



Andy

An educational tool to assess test code

<https://github.com/cse1110/andy>



Teaching software testing is fundamental!

- It is an important skill to learn
- Not a lot of universities teach it.
 - In most of them, testing is just a lecture in a SE course
 - Delft has a dedicated testing course!
- Lack of tools to support testing education!
 - How to make sure the student wrote the "right tests"?



Andy!

The image shows a screenshot of a GitHub repository page for 'cse1110/andy'. The browser address bar shows 'github.com/cse1110/andy'. The repository is public and has 62 stars, 7 forks, and 6 unwatchers. The repository name 'cse1110/andy' is displayed with a 'Public' badge. Navigation links include Code, Issues (25), Pull requests, Discussions, Actions, Security, Insights, and Settings. The repository has 2 branches and 39 tags. A commit by 'mauricioaniche' and 'mauricio' is highlighted, titled 'disable CI workflows that need re-work (#187)', committed 12 minutes ago. The file list includes .github, andy-aws-lambda, andy-maven-plugin, andy, assignments, .gitignore, LICENSE, README.md, and pom.xml. The README.md file is open, showing the title 'Andy'. The right sidebar contains an 'About' section with the text 'Andy assesses student's test code. It's used in CSE1110, TU Delft.', a 'Releases' section with 39 tags, and a 'Packages' section with no published packages.

github.com/cse1110/andy

Managed Bookmarks Tech Academy Workday Adyen TU Delft E-Learning Personal Tango Testing metrics

Search or jump to...

Code Issues 25 Pull requests Discussions Actions Security Insights Settings

Go to file Add file Code

main 2 branches 39 tags

mauricioaniche and mauricio disable CI workflows that need re-work (#187) 589be53 12 minutes ago 1,122 commits

| File | Commit | Time |
|-------------------|---|----------------|
| .github | disable CI workflows that need re-work (#187) | 12 minutes ago |
| andy-aws-lambda | Draft: Assignments (#182) | 8 hours ago |
| andy-maven-plugin | Draft: Assignments (#182) | 8 hours ago |
| andy | Draft: Assignments (#182) | 8 hours ago |
| assignments | Draft: Assignments (#182) | 8 hours ago |
| .gitignore | Changes the README to uphold the sanctity of human life | last year |
| LICENSE | README and LICENSE | 2 years ago |
| README.md | Move screenshot folder to .github | 5 months ago |
| pom.xml | Draft: Assignments (#182) | 8 hours ago |

