TECHNISCHE UNIVERSITÄT WIEN

Welcome.TU.code

Dealing With Errors, Exceptions, Recursion

https://github.com/votacom/it4refugees

Agenda

- Short Recap
- Exception Handling
- Recursion

Recap Questions

“Who can explain what a variable is?”

Recap Questions

“Who can explain the difference between an declaration and initialisation of an variable?”

Recap Questions

“Who can explain what an array is?”

Recap Questions

“Who can explain how to use an array?”

Recap Questions

“Who can explain when if-statements are used?”

Recap Questions

“Who can name three different "loop" variations and explain how they work?”

Recap Questions

“Who can explain what a function is? why do we use functions?”

Recap Questions

“Who can explain how to declare a function?”

General Types of Errors in Java

- Syntax Errors
- Semantic Errors
- Runtime Errors

Syntax Errors

Occurs due to incorrect grammar

- Spelling mistakes
- Missing semicolons
- Improperly matches parentheses

Semantic Errors

This types of Errors indicate an improper use of the Java programming language.

- use of a non-initialized variable
- type incompatibility

Runtime Errors

- Occur during execution of an programm
- "***Exception Handling***" deals with this types of errors

Exceptions

- occur if something goes wrong
- often give a hint on the problem
- can be caught
- can be manually thrown
- can be created

Types of Exceptions

- RuntimeException/Unchecked Exceptions
  - ArrayIndexOutOfBoundsException
  - NullPointerException
  - etc.
- Checked Exceptions (require catch block)
  - IOException
  - etc.

Errors

- errors you can’t really do anything about at runtime
- typically ignored in code, tried to avoid as good as possible
- e.g.: StackOverFlowError

Handle Exceptions

- try-Block: what you try to do
- catch-clause: which exceptions you want to catch, what to do with the information of these
- finally Block: always executed, even if exception occurs, used to close open file etc.
- or instead of all of this: ‘throws’ clause in function header

Live Example

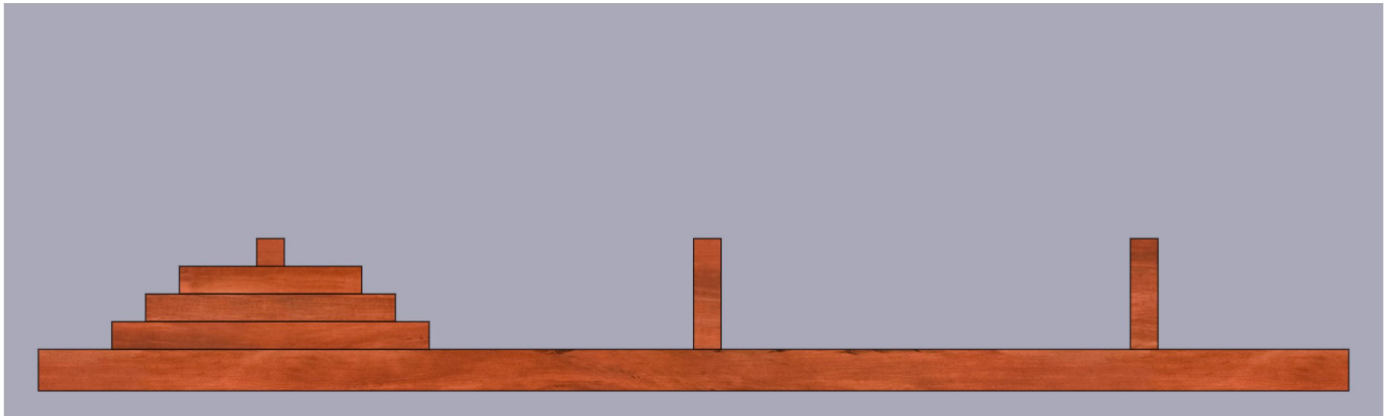
```
public static void main(String[] args) {
    try {
        //something causing an exception
    } catch(Exception e) {
        //what to do with the information of the exception
    } finally {
        //what you always want to do
    }
}
```

Recursion

- calling your function in the same function again
- simplifying code for specific problems

