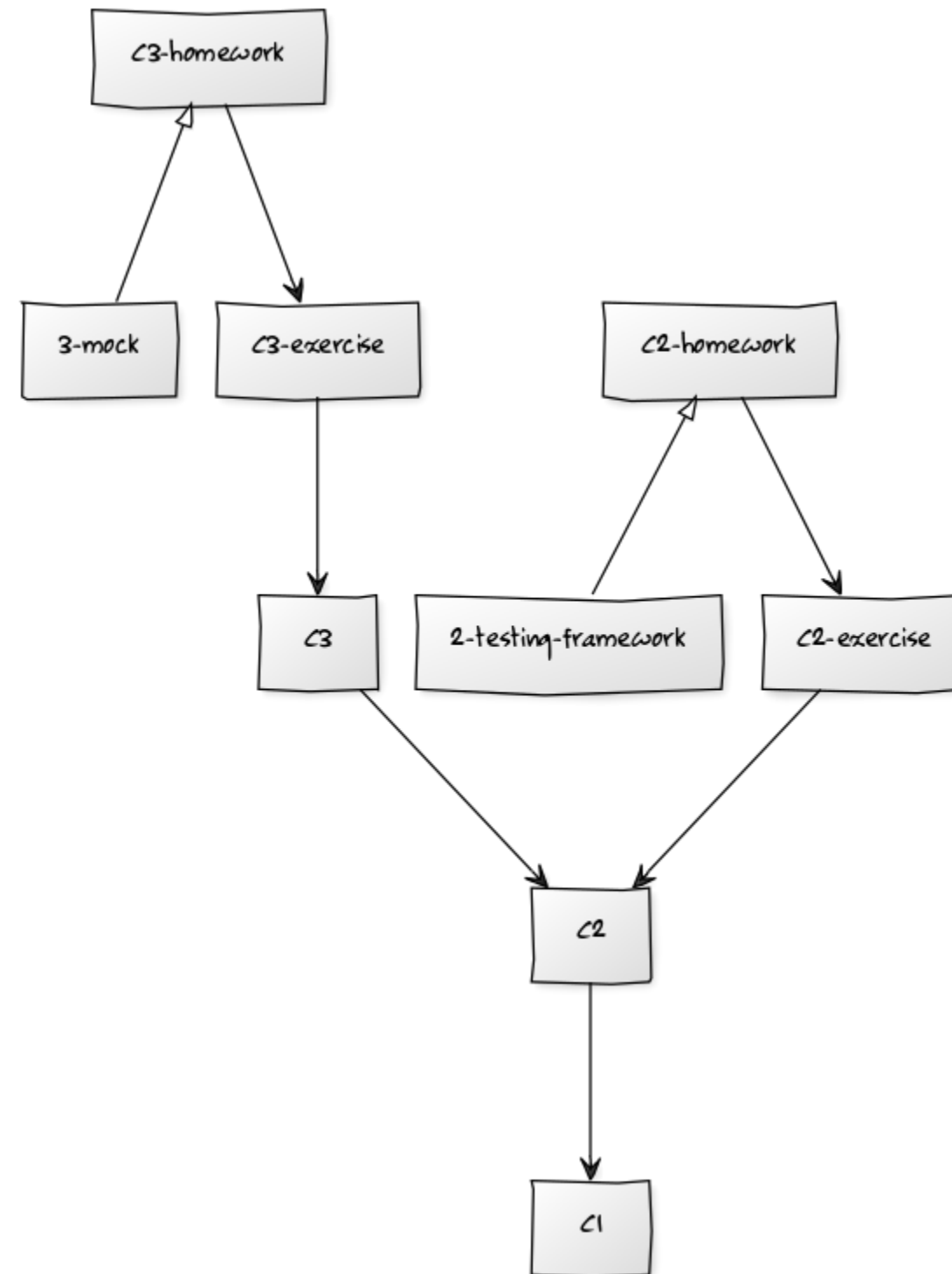


HOMEWORK REVIEW

获取基础代码

- `$ git fetch upstream master`
- `$ git checkout -b 3-mock upstream/master`



PUSH.DEFAULT

- matching
- simple

- `$ git config --global push.default simple`
- `$ git push --set-upstream origin your-branch-name`

TEST-DRIVEN DEVELOPMENT (TDD)

