

1 Little Use-case-Analysis for existing Perfidix Version

Perfidix was working in version 1.0. With this framework already existing, some specific use-cases for the next version of perfidix with annotations were made.

1.1 Specific "setUp" and "tearDown" for each bench-method

Often, the preparation of a bench differ from the preparation of another bench. So there was a need for specifying different "setUp" and "tearDown" methods for different benches.

1.2 Bench-methods should only run a given number of times

Because of the different execution time of the bench-methods, it was useful to define the number of times a bench-method should run.

2 Existing Perfidix

Perfidix at version 1.0 worked with inheritance and reflection. A benchmark was implemented in the following way for example:

Listing 1: Perfidix 1.0 Class

```
1 public class SomeBenchmark extends Benchmarkable {
2     CompressedHandler c;
3     SimpleFileHandler s;
4
5     //Constructor, in fact useless
6     public SomeBenchmark() {
7         c = new CompressedHandler();
8         s = new SimpleFileHandler();
9     }
10
11     //setUp, invoked before each run of each bench-method
12     public void setUp() {
13         super.setUp();
14         c = new CompressedHandler();
15         s = new SimpleFileHandler();
16     }
17
18     //tearDown, invoked after each run of each bench-method
19     public void tearDown() {
20         c = null;
21         s = null;
22         super.tearDown();
23     }
24
25     //bench 1
26     public void benchCWrite() {
27         c.write("hello_world");
28     }
29
30     //bench 2
31     public void benchSWrite() {
32         s.write("hello_world");
33     }
```

```

34
35 //bench 3
36 public void benchCRead() {
37     c.read();
38 }
39
40 //bench 4
41 public void benchSRead() {
42     s.read();
43 }
44
45 }

```

After implementing the bench, it can be invoked for example in the following way:

Listing 2: Invoking the bench

```

1 .....
2 final Benchmark bench = new Benchmark();
3 bench.add(new SomeBenchmark());
4 bench.run(100);
5 .....

```

In this example, the upper class is build and benched over 100 times.

3 Perfidix Annotations

Perfidix, which is currently working with reflection and inheritance, is moved to annotations. For this, a small description of possible annotations is listed below. All methods annotated with the following metadata should be parameter and returnvalue free. Each annotation mustn't occur more than one time in a class, except the *Bench* annotation:

3.1 ”@BenchClass”

The annotation has to be placed before the class declaration. The number of runs of each void-method in this class, except the methods annotated below, can be set as a parameter.

3.1.1 ”@ BenchClass(runs=)”

The number of runs of each bench-method can be set here. This setting can be overridden by each method with an extra *Bench* annotation..

3.2 ”@BeforeBenchClass”

A method with this annotation is called before the first bench-method but just once per class.

3.3 ”@BeforeBenchMethod”

A method with this annotation is called before the bench-method but just once for all runs.

3.4 ”@BeforeBenchRun”

A method with this annotation is called before the bench-method before each run. This method can be overridden by a method defined in a *Bench* annotation for bench-specific calling. In this case, a method with this annotation is ignored.

3.5 ”@Bench”

This annotation marks a method which should be benched. The method itself should be parameter-free. Perfidix-specific settings can be made through parameter of the annotation

3.5.1 ”@Bench(beforeMethod=)”

Here, a specific setUp method for this specific bench-method can be defined.

3.5.2 ”@Bench(afterMethod=)”

Here, a specific tearDown method for this specific bench-method can be defined.

3.5.3 ”@Bench(runs=)”

The number of runs of each bench-method can be set here. This setting overrides the global setting of a possible *BeforeBench* annotation of the corresponding class.

3.6 ”@AfterBenchRun”

A method with this annotation is called after the bench-method before each run. This method can be overridden by a method defined in a *Bench* annotation for bench-specific calling. In this case, the method with this annotation is ignored.

3.7 ”@AfterBenchMethod”

A method with this annotation is called after the bench-method but just once for all runs.

3.8 ”@AfterBenchClass”

A method with this annotation is called after the last bench-method but just once per class.

4 Example Use Cases

Perfidix 2.0 offers a very flexible usage based on annotation. The examples contains all the same usecase but are different implemented. A compressed file access is compared to a normal file access.

4.1 Example 1

The code in Listing 1 and Listing 3 are doing exact the same. The *setUp()* and *tearDown()* methods are invoked before each run of each method. But with Perfidix 2.0 we can do much more.

Listing 3: Perfidix 2.0

```
1 public class SomeAnnoBenchmark {
2     CompressedHandler c;
3     SimpleFileHandler s;
4
5     //setUp, invoked before each run
6     @BeforeBenchRun
7     public void setUp() {
8         c = new CompressedHandler();
9         s = new SimpleFileHandler();
10    }
11
12    //tearDown, invoked after each run
13    @AfterBenchRun
14    public void tearDown() {
15        c = null;
16        s = null;
17    }
18
19    //bench Method 1
20    @Bench
21    public void benchCWrite() {
22        c.write("hello_world");
23    }
24
25    //bench Method 2
26    @Bench
27    public void benchSWrite() {
28        s.write("hello_world");
```

```

29         }
30
31         //bench Method 3
32         @Bench
33         public void benchCRead() {
34             c.read();
35         }
36
37         //bench Method 1
38         @Bench
39         public void benchSRead() {
40             s.read();
41         }
42     }

```

4.2 Example 2

In Listing 4 you see the usage of specific `setUp` and `tearDown` methods. These methods have the same behaviour than methods with the *BeforeBenchmark* annotation.

