Test driven development
TDD

TDD rules RED -> GREEN -> (REFACTOR?)

1. You are not allowed to write any production code unless it is to make a failing unit test pass.

2. You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.

3. You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

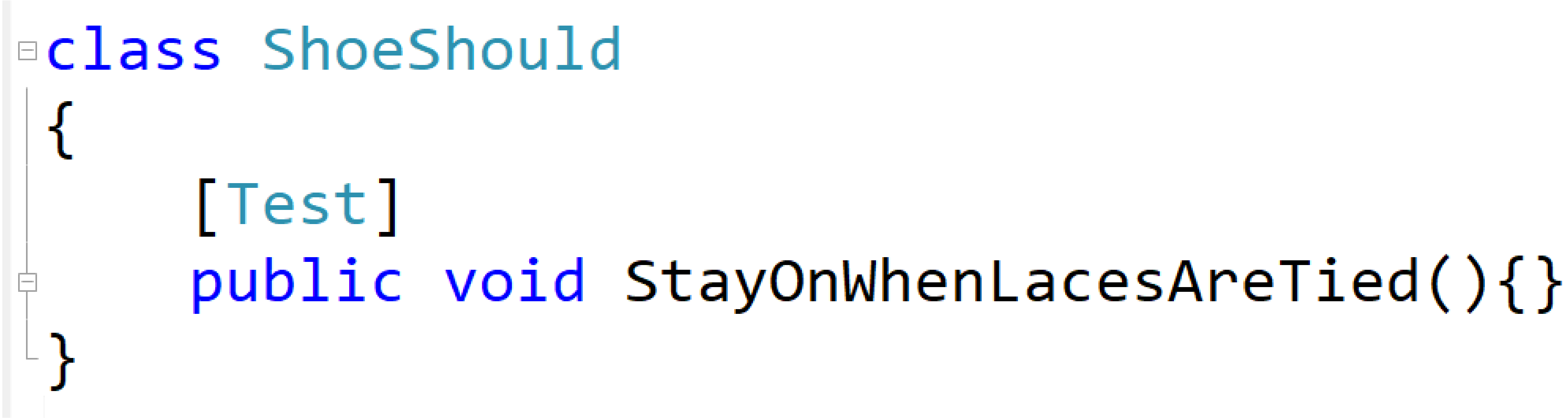
Robert C. Martin (Uncle Bob), “The three rules of TDD”

Approaches To A Failing Test

- Obvious implementation.
- Fake implementation.
- Triangulation

A Good Test Naming Convention

Avoid mention of method names and other implementation details



```
class ShoeShould
{
    [Test]
    public void StayOnWhenLacesAreTied(){}
}
```

Some Habits

- Test only one behavior at a time.
- Make sure test fails for right reason
- Extract duplicates in refactoring phase when you see them 3 times

  Keep your code DRY (Do not Repeat Yourself)
- Write test with arrange, act and assert sections.

  Given – when – then
- Keep tests isolated and repeatable

  Avoid flaky tests that might fail or pass.