Ciências ULisboa | Informática

VVS (2022/2023)
Assignment 1

1 Introduction

This assignment goal is to prepare a set of unit tests to verify an implementation of a Trie data structure, a tree for fast insertion and retrieval of key-value pairs. In this context, the SUT to analyse is a variant using ternary trees.

Please read section 5.2 from Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne, for details about this data structure[1]. Import maven project `vvs_assignment1` to access the source code implementing Tries.

2 Requirements

The tests you should include must satisfy the requirements produced by the following criteria:

1. Line and Branch Coverage for each public method.
2. Edge-Pair Coverage and Prime Path Coverage for method `longestPrefixOf`
3. All-Du-Paths Coverage for method `longestPrefixOf`
4. All-Coupling-Use-Paths Coverage for private method `put`
5. Select and apply one Logic-based test coverage for method `longestPrefixOf`, justify your option

For method `put`, include a test set based on Input State Partitioning, namely Base Choice Coverage, using the following characteristics:

1. Trie already includes the new key
2. Trie already includes some prefix of the new key
3. Trie is empty
4. The new key is the smallest/largest/a typical key (in lexicographic terms)

Verify your test set via program mutation using PIT. Write a report comparing the mutation coverage achieved by each criteria for method `longestPrefixOf`. Then, if needed, add new tests to improve mutation coverage.

Use JUnit QuickCheck to create a Trie random generator. Include an `equals()` and a `delete()` method in the SUT class in order to test the following properties:

1. The order of insertion of different keys does not change the final tree value
2. If you remove all keys from a tree, the tree must be empty

[1] There are video explanations by the author in YouTube: `https://youtu.be/el9-Vb6iNPM`, `https://youtu.be/CqE9MyOErRI`, `https://youtu.be/TdP3ty3w9_A`

3. Given a tree, inserting and then removing the same key value will not change its initial value

4. Selecting a stricter prefix `keysWithPrefix` returns a strict subset result. Eg, `keysWithPrefix("sub")` $\subseteq$ `keysWithPrefix("su")`

For each coverage criteria paradigm, include its test cases in different classes or even different packages (if you have several classes per paradigm), all of them properly named. There is no problem if you repeat test cases between packages.

For each unit test include appropriate Java comments that detail what test requirements it satisfies.

3 Deliver

Upload a zip with your work into the course's moodle webpage until 23:59 of April 24. The zip file should be named `vvs01_XXXXX.zip` where `XXXXX` is your student number.

The zip must include:

- the maven project (with just the source code and `pom.xml`, do not include binaries or Eclipse metadata)
- a pdf report with your analysis of the several coverage criteria

4 Academic Integrity

As a future IT professional, we expect from you an irreproachable attitude, in both ethical and moral terms. Be specially attentive in observing and enforcing the laws of computer crime. Students caught in fraud or plagiarism situations (plagiarists and plagiarised) in assignments or exams automatically fail to complete the course and will be prosecuted, leading to a process that may end up in suspension or in abandoning the university.