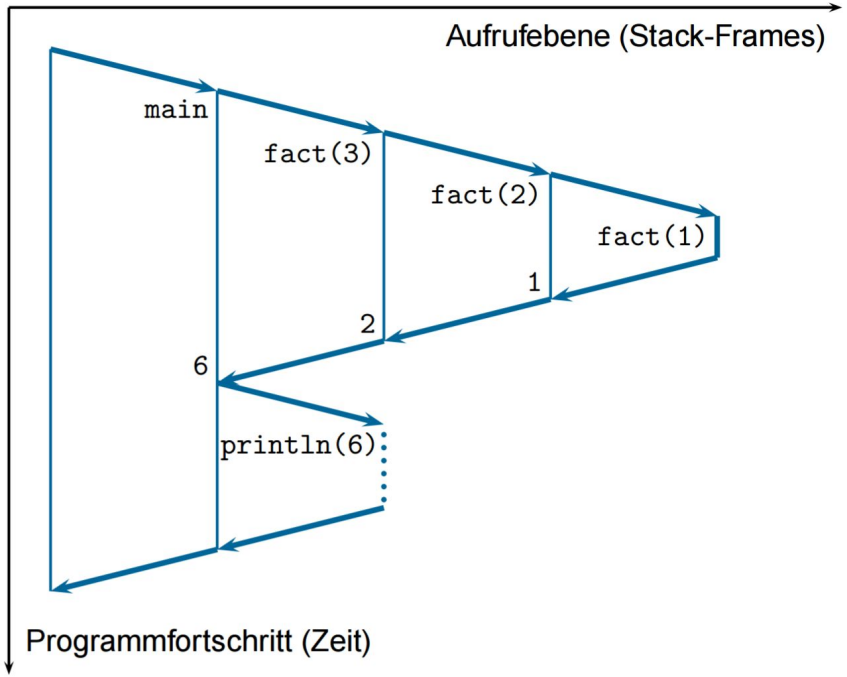
Live Examples

- Towers of Hanoi

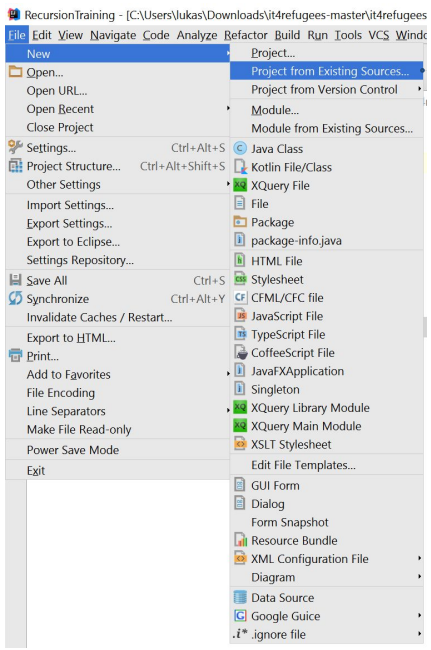


Problems when using Recursion