Listing 4: Perfidix 2.0

```

1 public class SomeSpecificSetUpTearDownBenchmark {
2     CompressedHandler c;
3     SimpleFileHandler s;
4
5     //setUp for benchCRead/benchCWrite. Invoked via @Bench-params
6     public void setUpCompressed() {
7         c = new CompressedHandler();
8     }
9
10    //tearDown for benchCRead/benchCWrite. Invoked via @Bench-params
11    public void tearDownCompressed() {
12        c = null;
13    }
14
15    //setUp for benchSRead/benchSWrite. Invoked via @Bench-params
16    public void setUpSimple() {
17        s = new SimpleFileHandler();
18    }
19

```

```

20 //tearDown for benchSRead/benchSWrite. Invoked via @Bench-params
21 public void tearDownSimple() {
22     s = null;
23 }
24
25 //bench Method 1
26 @Bench(beforeMethod="setUpCompressed"
27     ,afterMethod="tearDownCompressed")
28 public void benchCWrite() {
29     c.write("hello_world");
30 }
31
32 //bench Method 2
33 @Bench(beforeMethod="setUpSimple"
34     ,afterMethod="tearDownSimple")
35 public void benchSWrite() {
36     s.write("hello_world");
37 }
38
39 //bench Method 3
40 @Bench(beforeMethod="setUpCompressed"
41     ,afterMethod="tearDownCompressed")
42 public void benchCRead() {
43     c.read();
44 }
45
46 //bench Method 4
47 @Bench(beforeMethod="setUpSimple"
48     ,afterMethod="tearDownSimple")
49 public void benchSRead() {
50     s.read();
51 }
52
53 }

```

4.3 Example 3

In Listing 5 the same Bench is a little bit modified:

First of all, the class-annotation *BenchClass* with the param *runs* is used. That means that every method which is parameter-free and is not annotated with

a `setUp` / `tearDown` annotation, is benched 10 times, except the `benchSWrite` method, which has an extra `Bench` annotation with a run parameter. This method is benched 60 times.

Additional to that, every possible `setUp` and `tearDown` method is used in this example. A description is given in the code and in Section 3.

Listing 5: Perfidix 2.0

```

1 @BenchClass( runs=10)
2 public class ClassAnnoBenchmark {
3
4     CompressedHandler c;
5     SimpleFileHandler s;
6
7     String toTest;
8     long testLength;
9
10    //classwide setUp, invoked just one time, just setting the length
11    @BeforeBenchClass
12    public void beforeClass() {
13        Math.abs(testLength = new Random().nextInt(100));
14    }
15
16    //methodWide setUp, invoked just one time per method, building a
17    @BeforeBenchMethod
18    public void beforeMethod() {
19        for(int i = 0; i<testLength; i++) {
20            toTest = toTest + (char)(new Random().nextInt(CL
21        }
22    }
23
24    //normal setUp, invoked one time per method per run, instantiating
25    @BeforeBenchRun
26    public void beforeRun() {
27        c = new CompressedHandler();
28        s = new SimpleFileHandler();
29    }
30
31    //normal tearDown, invoked one time per method per run, removing
32    @AfterBenchRun
33    public void afterRun() {
34        c = null;

```

```

35         s = null;
36     }
37
38     //methodWide tearDown, invoked just one time per Method, resetting
39     @AfterBenchMethod
40     public void afterMethod(){
41         toTest = null;
42     }
43
44     //classwide tearDown, invoked just one time, resetting the length
45     @AfterBenchClass
46     public void afterClass() {
47         testLength = -1;
48     }
49
50     //bench 1, invoked because of class-annotation
51     public void benchCWrite() {
52         c.write("hello_world");
53     }
54
55     //bench 2, invoked because of method-annotation
56     @Bench(runs=60)
57     public void benchSWrite() {
58         s.write("hello_world");
59     }
60
61     //bench 3, invoked because of class-annotation
62     public void benchCRead() {
63         c.read();
64     }
65
66     //bench 4, invoked because of class-annotation
67     public void benchSRead() {
68         s.read();
69     }
70
71
72 }

```

The benches were invoked in the following way:

Listing 6: Perfidix 2.0

```
1  ....
2  Benchmark b = new Benchmark("Compressed_vs.Simple");
3  b.add(new SomeAnnoBenchmark());
4  b.add(new SomeSpecificSetUpTearDownBenchmark());
5  b.add(new ClassAnnoBenchmark());
6  Result r = b.run(32);
7  ....
```

All examples are avliable in the perfidix project in the example folder.