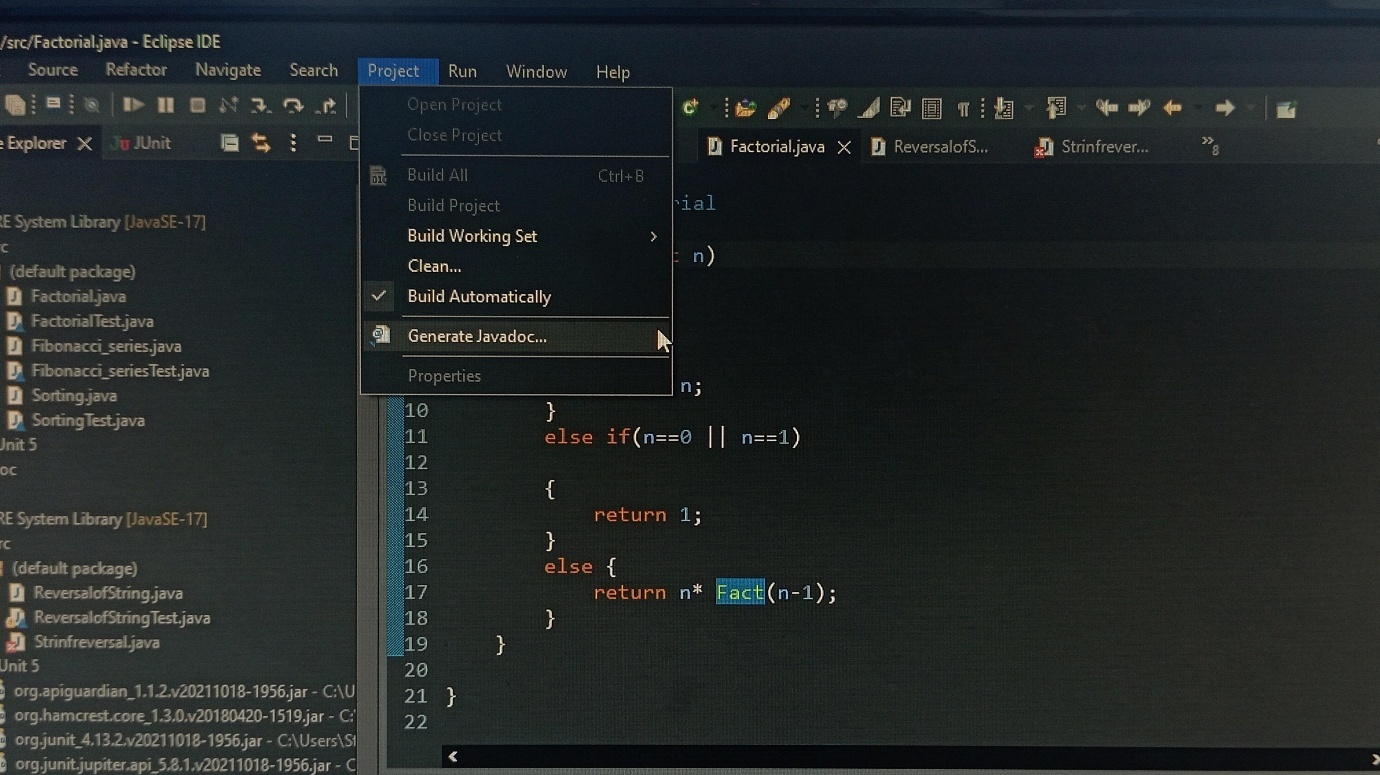
• Manual Program

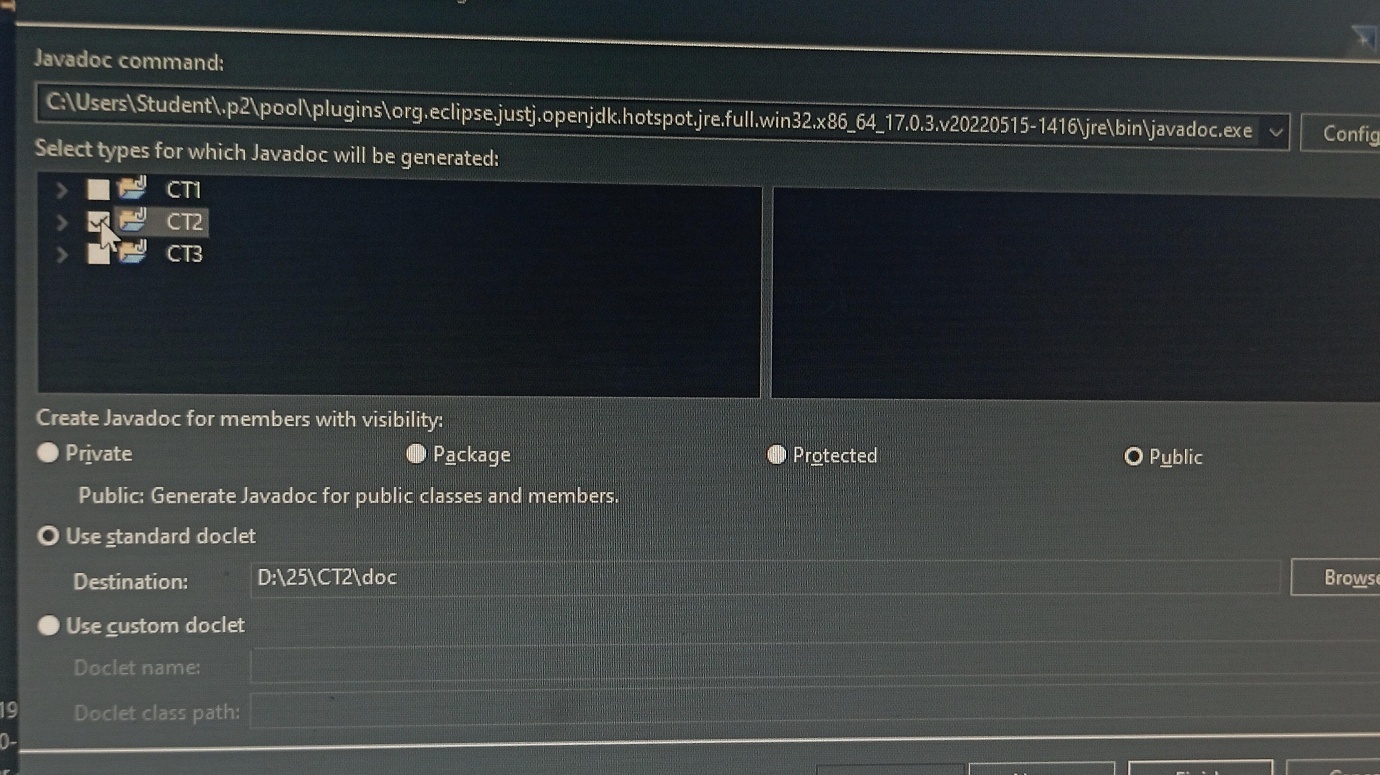