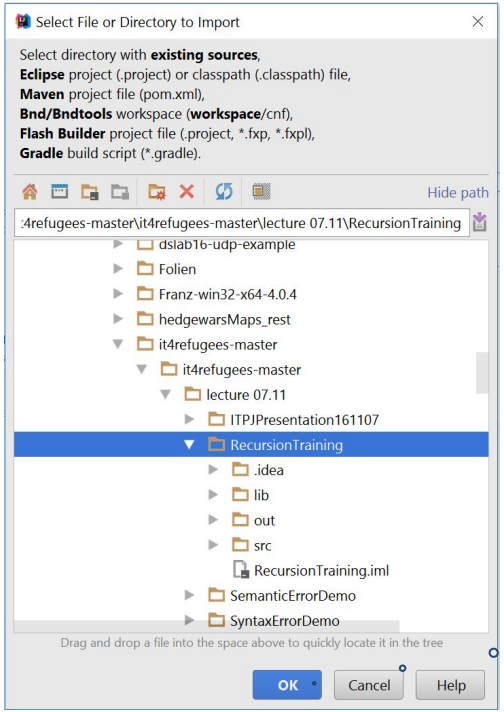
StackOverflowErrors



How To Homework 1

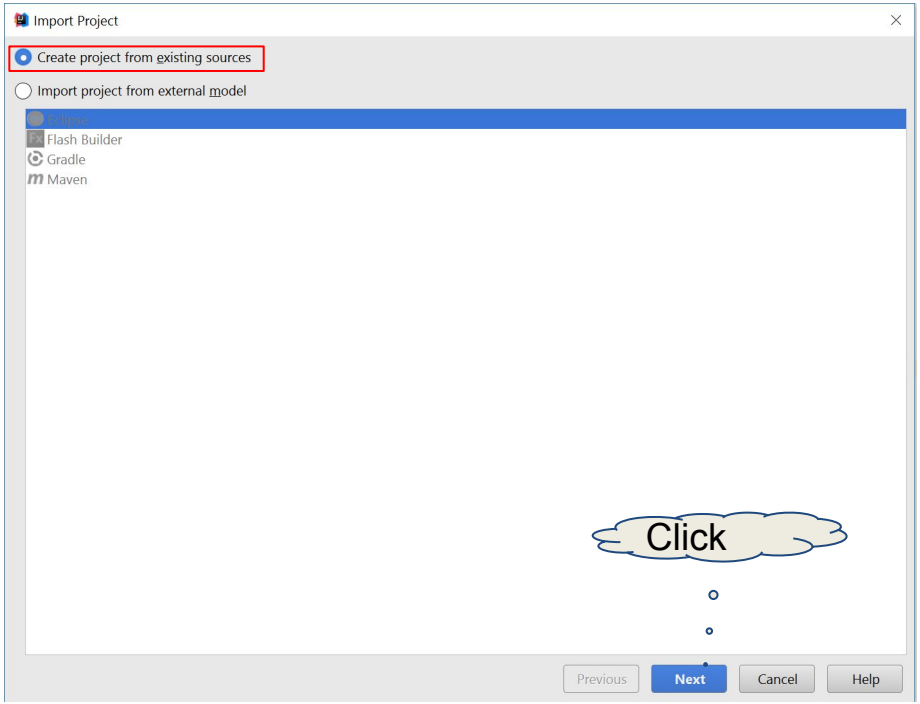


Click

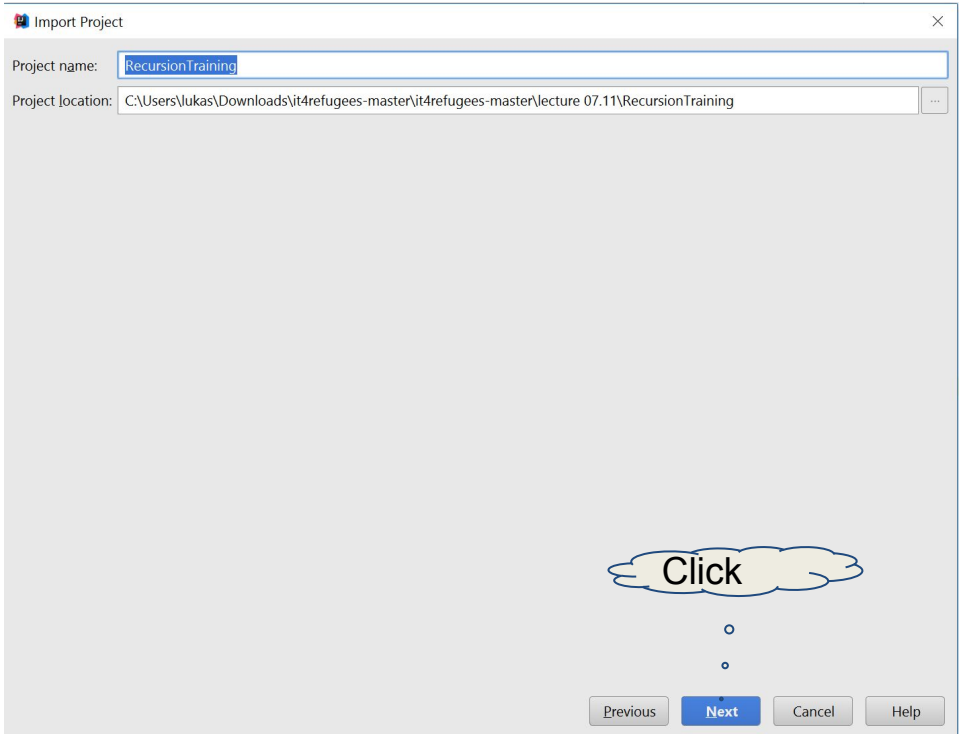