README.md

Andy

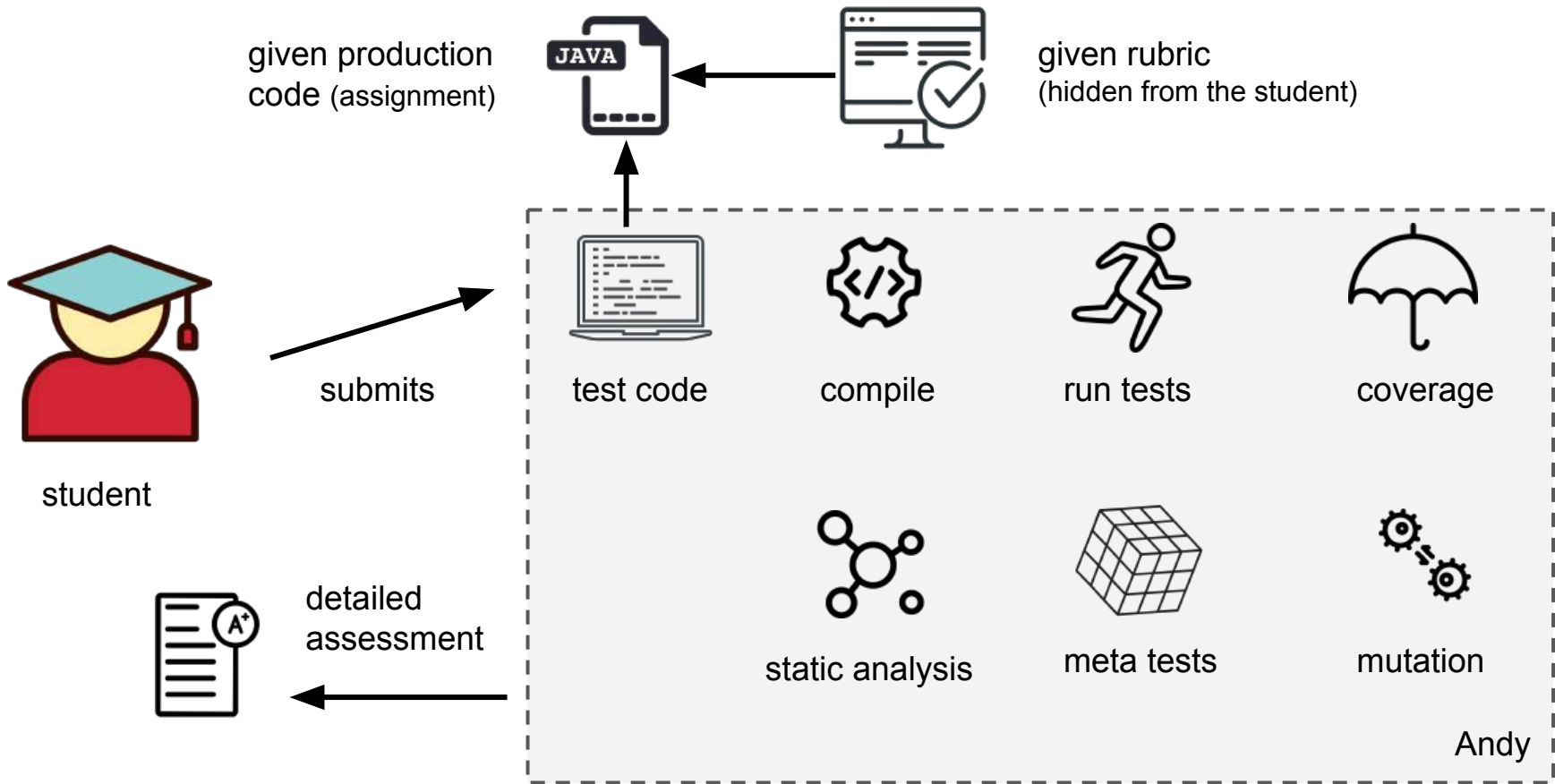
Andy assesses student's test code. It's used in CSE1110, TU Delft.

Readme MIT license Activity 62 stars 6 watching 7 forks Report repository

Releases 39 tags Create a new release

Packages No packages published Publish your first package

Used by 18



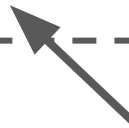
Some quick explanation of the assignment



The `concatenate` method concatenates a sequence of arrays. The return array consists of the entries of the input arrays concatenated in the order they appear in the argument list. Null arrays cause `NullPointerExceptions`; zero length arrays are allowed (contributing nothing to the output array).

Write tests for this program.

(Method from Apache Commons Math: <https://github.com/apache/commons-math/blob/acfc27083412f57a71a497e500070a08a956300d/commons-math-legacy-core/src/main/java/org/apache/commons/math4/legacy/core/MathArrays.java#L1075>)



Reference to where we got inspiration from

```
class MathArrays {
```

```
/** <code>MathArrays</code> should not normally be instantiated. */
```

```
private MathArrays() {  
}
```

```
/**
```

```
 * Concatenates a sequence of arrays. The return array consists of the entries  
 * of the input arrays concatenated in the order they appear in the argument  
 * list. Null arrays cause NullPointerExceptions; zero length arrays are allowed  
 * (contributing nothing to the output array).  
 *
```

```
 * @param x
```

```
 *     list of double[] arrays to concatenate
```

```
 * @return a new array consisting of the entries of the argument arrays
```

```
 * @throws NullPointerException
```

```
 *     if any of the arrays are null
```

```
 * @since 3.6
```

```
 */
```

```
public static double[] concatenate(double[]... x) {
```

```
    int combinedLength = 0;
```

```
    for (double[] a : x) {
```

```
        combinedLength += a.length;
```

```
    }
```

```
    int offset = 0;
```

```
    int curLength = 0;
```

```
    final double[] combined = new double[combinedLength];
```

```
    for (int i = 0; i < x.length; i++) {
```

```
        curLength = x[i].length;
```

```
        System.arraycopy(x[i], 0, combined, offset, curLength);
```

```
        offset += curLength;
```


```
    }
```

```
    return combined;
```

```
}
```

```
}
```