• Scripted Program

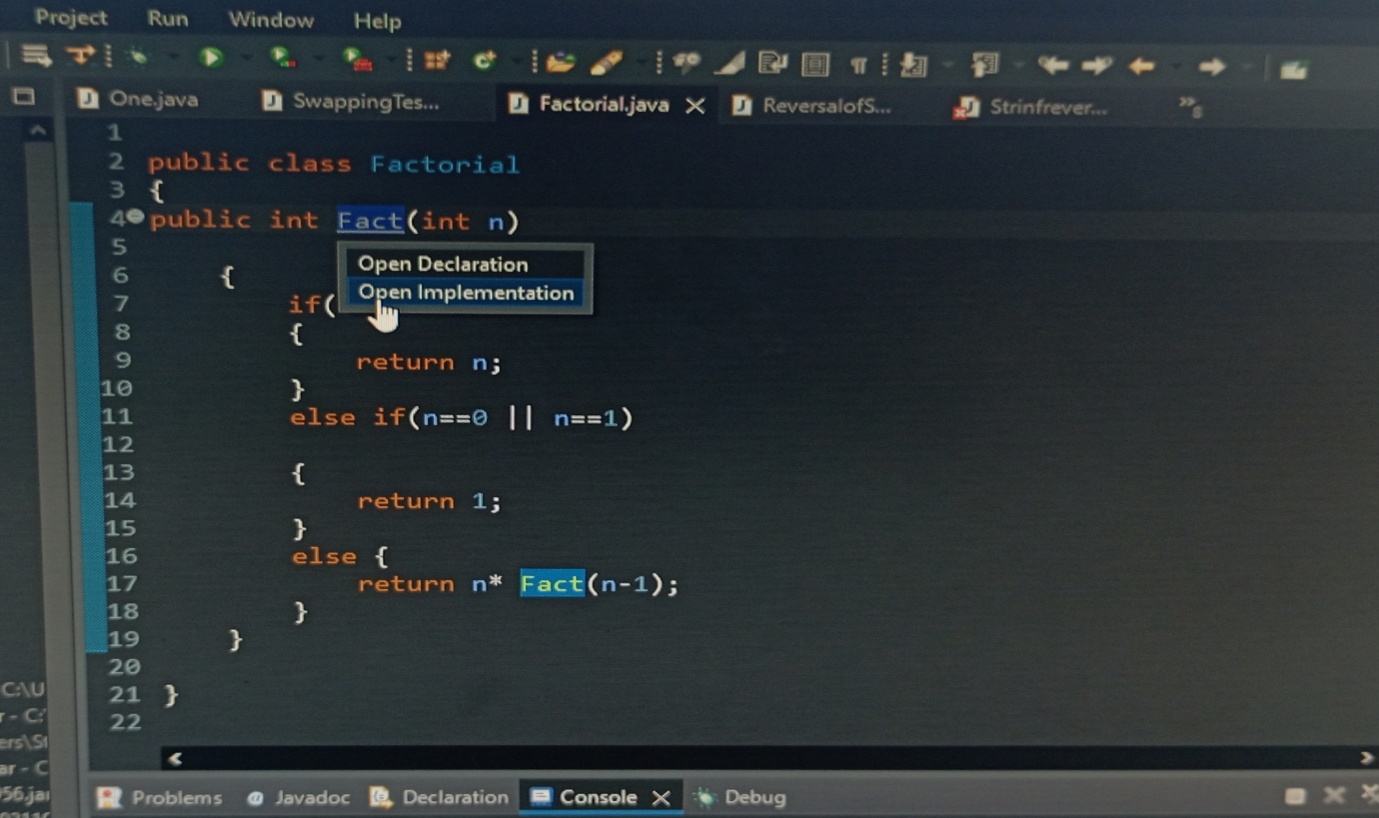