Click

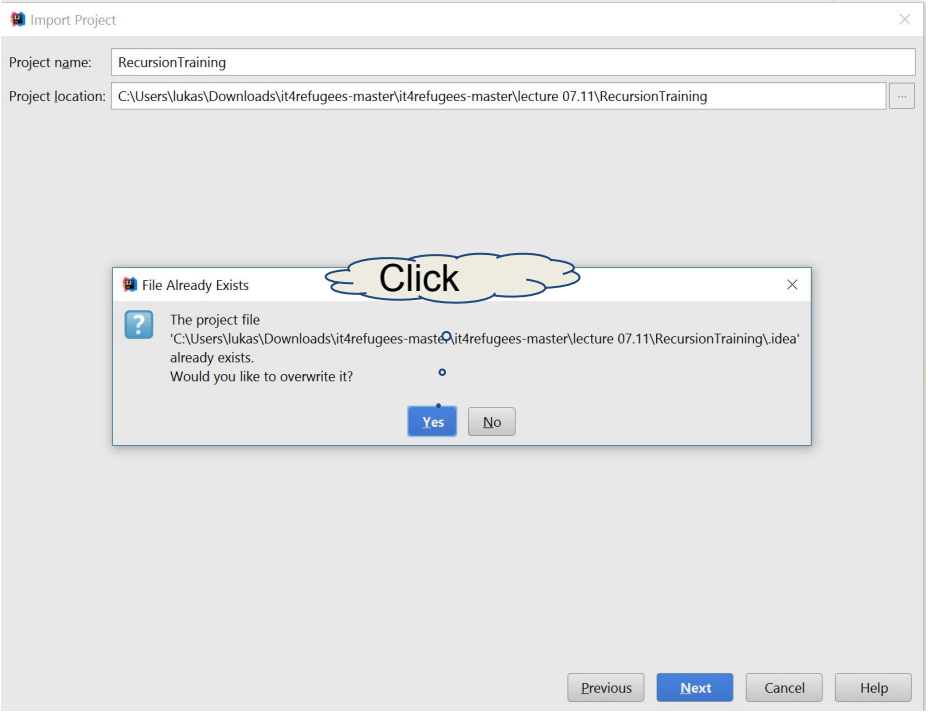
How To Homework 2



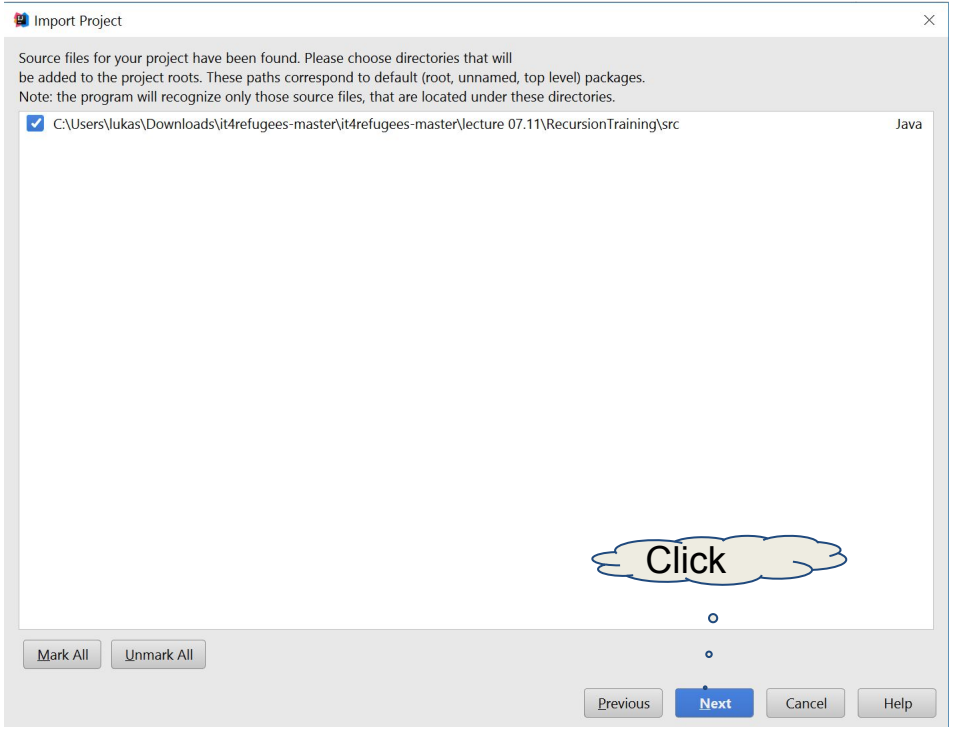
How To Homework 3



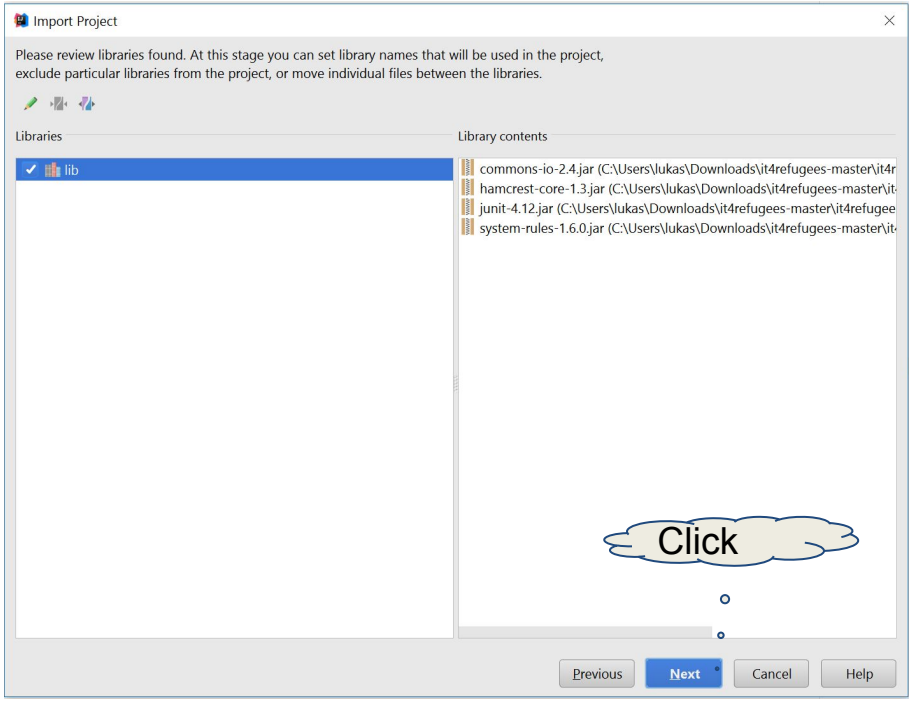
