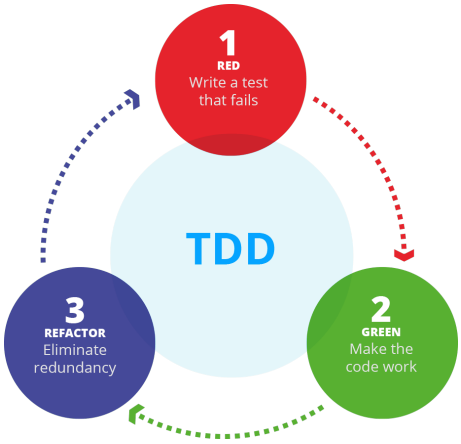
TDD: Test-Driven Development



1. In a "core" package, write the class FizzBuzz with a static method fizzBuzz taking an integer as a parameter and returning a String. The method body should only return null.
2. Create a "test" source folder, similar to "src," with a "core" package (mirror of packages between source classes and test classes).
3. Select the class and generate a test class.

   With Eclipse: Right-click New > JUnit Test Case

4. Notice that the default generated test automatically fails. Execute this test and ensure that it fails.
5. Write an initial test method like this and execute it. It should fail.

```
@Test
public final void testAvecValeurUn(){
    String res = FizzBuzz.fizzBuzz(1);
    assertEquals("1", res, "Problem with the number 1");
}
```

6. Write the simplest and shortest code to make the test pass.
7. Write an additional test method (which should fail) to verify that fizzBuzz of 2 equals 2.
8. Write the simplest and shortest code to pass this test and the first one.
9. Refactor the code to generalize for all possible values.
10. Write an additional test method to verify that fizzBuzz of 3 equals "Fizz."
11. Write an additional test method to verify that fizzBuzz of different multiples of 3 equals "Fizz."
12. Write the simplest code to pass this new test.
13. Since we have multiple test cases differing only in the data set and oracle, we will use test class parameterization. Modify your code as follows:

```
@ParameterizedTest
@CsvSource({
    "1, 1",
    "2, 2",
    "3, Fizz",
    "4, 4",
    "6, Fizz",
    "9, Fizz"
})
public final void testFizzBuzz(int input, String expected){
    String res = FizzBuzz.fizzBuzz(input);
    assertEquals(expected, res, "!!!");
}
```

14. Execute the test class and observe how different data sets are utilized.
15. Add a data set for fizzBuzz of 5 to be "Buzz": this test should fail.
16. Modify the code and pass this new test.
17. Test with other values multiples of 5.
18. Write the simplest code to pass these new tests.
19. Add a data set for fizzBuzz of 15 to be "Fizz Buzz": this test should fail.
20. Modify the code and pass this new test.
21. Add the following test for fizzBuzz of 0 to throw the "IllegalArgumentException" exception.

```
@Test
public final void testAvecZero(){
    Throwable exception = assertThrows(Throwable.class, () -> {
        FizzBuzz.fizzBuzz(0);});
    assertEquals(IllegalArgumentException.class, exception.
        getClass());
}
```

22. Modify the code and pass this new test.