Javadoc that may
complement the
requirement




Plain old Java code




```
10 public class Configuration extends RunConfiguration {
11
12     @Override
13     public Map<String, Float> weights() {
14         return new HashMap<>() {{
15             put("coverage", 0.15f); // was 1.0
16             put("mutation", 0.45f); // was 0.0
17             put("meta", 0.4f); // was 0.0
18             put("codechecks", 0.0f);
19         }};
20     }
21
22     @Override
23     public List<String> classesUnderTest() {
24         return List.of("delft.MathArrays");
25     }
26
27     @Override
28     public List<MetaTest> metaTests() {
29         return List.of(
30             MetaTest.insertAt("does not work with no arrays provided", 23,
31                 "if (x.length == 0) throw new RuntimeException(\"killed the mutant\");"),
32             MetaTest.insertAt("does not work with a single array", 23,
33                 "if (x.length == 1) throw new RuntimeException(\"killed the mutant\");"),
34             MetaTest.withStringReplacement("does not work with more than 2 arrays",
35                 "i < x.length;",
36                 "i < x.length && i < 2;"),
37             MetaTest.insertAt("does not work with an empty array provided", 32,
38                 "if (curLength == 0) throw new RuntimeException(\"killed the mutant\");"),
39             MetaTest.withStringReplacement("does not work with arrays of different sizes",
40                 "curLength = x[i].length;",
41                 "curLength = x[0].length;")
42         );
43     }
44 }
45 }
```

How much each part contributes to the final assessment



The name of the class under test



Meta tests



```
23 class ReleaseEditions {
24
25     private final BookService bookService;
26
27     private final EmailService emailService;
28
29     public ReleaseEditions(BookService bookService, EmailService emailService) {
30         this.bookService = bookService;
31         this.emailService = emailService;
32     }
33
34     /**
35     * Creates a copy of the book, now with the new edition in the title.
36     *
37     * @param author
38     *     name of the author
39     * @param keyword
40     *     keyword to search
41     * @param edition
42     *     number of the new edition
43     */
44     public void releaseNewEdition(String author, String keyword, int edition) {
45         List<String> allBooks = bookService.retrieveBooks(author);
46         for (String bookTitle : allBooks) {
47             if (bookTitle.contains(keyword)) {
48                 bookService.addBook(author, bookTitle + " - edition " + edition);
49             }
50         }
51     }
52 }
```


Requires mocking!




```
@Override
public CheckScript checkScript() {
    return new CheckScript(List.of(
        new SingleCheck("EmailService should be mocked",
            new MockClass("EmailService")),
        new SingleCheck("BookService should be mocked",
            new MockClass("BookService")),
        new SingleCheck("BusinessImpl should not be mocked", true,
            new MockClass("BusinessImpl")),
        new SingleCheck("Spies should not be used", true,
            new MockitoSpy()),
        new SingleCheck(2, "retrieveBooks should not be verified", true,
            new MockitoVerify("retrieveBooks", MockitoVerify.MethodType.TEST,
                Comparison.GTE, 1)),
        new SingleCheck("addBook should be verified",
            new MockitoVerify("addBook", MockitoVerify.MethodType.TEST,
                Comparison.GTE, 1, true))
    ));
}

@Override
public List<MetaTest> metaTests() {
    return List.of(
        MetaTest.withLineReplacement("change condition", 45, 50,
            """
                List<String> allBooks = bookService.retrieveBooks(author);
                for (String bookTitle : allBooks) {
                    if (!bookTitle.contains(keyword)) {
                        bookService.addBook(author, bookTitle + " - edition " + edition);
                    }
                }
            """)
    );
}
}
```

Static analysis to
identify whether
mocks were used
properly!



