Perago Technical Evaluation

## 1 Objective

The goal of this exercise is to create a library that provides:

1. A diff service that:

* can calculate the differences between two objects and return the resulting "diff"
* can apply a previously created "diff" against an original object, so that the returned object matches the modified object that was used to calculate the diff.

2) A diff renderer component that can display a "diff" in a hierarchical, human-readable format

## 2 Specification

1) The diff service must implement the DiffEngine interface (see provided source)

2) The diff renderer must implement the DiffRenderer interface (see provided source)

Your solution must use the above interfaces as provided. **DiffEngine.java and DiffRenderer.java must not be modified.**

You are free to implement the details of the Diff and DiffException classes as you see fit (within the constraints specified by the requirements below)

## 3 Detailed Requirements and Examples

The diff service and diff renderer must satisfy the below requirements, and should be capable of handling the illustrative examples. **Please also provide unit-tests that verify your implementation**.

List any assumptions you made that the correctness of your implementation depends on.

It is not necessarily expected that you will be able to satisfy all of the requirements in the allocated time. However, the more complete your implementation, the more favourably it is likely to be scored. For any requirements that you are unable to complete in the allocated time, please explain what made implementing them challenging, and how you could possibly satisfy them if you had more time.

### Requirement 1

* Diffs must reflect all information that was created/updated/deleted. Information that was not changed must not be reflected in a Diff.
* A Diff object must not contain a reference to the modified object or a reference to a clone of the modified object

### Requirement 2

* diffService.calculate() and diffService.apply() must not modify or change the objects that were passed in as their parameters.

### Requirement 3

* A null original object diffed against non-null modified object must reflect as being “created” in diff.
* A non-null original object diffed against null modified object must reflect as being “deleted” in diff
* Non-null original object diffed against non-null modified object must reflect as being “updated” if there are properties of original whose values differ from the properties of modified

### Requirement 4

* Properties of original must in reflected as “updated” if their value differs from that of modified

### Requirement 5

* Diffs must recursively reflect modifications to all child objects

### Requirement 6

* Properties of original or modified that are collections (arrays, sets, lists, maps etc.) must be taken into consideration when calculating and applying diffs

### Requirement 7

* Cyclic relationships in original or modified should be taken into consideration when calculating and applying diffs

### Examples

In all examples below, the value of the “diff” argument passed to diffRenderer is assumed to be the output of diffService.calculate(original,modified).

For all examples, the result returned from diffService.apply(original,diff) must be equal to modified. i.e.

* If modified == null , then diffService.apply(original,diff) == null
* otherwise, modified.equals(diffService.apply(original,diff)) == true

Please see the provided source code for the Person class used in the examples.

|  |  |  |
| --- | --- | --- |
| **Original** | **Modified** | **Expected output of diffRenderer.render(diff)** |
| Null | |  |  | | --- | --- | | Person | | | firstName | “Fred” | | surname | “Smith” | | friend | Null | | pet | Null | | nickNames | Null | | 1 Create: Person   * 1. Create: firstName as “Fred”   2. Create: surname as “Smith”   3. Create: friend as null   4. Create: pet as null   5. Create: nickNames as null |
| |  |  | | --- | --- | | Person | | | firstName | “Fred” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | null | | Null | 1 Delete: Person |
| |  |  | | --- | --- | | Person | | | firstName | “Fred” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | null | | |  |  | | --- | --- | | Person | | | firstName | “Fred” | | surname | “Jones” | | friend | null | | pet | null | | nickNames | null | | 1 Update: Person   * 1. Update: surname from “Smith” to Jones” |
| |  |  | | --- | --- | | Person | | | firstName | “Fred” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | null | | |  |  | | --- | --- | | Person (modified) | | | firstName | “Fred” | | surname | “Smith” | | friend |  | | pet | null | | nickNames | null |  |  |  | | --- | --- | | Person | | | firstName | “Tom | | surname | “Brown” | | friend | null | | pet | null | | nickNames | null | | “1 Update: Person   * 1. Update: friend      1. Create: Person         1. Create: firstName as “Tom”         2. Create: surname as “Brown”         3. Create: friend as null         4. Create: pet as null         5. Create: nickNames as null” |
|  |  |  |
| |  |  | | --- | --- | | Person (original) | | | firstName | “Fred” | | surname | “Smith” | | friend |  | | Pet |  | | nickNames | null |  |  |  | | --- | --- | | Pet | | | type | “Dog” | | name | “Rover” |  |  |  | | --- | --- | | Person | | | firstName | “Tom” | | surname | “Brown” | | friend | null | | nickNames | null | | |  |  | | --- | --- | | Person (modified) | | | firstName | “Fred” | | surname | “Jones” | | friend |  | | Pet |  | | nickNames | null |  |  |  | | --- | --- | | Person | | | firstName | “Jim” | | surname | “Brown” | | friend | null | | nickNames | null |  |  |  | | --- | --- | | Pet | | | type | “Dog” | | name | “Spot” | | 1 Update: Person   * 1. Update: surname from “Smith” to “Jones   2. Update: friend      1. Update: firstName from “Tom” to “Jim”   3. Update: pet      1. Update: name from “Rover” to “Spot” |
| |  |  | | --- | --- | | Person (original) | | | firstName | “Fred” | | surname | “Smith” | | friend |  | | pet | null | | nickNames | null |  |  |  | | --- | --- | | Person | | | firstName | “Tom” | | surname | “Brown” | | friend |  | | pet | null | | nickNames | null | | |  |  | | --- | --- | | Person (modified) | | | firstName | “Fred” | | surname | “Smith” | | friend |  | | pet | Null | | nickNames | null |  |  |  | | --- | --- | | Person | | | firstName | “Jim” | | surname | “Brown” | | friend |  | | pet | null | | nickNames | null | | 1 Update: Person  1.1 Update: friend  1.1.1 Update: firstName from “Tom” to “Jim” |
| |  |  | | --- | --- | | Person (original) | | | firstName | “Fred” | | surname | “Smith” | | friend |  | | pet | null | | nickNames | null |  |  |  | | --- | --- | | Person | | | firstName | “Tom” | | surname | “Brown” | | friend |  | | nickNames | null | | |  |  | | --- | --- | | Person (modified) | | | firstName | “John” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | null | | 1 Update: Person   * 1. Update: firstName from “Fred” to “John”   2. Update: friend      1. Delete: Person |
| |  |  | | --- | --- | | Person (original) | | | firstName | “Fred” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | {“scooter”, “biff”} | | |  |  | | --- | --- | | Person (modified) | | | firstName | “Fred” | | surname | “Smith” | | friend | null | | pet | null | | nickNames | {“biff”, “polly”} | | 1 Update: Person  1.1 Update nickNames from {“scooter”, “biff”} to {“biff”, “polly”} |