FIZZBUZZWHIZZ

1. 你首先说出三个不同的特殊数，要求必须是个位数，比如3、5、7。
2. 让所有学生拍成一队，然后按顺序报数。
3. 学生报数时，如果所报数字是第一个特殊数（3）的倍数，那么不能说该数字，而要说Fizz；如果所报数字是第二个特殊数（5）的倍数，那么要说Buzz；如果所报数字是第三个特殊数（7）的倍数，那么要说Whizz。
4. 学生报数时，如果所报数字同时是两个特殊数的倍数情况下，也要特殊处理，比如第一个特殊数和第二个特殊数的倍数，那么不能说该数字，而是要说FizzBuzz, 以此类推。如果同时是三个特殊数的倍数，那么要说FizzBuzzWhizz。
5. 学生报数时，如果所报数字包含了第一个特殊数，那么也不能说该数字，而是要说相应的单词，比如本例中第一个特殊数是3，那么要报13的同学应该说Fizz。如果数字中包含了第一个特殊数，那么忽略规则3和规则4，比如要报35的同学只报Fizz，不报BuzzWhizz。

TASKING

- ~~重构当前代码，支持增加新的规则~~
- 增加新的规则

第一步做什么？

目标：CLEAN CODE THAT WORKS

OVERVIEW

- a software development process
- 极限编程
- by Kent Beck

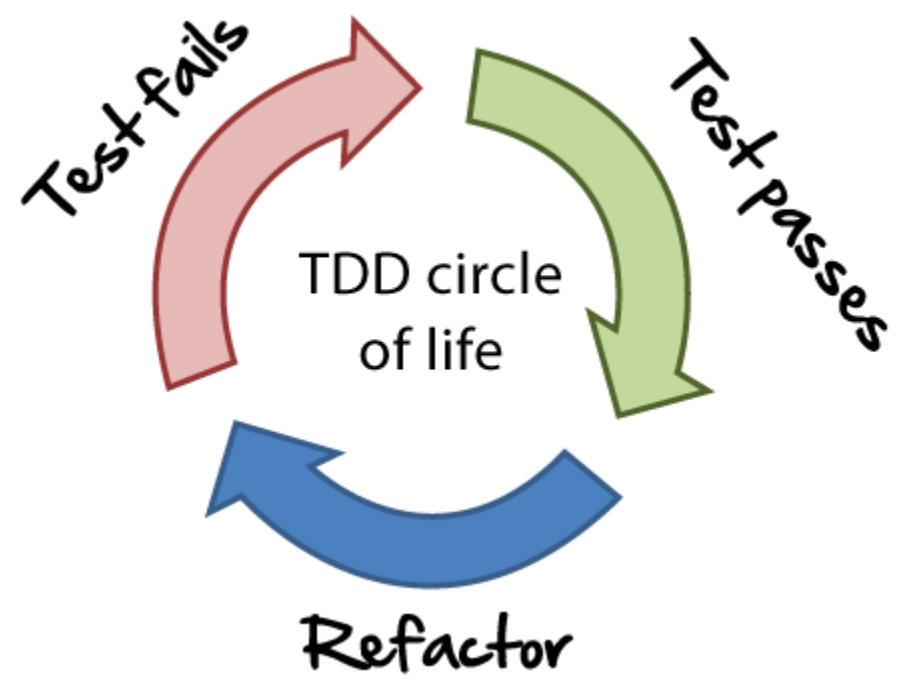
测试驱动开发的规则

- 只有在测试失败时，我们才写新的代码
- 消除重复

这意味着

- 用测试来驱动开发
- 一边写测试一边开发
- 开发环境可以对小改动快速响应
- 为方便测试，代码必须高内聚低耦合

步骤



动机是什么？

TDD是一种可以在开发过程中控制忧虑感的开发方法

担忧的负面影响

- 让你犹豫不决
- 不愿意与别人交流
- 羞于面对反馈
- 脾气暴躁

我们需要

- 尽快开始动手
- 更多参与沟通与交流
- 主动寻求反馈
- 改掉脾气暴躁（这个还是要靠自己）

更新代码

程序的输入

```
public static void main(String[] args) {  
    final GameInput gameInput = new GameInput(3, 5, 7);  
    final List<GameRule> rules = Lists.newArrayList(new ContainGameRule(gameInput.getNumber1(), gameInput.getNumber2(), gameInput.getNumber3()));  
    for (int i = 1; i <= 100; i++) {  
        System.out.println(new Student(rules, i).say());  
    }  
}
```

EXERCISE 1

- when sbt "run 4 5 6" student 4 should say Fizz
- when sbt "run 3 5" it should throw RuntimeException