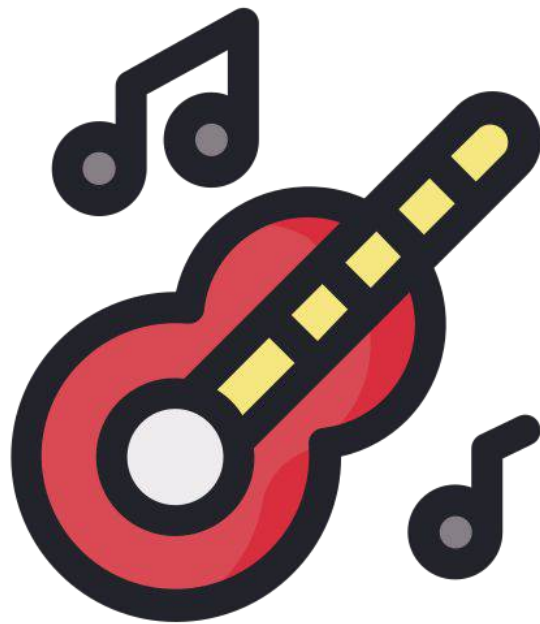
Code checks

The tool contains different checks for JUnit, Mockito, and JQWik tests:

- Test methods:
 - `NumberOfTests` : checks whether the test suite has a minimum number of tests.
 - `TestMethodsHaveAssertions` : checks whether all test methods have assertions.
 - `LoopInTestMethods` : checks whether there is a loop in a test method.
 - `UseOfStringLiterals` : checks whether there is a string literal in a test method.
 - `MethodCalledInTestMethod` : checks whether a method was invoked in a test method.
- Mockito:
 - `MockClass` : Checks whether a class was mocked in the test suite.
 - `MockitoSpy` : Checks whether spies are used.
 - `MockitoVerify` : Checks whether a specific verify has happened.
 - `MockitoWhen` : Checks whether a specific when() has happened.
- JQWik:
 - `JQWikProperty` : checks whether the test suite has a minimum number of properties.
 - `JQWikProvide` : checks whether the test suite has a minimum number of provide.
 - `JQWikCombinator` : checks whether a Combinator was used.
 - `JQWikArbitrary` : checks whether a specific Arbitrary is provided by any method in the test suite.
 - `JQWikArbitraries` : checks whether a Arbitraries.x() is used in the test suite.
 - `JQWikProvideAnnotations` : checks whether tests use Provide annotations, e.g., `@ForAll`, `@Positive`.

Each of these checks receive different parameters. Check their specific Javadoc for more details.

It's demo time!



Assignments

Actions

Enable Student Mode

- Assignment
- Answer
- Submissions
- Discussions
- Manage
- More ...

Description

No description available

| | Weight | Started | Completed | Passed | Mean (Median) | Due |
|---|------------------|---------|-----------|----------------|---------------|-----|
| Assignments <small>open</small> | 1.0 | 468 93% | 0 0% | no minimum set | 4.0 (x 4.5) | |
| Weekly Assignments <small>open</small> | 1.0 of total 3.0 | 461 99% | 0 0% | no minimum set | 3.4 (x 3.5) | |
| Practice exams <small>open</small> | 1.0 of total 3.0 | 316 68% | 0 0% | no minimum set | 2.3 (x 2.3) | |
| Exams <small>open</small> | 1.0 of total 3.0 | 398 85% | 391 98% | no minimum set | 7.9 (x 8.6) | |
| Drafts | 1.0 | | | | | |
| Exam question drafts | 1.0 | | | | | |

Statistics and aggregated grades were last updated 2 minutes ago. An update is scheduled which may take up to 10 minutes to complete.

- New Folder / Variants
- New Assignment
- Rearrange assignments

Management Discussion (0)

Latest Submission Comments (Aggregated)

Check for new comments (last checked: Thu, Jun 1, 2023 08:23:34)

This is what the initial page looks like to students (besides the draft folders)

Week 2

Enable Student Mode

Assignment Answer Submissions Discussions Manage More ...

