

Integrazione e Test di Sistemi Software 2024



Cristian Pazienza
ITPS 775695

Sommario

| | |
|--|-----------|
| Specification test | 3 |
| Step 1: Understanding the requirements | 3 |
| Step 2: Explore what the program does for various inputs | 4 |
| Step 3: Explore inputs, outputs and identify partitions | 5 |
| Step 4: Identify boundary cases | 6 |
| Step 5: Devise test cases | 7 |
| Step 6: Automate test cases..... | 8 |
| Step 7: Augment the test suite with creativity and experience | 10 |
| Structural testing..... | 11 |
| Code coverage | 11 |
| Mutation test..... | 11 |

Specification test

Step 1: Understanding the requirements

Il metodo `maskAndFillToLength()` modifica una stringa applicando una maschera su determinati caratteri a intervalli definiti e ripetendo la stringa fino a raggiungere una lunghezza massima.

Il metodo riceve quattro parametri in input:

1. **str**: la stringa di input per le modifiche
2. **maskChar**: un carattere per effettuare la mascherazione
3. **interval**: l'intervallo dei caratteri
4. **maxLength**: la lunghezza finale della stringa

Il metodo restituisce:

1. Una **stringa** modificata
2. **Null** se non vengono soddisfatti le precondizioni

Step 2: Explore what the program does for various inputs

Il metodo andrà ad eseguire queste azioni

1. Maschera i caratteri a intervalli definiti.
2. Ripete la stringa modificata fino a raggiungere maxLength.
3. Restituisce null in caso di:
 - str è null.
 - maxLength <= 0.
 - interval <= 0.

Eseguo dei semplici test per verificare se il metodo funzioni correttamente

```
import Esame.MaskAndRepeatUtils;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class SimpleExamTest {
    @Test
    public void simpleTest() {
        Assertions.assertEquals(MaskAndRepeatUtils.maskAndFillToLength("abc", '*', 2, 6),
            "a*c*a*c");
    }
    @Test
    public void simpleTest2() {
        Assertions.assertEquals(MaskAndRepeatUtils.maskAndFillToLength("abc", '*', 2, 4),
            "a*c*a");
    }
}
```

SimpleExamTest,simpleTest and 1 more: 2 total, 2 passed15 ms

Collapse | Expand

"C:\Program Files\Eclipse Adoptium\jdk-21.0.12-hotspot\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=55401:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "C:\Users\Cristian\.m2\repository\org\junit\platform\junit-platform-launcher\1.10.0\junit-platform-launcher-1.10.0.jar;C:\Users\Cristian\.m2\repository\org\junit\platform\junit-platform-engine\1.10.0\junit-platform-engine-1.10.0.jar;C:\Users\Cristian\.m2\repository\org\opentest4j\opentest4j\1.3.0\opentest4j-1.3.0.jar;C:\Users\Cristian\.m2\repository\org\junit\platform\junit-platform-commons\1.10.0\junit-platform-commons-1.10.0.jar;C:\Users\Cristian\.m2\repository\org\apiguardian\apiguardian-api\1.1.2\apiguardian-api-1.1.2.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\plugins\junit\lib\junit5-rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\plugins\junit\lib\junit-rt.jar;C:\Users\Cristian\IdeaProjects\Test\build\classes\java\test;C:\Users\Cristian\IdeaProjects\Test\build\classes\java\main;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\junit\jupiter\jupiter-5\10.0\8fea1d9c58b2156f1b998f2f18da04bc9e08774\junit-jupiter-5-10.0.jar;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\junit\jupiter\jupiter-params\5.10.0\9041c7365495a897a64782ea5a6fdb99dab1814e\junit-jupiter-params-5.10.0.jar;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\apiguardian\apiguardian-api\1.1.2\2a231e0d844d2721b0fa1b238006d15c6ded6842a\apiguardian-api-1.1.2.jar;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\opentest4j\opentest4j\1.3.0\0152ea56b3a72f655d4fd677fc0ef2596c3dd5e6e\opentest4j-1.3.0.jar;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\junit\jupiter\jupiter-engine\5.10.0\90587932d718fc51a48112d33045a18476c542ad\junit-jupiter-engine-5.10.0.jar;C:\Users\Cristian\gradle\caches\modules-2\files-2.1\org\junit\platform\junit-platform-engine\1.10.0\0276c4edcf64abb5a139fa7b4f99330d7a93b804\junit-platform-engine-1.10.0.jar" com.intellij.rt.junit.JUnitStarter -ideVersion5 -junit5 @w@C:\Users\Cristian\AppData\Local\Temp\idea_working_dirs_junit.tmp @C:\Users\Cristian\AppData\Local\Temp\idea_junit.tmp -socket55400 Process finished with exit code 0

| | | |
|----------------|--------|-------|
| SimpleExamTest | | 15 ms |
| simpleTest() | passed | 15 ms |
| simpleTest2() | passed | |

