Algorithm of subroutine **int r2i(String roman)** of class RomanToInteger.java.

1. Loop from left, read character wise and determine its corresponding Integer equivalent.
2. If current number is lesser than next number apply subtractive rules. Following rules are accepted for this combination.

   i. Only one I, X, and C can be used as the leading numeral in part of a subtractive pair.
   ii. I can only be placed before V and X.
   iii. X can only be placed before L and C.
   iv. C can only be placed before D and M.

   Following subroutine does this job : private static void *validateSubtractiveRules*(String roman, int index)

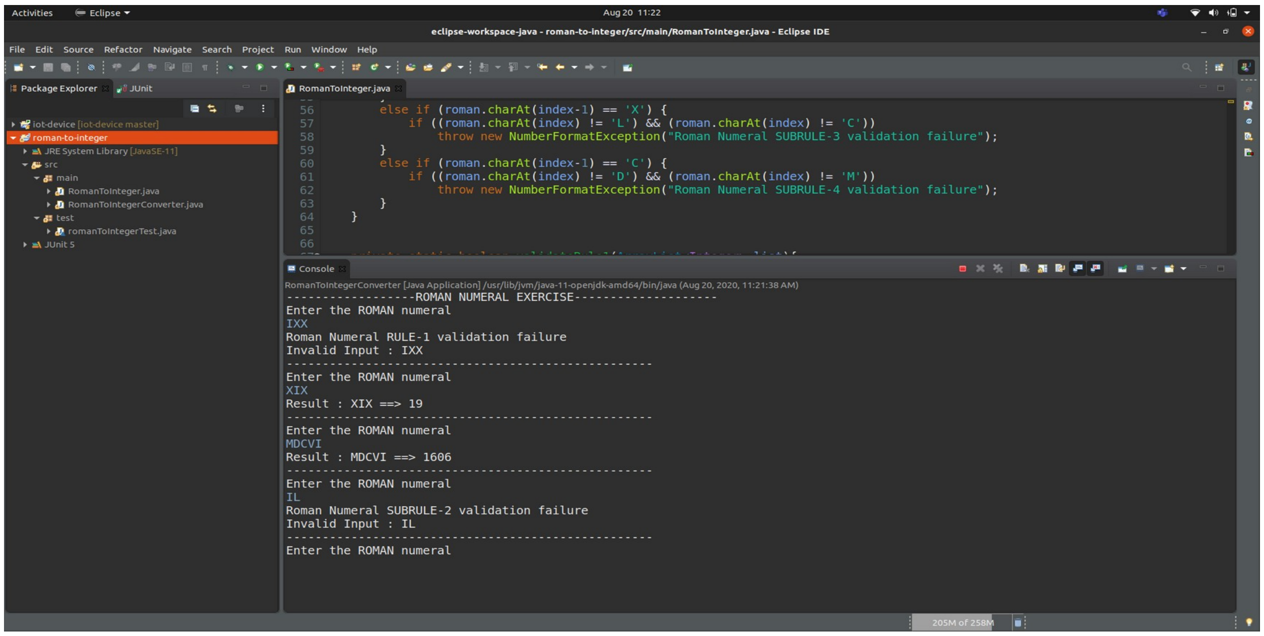
3. If Subtractive pair is valid add the difference between next and current value and add the difference to running sum.

4. If numbers in descending order keep adding values to running sum.

5. Once the running sum is obtained, now validate below rules. If rule validation fails raise exception.

1. Numerals must be arranged in descending order of size.
2. M, C, and X cannot be equaled or exceeded by smaller denominations.
- If No. of I >= 10 input is invalid.
- if No. of V >= 2 input is invalid. (Covered in RULE-3)
- if No. of X >= 10 input is invalid.
- If No. of L >= 2 input is invalid. (Covered in RULE-3)
- if No. of C >=10 input is invalid.
- if No. of D >=2 input is invalid. (Covered in RULE-3)
3. D, L, and V can each only appear once.

Code execution results:



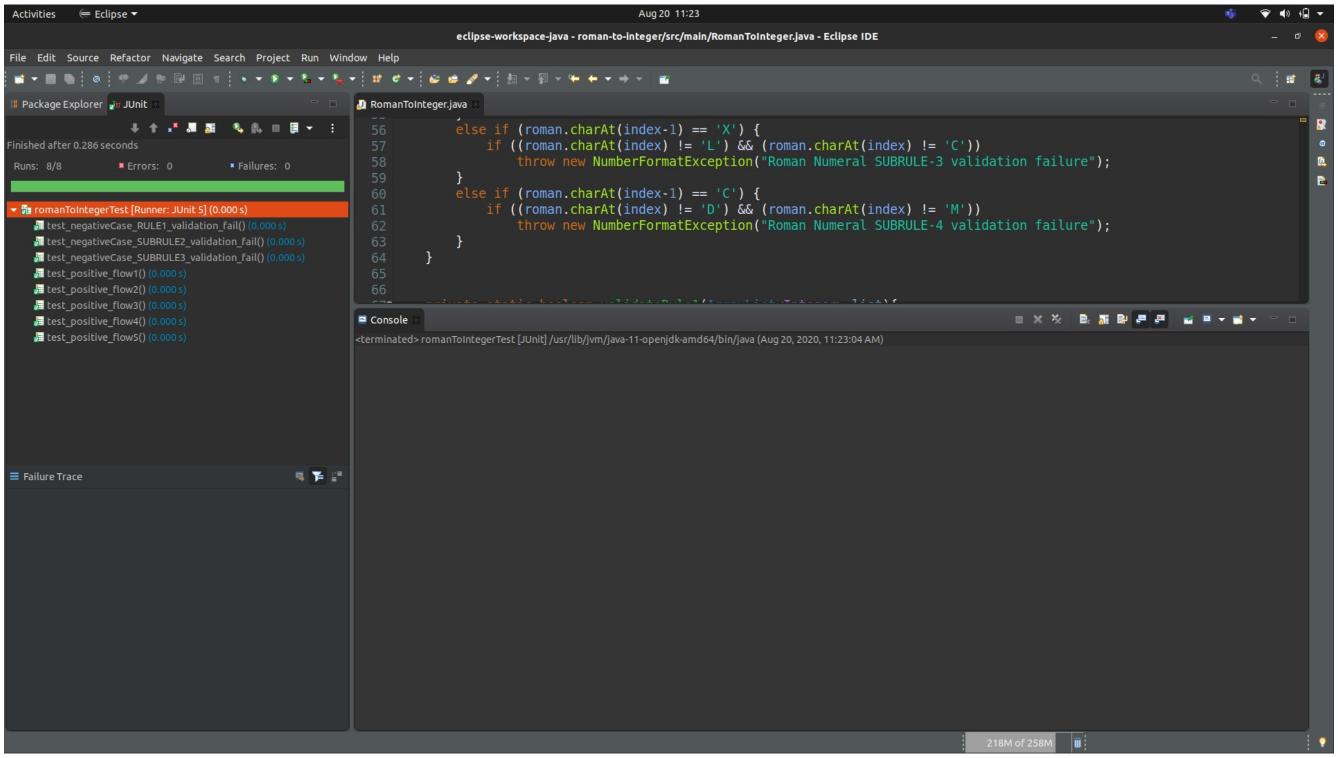
```java
        else if (roman.charAt(index-1) == 'X') {
            if ((roman.charAt(index) != 'L') && (roman.charAt(index) != 'C'))
                throw new NumberFormatException("Roman Numeral SUBRULE-3 validation failure");
        }
        else if (roman.charAt(index-1) == 'C') {
            if ((roman.charAt(index) != 'D') && (roman.charAt(index) != 'M'))
                throw new NumberFormatException("Roman Numeral SUBRULE-4 validation failure");
        }
    }
```

```
RomanToIntegerConverter [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Aug 20, 2020, 11:21:38 AM)
-----------------ROMAN NUMERAL EXERCISE--------------------
Enter the ROMAN numeral
IXX
Roman Numeral RULE-1 validation failure
Invalid Input : IXX
---------------------------------------------------
Enter the ROMAN numeral
XIX
Result : XIX ==> 19
---------------------------------------------------
Enter the ROMAN numeral
MDCVI
Result : MDCVI ==> 1606
---------------------------------------------------
Enter the ROMAN numeral
IL
Roman Numeral SUBRULE-2 validation failure
Invalid Input : IL
---------------------------------------------------
Enter the ROMAN numeral
```

Unit tests:

Please refer **romanToIntegerTest.java** for unit tests.

Tests are self explanatory.


Activities
Eclipse
Aug 20 11:23
eclipse-workspace-java - roman-to-integer/src/main/RomanToInteger.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
RomanToInteger.java
Console
Failure Trace
218M of 258M

```
Finished after 0.286 seconds
Runs: 8/8        Errors: 0        Failures: 0

romanToIntegerTest [Runner: JUnit 5] (0.000 s)
    test_negativeCase_RULE1_validation_fail() (0.000 s)
    test_negativeCase_SUBRULE2_validation_fail() (0.000 s)
    test_negativeCase_SUBRULE3_validation_fail() (0.000 s)
    test_positive_flow1() (0.000 s)
    test_positive_flow2() (0.000 s)
    test_positive_flow3() (0.000 s)
    test_positive_flow4() (0.000 s)
    test_positive_flow5() (0.000 s)
```

```java
        else if (roman.charAt(index-1) == 'X') {
            if ((roman.charAt(index) != 'L') && (roman.charAt(index) != 'C'))
                throw new NumberFormatException("Roman Numeral SUBRULE-3 validation failure");
        }
        else if (roman.charAt(index-1) == 'C') {
            if ((roman.charAt(index) != 'D') && (roman.charAt(index) != 'M'))
                throw new NumberFormatException("Roman Numeral SUBRULE-4 validation failure");
        }
    }

```

```
<terminated> romanToIntegerTest [JUnit] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Aug 20, 2020, 11:23:04 AM)
```