How To Homework 4



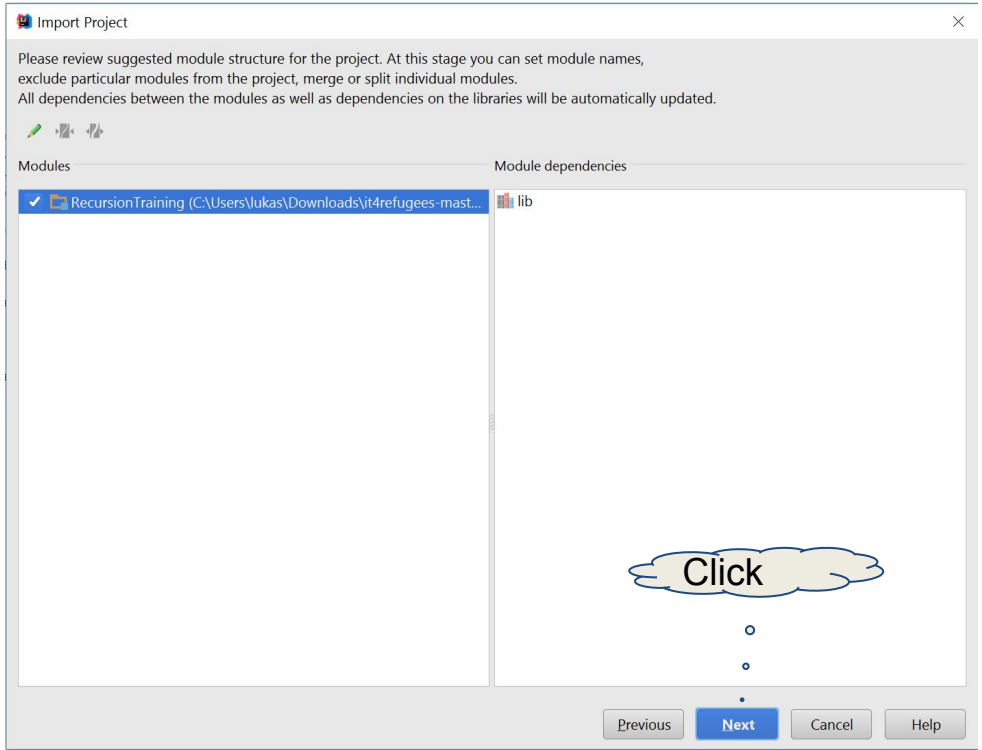
How To Homework 5



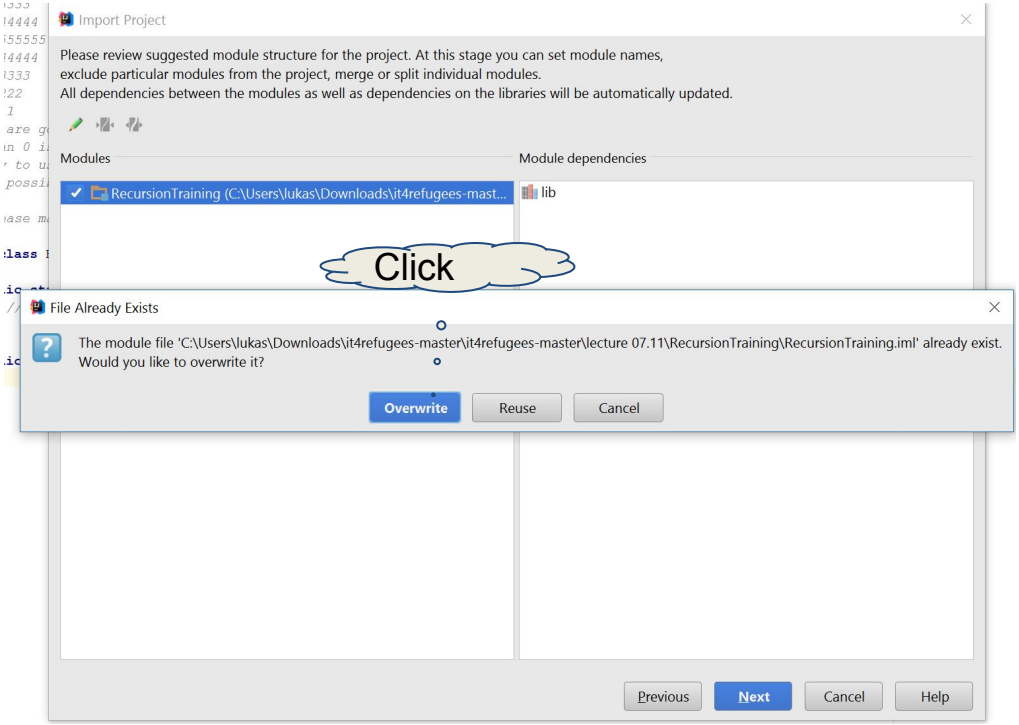
How To Homework 6