Generated by IntelliJ IDEA on 12/11/24, 18:21

Con questo semplice test osservo che il metodo funzioni come mi aspetto

4

Step 3: Explore inputs, outputs and identify partitions

Le classi di test per l'input sono:

- Str:
 1. Una stringa vuota
 2. Null
 3. Lunghezza della stringa 1
 4. Lunghezza della stringa maggiore 1
- maskChar
 1. Un singolo carattere
 2. Nessun carattere
- Interval
 1. Positivo
 2. Negativo
- maxLength
 1. Positivo
 2. Negativo

Le combinazioni di input sono:

maxLength < lunghezza str

interval > lunghezza str

interval > lunghezza str e maxLength < lunghezza str

interval > lunghezza str e maxLength > lunghezza str

I test totali sono $4 \times 2 \times 2 \times 2 = 128$ test

Step 4: Identify boundary cases

Andiamo a identificare i boundary cases:

- `if (str == null || maxLength <= 0 || interval <= 0)`
per **str**:
on point: null, rende la condizione **vera** quindi sarà anche in point
off point: lunghezza di str >= 1, rende la condizione **falsa** quindi sarà anche out point
per **maxLength**:
on point: 0, rende la condizione **vera** quindi sarà anche in point
off point: -1, rende la condizione **falsa** quindi sarà anche out point
per **interval**:
on point: 0, rende la condizione **vera** quindi sarà anche in point
off point: -1, rende la condizione **falsa** quindi sarà anche out point
- `if(str.isEmpty())`
on point: empty, rende la condizione **vera**, quindi, sarà anche in point
off point: lunghezza di str >= 1, rende la condizione **falsa** quindi sarà anche out point
- `if ((i + 1) % interval == 0)`
on point: 0, rende la condizione **vera**, quindi, sarà anche in point
off point: $(i + 1) \% interval == 0 + 1$, rende la condizione **falsa** quindi sarà anche out point
- `if (maskedStr.length() >= maxLength)`
on point: maxLength, rende la condizione **vera**, quindi, sarà anche in point
T24: off point: maxLength-1, rende la condizione **falsa** quindi sarà anche out point
- `if (result.length() < maxLength)`
on point: maxLength, rende la condizione **falsa**
off point: maxLength+1, rende la condizione **vera**
in point: maxLength+1
out point: maxLenght

Step 5: Devise test cases

Testo un sottoinsieme di possibili test andando ad ottimizzare la suite di test

T1: str empty

T2: str null

T3: 1 char

T4: empty char

T5: interval negative

T6: maxLength negative

T7: maxLength < lunghezza str

T8: interval > lunghezza str e maxLength < lunghezza str

T9: interval > lunghezza str e maxLength > lunghezza str

Andando a testare 9 casi e non 128.

Step 6: Automate test cases

Utilizzo i test parametrici per evitare duplicazione di codice all'interno della suite di test

```
public class MaskAndRepeatUtilsTest {

    @ParameterizedTest
    @MethodSource("generator")
    @DisplayName("I test the maskAndFillToLength method of the MaskAndRepeatUtils class")
    public void testMaskAndRepeat(final String str, final char maskChar, final int interval,
final int maxLength, String expected) {
        Assertions.assertEquals(MaskAndRepeatUtils.maskAndFillToLength(str, maskChar,
interval, maxLength), expected);
    }

    static Stream<Arguments> generator() {
        return Stream.of(
            of("", "*", 1, 3, ""),
            of(null, "*", 1, 3, null),

            of("pippo", "*", 2, 5, "p*p*o"),
            of("pippo", " ", 2, 5, "p p o"),

            of("", "*", -1, 3, null),

            of("", "*", 1, -3, null),

            of("Ciao", "X", 2, 3, "CXa"),
            of("Ciao", "X", 5, 2, "Ci"),
            of("Ciao", "X", 5, 6, "CiaoCi")
        );
    }
}
```


MaskAndRepeatUtilsTest (1): 9 total, 9 passed101 ms

[Collapse](#) | [Expand](#)

| | |
|---|--------------|
| MaskAndRepeatUtilsTest | 73 ms |
| I test the maskAndFillToLength method of the MaskAndRepeatUtils class | 68 ms |
| [1] , *, 1, 3, | passed 24 ms |
| [2] null, *, 1, 3, null | passed 2 ms |
| [3] pippo, *, 2, 5, p*p*o | passed 1 ms |
| [4] pippo, , 2, 5, p p o | passed 1 ms |
| [5] , *, -1, 3, null | passed 0 ms |
| [6] , *, 1, -3, null | passed 1 ms |
| [7] Ciao, X, 2, 3, CXa | passed 1 ms |
| [8] Ciao, X, 5, 2, Ci | passed 1 ms |
| [9] Ciao, X, 5, 6, CiaoCi | passed 2 ms |

Generated by IntelliJ IDEA on 08/12/24, 11:44

I test passano tutti con successo

Step 7: Augment the test suite with creativity and experience

Aggiungo un test dove char sarà un carattere speciale

T10: char = "§"