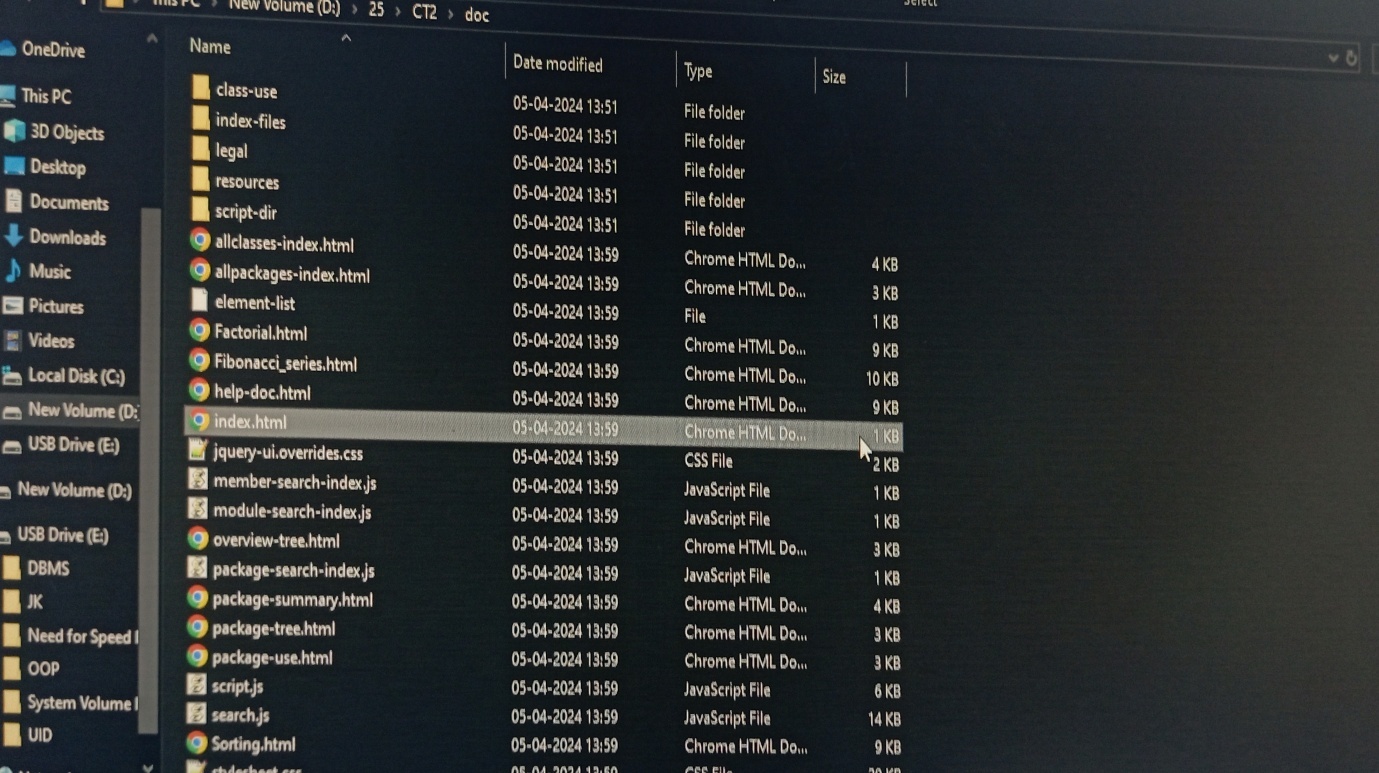
•Output

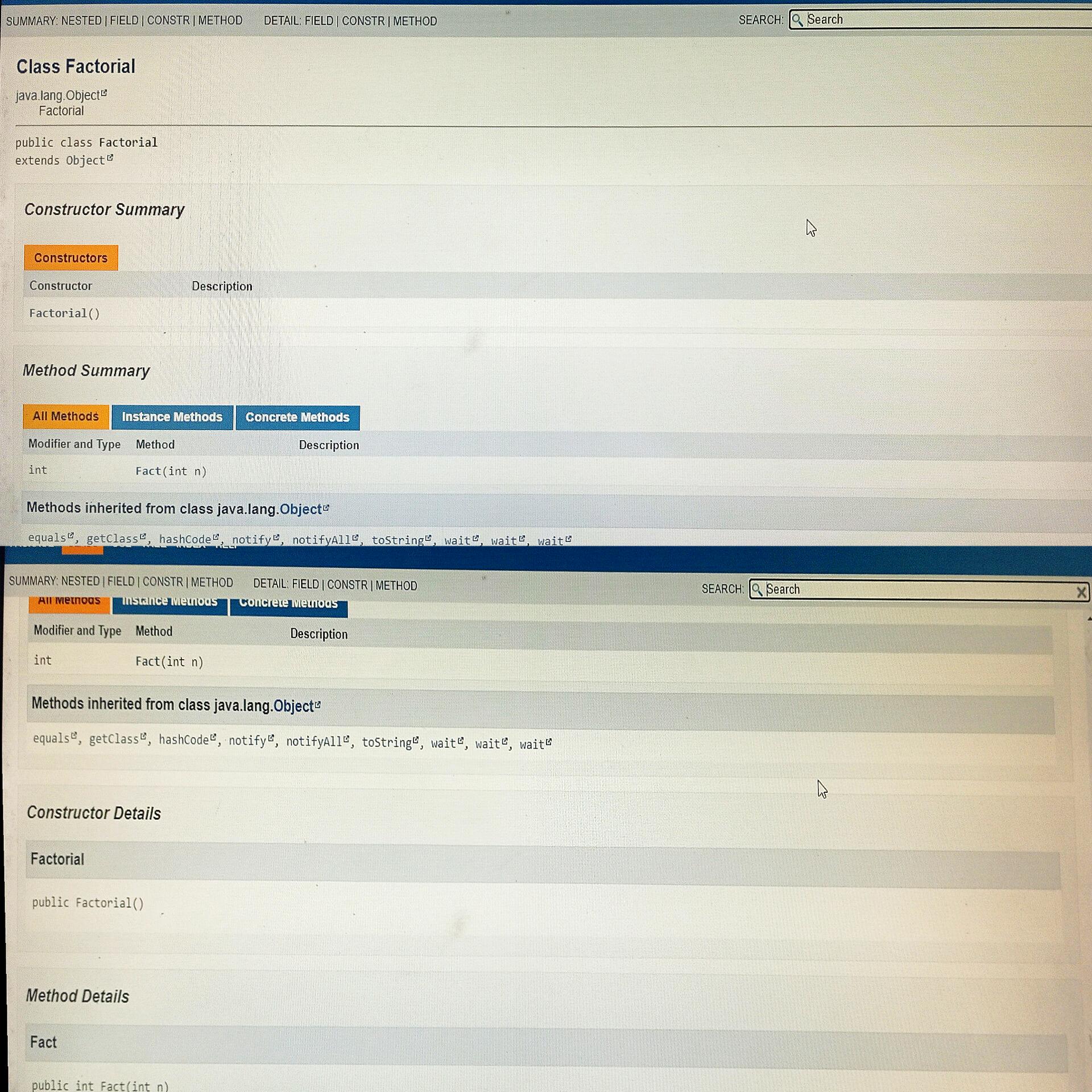
Javadoc

• Generate javadoc

• Select Project

• Open Implementation

• Open file

• Summary