Description

No description available

| | Weight | Started | Completed | Passed | Mean (Median) | Due |
|---|--------|--------------------------|-----------|----------------|---------------|-----|
| Week 2 open | A | 1.0 of total 9.0 395 86% | 218 55% | no minimum set | 5.1 (x 5.5) | |
| Chapter 3 - Structural testing open | A | 1.0 of total 3.0 353 89% | 257 73% | no minimum set | 4.3 (x 4.5) | |
| Chapter 4 - Designing contracts open | A | 1.0 of total 3.0 316 80% | 256 81% | no minimum set | 3.3 (x 3.8) | |
| Exam Practice open | A | 1.0 of total 3.0 355 90% | 312 88% | no minimum set | 9.3 (x 10.0) | |

Statistics and aggregated grades were last updated 13 hours ago, everything seems up to date for this assignment.

New Folder / Variants New Assignment Rearrange assignments

Exercises are divided by week

swapCase

Assignment Answer Submissions Discussions Manage More ...

Description

In this exercise, you will be testing the `swapCase` method.

This method swaps the case of a String using a word based algorithm:

- Upper case character converts to Lower case
- Lower case character converts to Upper case

Use **specification-based testing** to create a suitable test suite for this method.

This method was taken from the [Apache Commons Lang](#).

| | Weight | Started | Completed | Passed | Mean (Median) | Due |
|---------------|--------------------|----------|-----------|----------------|---------------|-----|
| swapCase open | A 1.0 of total 3.0 | 354 100% | 346 98% | no minimum set | 9.8 (9.10.0) | |

Statistics and aggregated grades were last updated 13 hours ago, everything seems up to date for this assignment.

```

Solution Test Library
14 * <li>Upper case character converts to Lower case
15 * <li>Other Lower case character converts to Upper case
16 * </ul>
17 *
18 * <p>
19 * Whitespace is defined by {@link Character#isWhitespace(char)}. A {@code null}
20 * input String returns {@code null}.
21 *
22 * @param str
23 *     the String to swap case, may be null
24 * @return the changed String, {@code null} if null String input
25 */
26 public static String swapCase(final String str) {
27     if (DelftWordUtilities.isEmpty(str)) {
28         return str;
29     }
30     final char[] buffer = str.toCharArray();
31     boolean whitespace = true;
32     for (int i = 0; i < buffer.length; i++) {
33         final char ch = buffer[i];
34         if (Character.isUpperCase(ch)) {
35             buffer[i] = Character.toLowerCase(ch);
36         } else if (Character.isLowerCase(ch)) {
37             buffer[i] = Character.toUpperCase(ch);
38         }
39     }
40     return new String(buffer);
41 }
42
43
44 /**
45 * Checks if a CharSequence is empty ("") or null. *
```

Inside a coding assignment (this is the method they need to test)

swapCase ✕ Not completed yet

Enable Student Mode

Submission Answer Submissions Discussions Manage More ...

Layout: III

Solution Test Library

Description

In this exercise, you will be testing the `swapCase` method.

This method swaps the case of a String using a word based algorithm:

- Upper case character converts to Lower case
- Lower case character converts to Upper case

Use **specification-based testing** to create a suitable test suite for this method.

This method was taken from the [Apache Commons Lang](#).

You write your tests in the Solution tab.

```
1 package delft;
2
3 import static org.assertj.core.api.Assertions.*;
4 import static org.junit.jupiter.api.Assertions.*;
5
6 import java.util.*;
7 import java.util.stream.*;
8 import org.junit.jupiter.api.*;
9 import org.junit.jupiter.params.*;
10 import org.junit.jupiter.params.provider.*;
11
12 class DelftWordUtilitiesTest {
13
14     @ParameterizedTest
15     @MethodSource("generator")
16     void swapCase(String description, String originalStr, String expected) {
17         assertThat(DelftWordUtilities.swapCase(originalStr)).isEqualTo(expected);
18     }
19
20     static Stream<Arguments> generator() {
21         Arguments tc1 = Arguments.of("str null", null, null);
22         Arguments tc2 = Arguments.of("str empty", "", "");
23         Arguments tc3 = Arguments.of("1 char upper", "A", "a");
24         Arguments tc4 = Arguments.of("1 char lower", "b", "B");
25         Arguments tc5 = Arguments.of("1 char whitespace", " ", " ");
26         Arguments tc6 = Arguments.of("multiple chars upper", "ABC", "abc");
27         Arguments tc7 = Arguments.of("multiple chars lower", "def", "DEF");
28         Arguments tc8 = Arguments.of("multiple chars whitespace", " ", " ");
29         Arguments tc9 = Arguments.of("lower case after whitespace", " k", " K");
30         Arguments tc10 = Arguments.of("multiple chars mixed", "AkDlc keCt", "kCdLc KeCt");
31         return Stream.of(tc1, tc2, tc3, tc4, tc5, tc6, tc7, tc8, tc9, tc10);
32     }
33 }
34
```