Un test in cui interval è il numero di macchina più grande

T11: interval = MAX_VALUE

Un test in cui interval è il numero di macchina più piccolo

T12: interval = MIN_VALUE

Un test in cui maxLength è il numero di macchina più piccolo

T13: maxLength = MIN_VALUE

Creo anche dei test per le combinazioni di interval e maxLength in cui sono

T14: maxLength = MAX_VALUE e interval = MIN_VALUE

T15: maxLength = MIN_VALUE e interval = MAX_VALUE

| | | |
|---|--------|-------|
| MaskAndRepeatUtilsTest (1): 15 total, 15 passed | | 96 ms |
| Collapse Expand | | |
| MaskAndRepeatUtilsTest | | 69 ms |
| I test the maskAndFillToLength method of the MaskAndRepeatUtils class | | 65 ms |
| [1] ., ., 1, 3, | passed | 22 ms |
| [2] null, ., 1, 3, null | passed | 1 ms |
| [3] pippo, ., 2, 5, p"p"o | passed | 0 ms |
| [4] pippo, ., 2, 5, p p o | passed | 1 ms |
| [5] ., ., -1, 3, null | passed | 0 ms |
| [6] ., ., 1, -3, null | passed | 1 ms |
| [7] Ciao, X, 2, 3, CXa | passed | 1 ms |
| [8] Ciao, X, 5, 2, Cl | passed | 1 ms |
| [9] Ciao, X, 5, 6, CiaoCl | passed | 1 ms |
| [10] Ciao, §, 2, 8, C§a§C§a§ | passed | 1 ms |
| [11] Ciao, ., 2147483647, 4, Ciao | passed | 1 ms |
| [12] Ciao, ., -2147483648, 4, null | passed | 1 ms |
| [13] Ciao, ., 3, -2147483648, null | passed | 0 ms |
| [14] Ciao, ., -2147483648, 2147483647, null | passed | 0 ms |
| [15] Ciao, ., 2147483647, -2147483648, null | passed | 0 ms |

I test passano tutti con successo

Structural testing

Code coverage

Utilizzo la code coverage di IntelliJ per controllare se i test coprono tutte le righe di codice e i branch, in questo caso copre il 100% di entrambe

| Element ^ | Class, % | Method, % | Line, % | Branch, % |
|--------------------|------------|------------|--------------|--------------|
| all | 25% (1/4) | 20% (1/5) | 32% (17/52) | 39% (18/46) |
| Capitolo3 | 0% (0/2) | 0% (0/3) | 0% (0/32) | 0% (0/26) |
| Esame | 100% (1/1) | 100% (1/1) | 100% (17/17) | 100% (18/18) |
| MaskAndRepeatUtils | 100% (1/1) | 100% (1/1) | 100% (17/17) | 100% (18/18) |
| org.example | 0% (0/1) | 0% (0/1) | 0% (0/3) | 0% (0/2) |

Mutation test

Utilizzato il framework **PITest** per effettuare i mutation test. Osserviamo che la suite di test sopravvive a quasi tutti i mutanti, questo indica che la suite è robusta.

Pit Test Coverage Report

Package Summary

Esame

| Number of Classes | Line Coverage | Mutation Coverage | Test Strength |
|-------------------|---|---|---|
| 1 | 94% <div><div></div><div></div></div> 17/18 | 74% <div><div></div><div></div></div> 17/23 | 74% <div><div></div><div></div></div> 17/23 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage | Test Strength |
|---|---|---|---|
| MaskAndRepeatUtils.java | 94% <div><div></div><div></div></div> 17/18 | 74% <div><div></div><div></div></div> 17/23 | 74% <div><div></div><div></div></div> 17/23 |

Report generated by [PIT](#) 1.15.8

Mutations

| | |
|----|---|
| 16 | 1. changed conditional boundary → KILLED 2. changed conditional boundary → KILLED 3. negated conditional → KILLED 4. negated conditional → KILLED 5. negated conditional → KILLED |
| 17 | 1. replaced return value with "" for Esame/MaskAndRepeatUtils::maskAndFillToLength → KILLED |
| 20 | 1. negated conditional → TIMED_OUT |
| 21 | 1. replaced return value with "" for Esame/MaskAndRepeatUtils::maskAndFillToLength → SURVIVED |
| 25 | 1. changed conditional boundary → KILLED 2. negated conditional → TIMED_OUT |
| 26 | 1. Replaced integer addition with subtraction → KILLED 2. Replaced integer modulus with multiplication → KILLED 3. negated conditional → KILLED |
| 33 | 1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED |
| 34 | 1. replaced return value with "" for Esame/MaskAndRepeatUtils::maskAndFillToLength → KILLED |
| 38 | 1. changed conditional boundary → SURVIVED 2. Replaced integer addition with subtraction → KILLED 3. negated conditional → KILLED |
| 42 | 1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED |
| 43 | 1. Replaced integer subtraction with addition → KILLED |
| 46 | 1. replaced return value with "" for Esame/MaskAndRepeatUtils::maskAndFillToLength → KILLED |