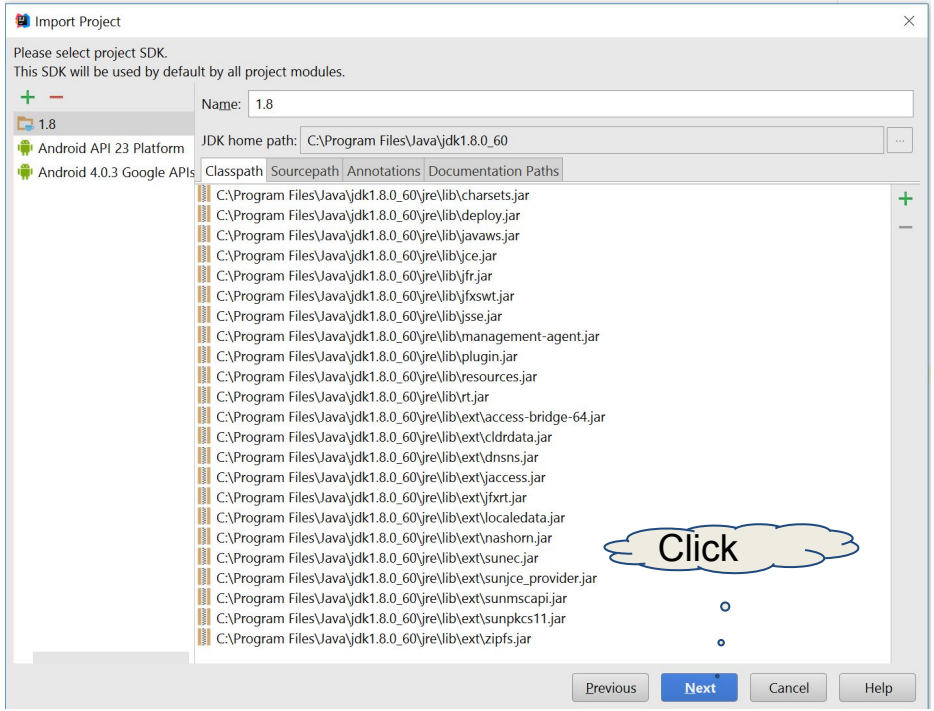
How To Homework 7



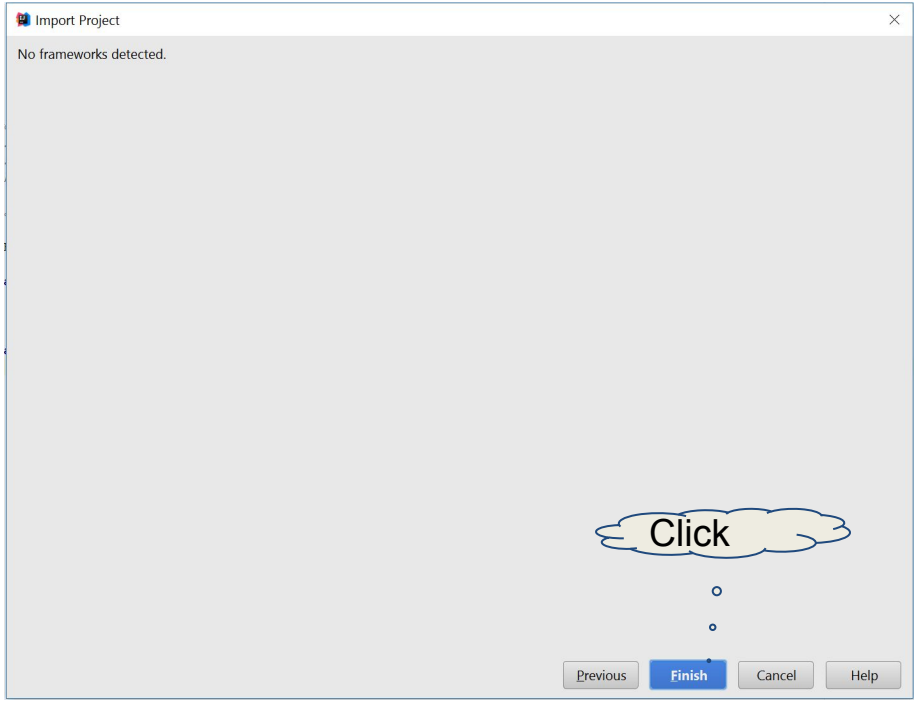
How To Homework 8



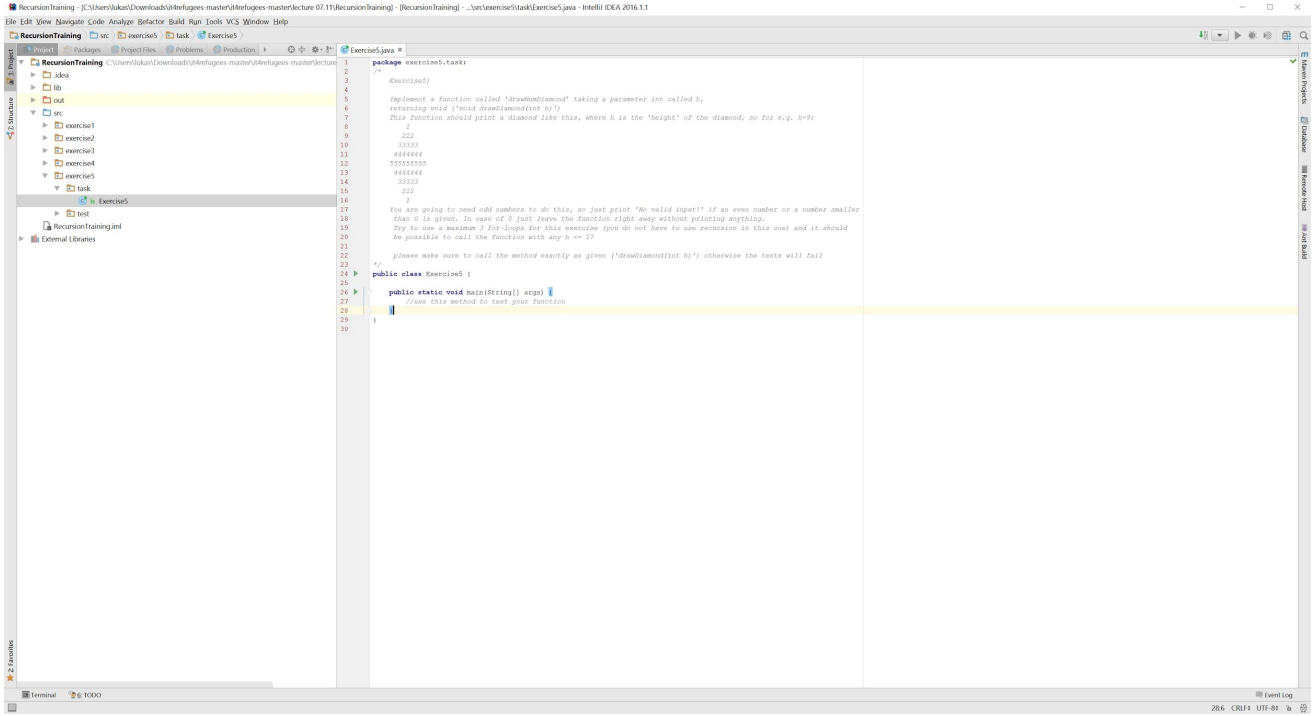
How To Homework 9



How To Homework 10



Homework - only Exercise5



don’t forget putting whitespace right of the numbers! otherwise the test will fail ;)