Console Discussion Revision History

Saved Only tests Analyse coverage Reset
Assess Get hints Mark Completed

Status: Running

Multiple running options

swapCase

Not completed yet

Enable Student Mode

Submission Answer Submissions Discussions Manage More ...

Layout

Solution Test Library

Description

In this exercise, you will be testing the `swapCase` method.

This method swaps the case of a String using a word based algorithm:

- Upper case character converts to Lower case
- Lower case character converts to Upper case

Use **specification-based testing** to create a suitable test suite for this method.

This method was taken from the [Apache Commons Lang](#).

```

1 package delft;
2
3 import static org.assertj.core.api.Assertions.*;
4 import static org.junit.jupiter.api.Assertions.*;
5
6 import java.util.*;
7 import java.util.stream.*;
8 import org.junit.jupiter.api.*;
9 import org.junit.jupiter.params.*;
10 import org.junit.jupiter.params.provider.*;
11
12 class DelftWordUtilitiesTest {
13
14     @ParameterizedTest
15     @MethodSource("generator")
16     void swapCase(String description, String originalStr, String expected) {
17         assertThat(DelftWordUtilities.swapCase(originalStr)).isEqualTo(expected);
18     }
19
20     static Stream<Arguments> generator() {
21         Arguments tc1 = Arguments.of("str null", null, null);
22         Arguments tc2 = Arguments.of("str empty", "", "");
23         Arguments tc3 = Arguments.of("1 char upper", "A", "a");
24         Arguments tc4 = Arguments.of("1 char lower", "b", "B");
25         Arguments tc5 = Arguments.of("1 char whitespace", " ", " ");
26         Arguments tc6 = Arguments.of("multiple chars upper", "ABC", "abc");
27         Arguments tc7 = Arguments.of("multiple chars lower", "def", "DEF");
28         Arguments tc8 = Arguments.of("multiple chars whitespace", " ", " ");
29         Arguments tc9 = Arguments.of("lower case after whitespace", " k", " K");
30         Arguments tc10 = Arguments.of("multiple chars mixed", "AkdLc keCtI", "aKdLc KeCtI");
31         return Stream.of(tc1, tc2, tc3, tc4, tc5, tc6, tc7, tc8, tc9, tc10);
32     }
33 }
34

```

Console Discussion Revision History

Saved Only tests Analyse coverage Assess Get hints Mark Completed Reset

Branch coverage: 12/12 (overall weight=0.10)
Mutation coverage: 10/10 (overall weight=0.10)
Meta tests: 12/12 (overall weight=0.80)
Final grade: 100/100

Super congrats!

```

/*(##%#*
//**/////|*|*/.
/+++++,.....,***#(
,###,.....,*,***/%
(/***,.....,***//
%/**,.....,*///|
(###,.....,##|,|/|
{(*,*,,|,/,...#/*#/#/##
**#/*,,...,**,.,./*,*,#/(
,.,./*,.....,*,*,|*,.,*/((
,.,(*,.,..*,./,/#/**#(##
,./#**/*,*,*(//|*#/.
|/**,*/#... ,*//*,##
*(|/,.....,*/#**,%(
%$*/*#, (#,.....,##
%/%%/|/|G%*.+.,**(@G%#
%/%/%/%#/#/%$G%G%#/#@G%G%#/%$G%
#%/%/%/%/%/%/%//**/|#%#G@G%G%G%/%$G
#%/%/%/%/%/%/%/%/%/#/%@G@G%#G%G%G%

```

Andy is running in PRACTICE mode and took 6.2 seconds to ...
Files were generated during this execution:
jaccoco/index.html
pitest/index.html

You see the results!