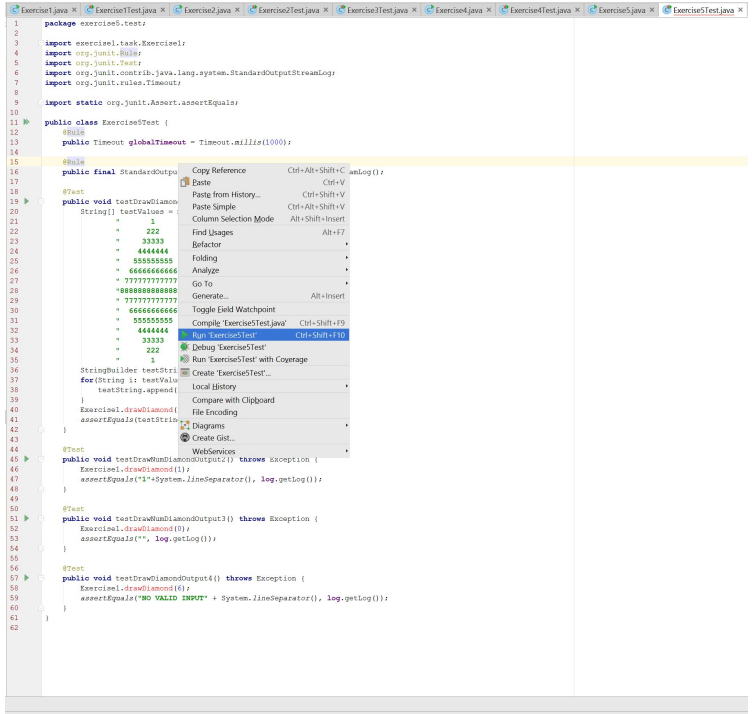
How To Run Tests

```
package exercise5.test;

import ...

public class Exercise5Test {

    @Rule
    public Timeout globalTimeout = Timeout.millis(1000);

    @Rule
    public final StandardOutputStreamLog log = new StandardOutputStreamLog();

    @Test
    public void testDrawNumDiamondOutput1() throws Exception {
        String[] testValues = new String[]{
                *       1       ",
                *      222      ",
                *     33333     ",
                *    4444444    ",
                *   555555555   ",
                *  66666666666  ",
                * 7777777777777 ",
                "888888888888888",
                * 7777777777777 ",
                *  66666666666  ",
                *   555555555   ",
                *    4444444    ",
                *     33333     ",
                *      222      ",
                *       1       ");
        StringBuilder testString = new StringBuilder();
        for (String i : testValues) {
            testString.append(i).append(System.lineSeparator());
        }
        Exercise5.drawDiamond(15);
        assertEquals(testString.toString(), log.getLog());
    }

    @Test
    public void testDrawNumDiamondOutput2() throws Exception {
        Exercise5.drawDiamond(1);
        assertEquals("1" + System.lineSeparator(), log.getLog());
    }

    @Test
    public void testDrawNumDiamondOutput3() throws Exception {
        Exercise5.drawDiamond(0);
        assertEquals("", log.getLog());
    }

    @Test
    public void testDrawNumDiamondOutput4() throws Exception {
        Exercise5.drawDiamond(-1);
        assertEquals("NO VALID INPUT" + System.lineSeparator(), log.getLog());
    }
}
```

Select Exercise5Test

→ will show error until you implement method ‘drawDiamon’ in the correct way!

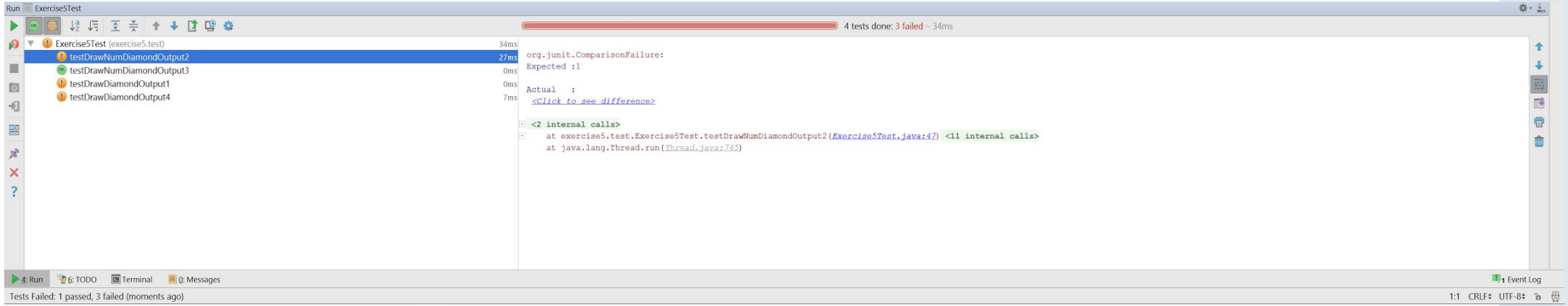
How To Run Tests



When implemented, select Exercise5Test and select “Run Exercise5Test()’

How To Run Tests

If the orange Circles turn green you finished the exercise and you have solved the exercise!



How To Run Tests

If you click <click to see difference> a new window opens, showing in grey what has been expected, and what was your output! You can use it to check why your test failed!