Challenges in building such a tool



- **Make it performant**
 - First version would simply do "mvn test coverage pitest static-analysis" and then parse the results. Too slow.
 - Makes native use of JaCoCo, Pitest and JUnit.
 - Around 2 seconds in my M1, around 5 seconds in AWS lambda. More in complex PBT assignments.
- **Too much classloading magic and shadowing**
 - All these tools bring their own classloaders, putting all of them together was tricky
 - Pitest shadows JUnit, it's hard to bump dependency versions.
- **Create a DSL for configuration that's not too hard to use**
 - First version was a simple text file with conventions. Became too hard to express everything.
 - Java version is a bit better although requires some knowledge to use it.
 - Trying YAML (as anyone, I'm sure we'll regret that soon enough)
- **The Maven plugin**
 - Due to the so much magic we do, implementing a Maven plugin was harder than we thought.
 - Things would fail, classes wouldn't be found, etc...

Ongoing and next steps

~~● Support Selenium tests~~

- Unit tests, mocking, property-based tests, SQL tests, Selenium tests.

~~● Analytics~~

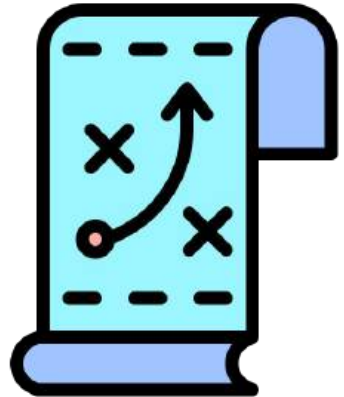
~~● Performance improvements~~

● Make it easy for others to adopt

- You can use WebLab for 5 euros per student
- Maven plugin
- AWS lambda if you want to embed in your IDE
- Github, help me!!! (Challenge: get approved by Delft for exams)

● Write a paper!

- I leave academia, but academia doesn't leave me!



Based on my book!

Learn how to write tests in an effective and systematic way!

- ✓ Create tests that really find bugs
- ✓ Use code coverage in the right way
- ✓ When to go for unit, integration or system testing
- ✓ Use mocks and stubs to simplify your unit testing
- ✓ Write property-based tests
- ✓ Design testable systems
- ✓ Write maintainable test code



TU Delft education fellow



Mauricio Aniche

We already know by now the importance of programming skills. But whenever we think of becoming programmers, we think of learning a programming language or algorithms and data structures. However, a skill that is as important as writing programs is to test those programs, and make sure that they actually work. After all, society depends on working software!

The goal of my fellowship is to develop a platform fully focused on software testing education. Software testing education offers lots of challenges for both teachers and students. In this platform, students will exercise different testing skills, such as writing automated test cases that achieve a certain level of coverage, chasing bugs in real-world programs, and developing code in a test-driven manner. Teachers will also leverage the analytics data to understand how their students are doing. The platform will also keep proposing new exercises based on real-world open source bugs.

Thanks to the entire team!

- Martin Mladenov, the hand!
- Jan Warchocki
- Paul Hübner
- Florena Buse
- Thijs Nulle
- Diman Uzunov
- Nadine Kuo
- Teodor Oprescu
- Yoon Hwan Jeong
- Wouter Polet (earlier versions)
- The WebLab development team

Why is it called Andy?



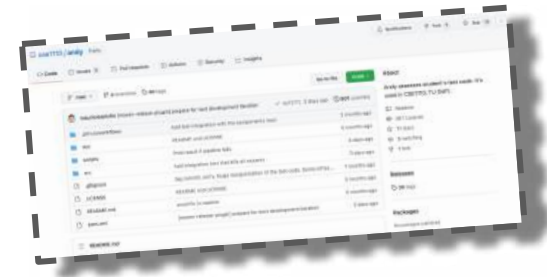
It is a tribute to **Dr. Andy Zaidman**, Professor in Software Quality and Director of Studies at TU Delft.


Andy is a strong proponent of testing. Students meet Andy in their first quarter. Andy teaches them introduction to programming. In weeks 3-4 (!!), Andy already shows them JUnit!

<https://azaidman.github.io/>

Questions?

- Andy: <https://github.com/cse1110/andy>
- Manual: README in Andy's repository



Give us a  !

Maurício Aniche

mauricioaniche@gmail.com

@mauricioaniche