BEHAVIOR-DRIVEN DEVELOPMENT (BDD)

- a software development process
- developed by Dan North
- as a response to the issues encountered teaching TDD

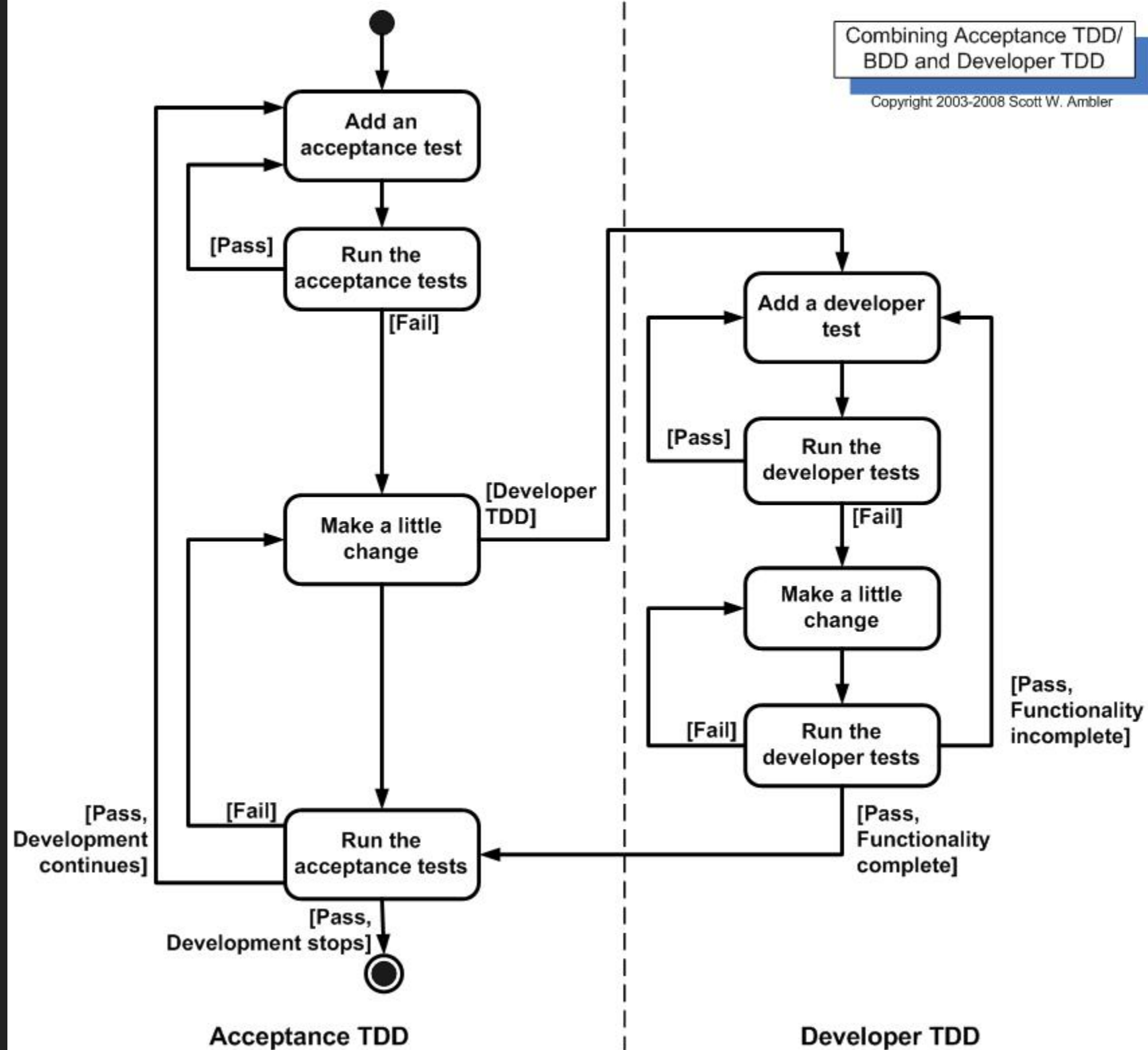
- Test method names should be sentences
- A simple sentence template keeps test methods focused
- An expressive test name is helpful when a test fails
- Behaviour is a more useful word than test
- given-when-then template
 - Given some initial context (the givens)
 - When an event occurs
 - then ensure some outcomes

2 LEVELS OF TDD

- Acceptance TDD
- Developer TDD

Combining Acceptance TDD/ BDD and Developer TDD

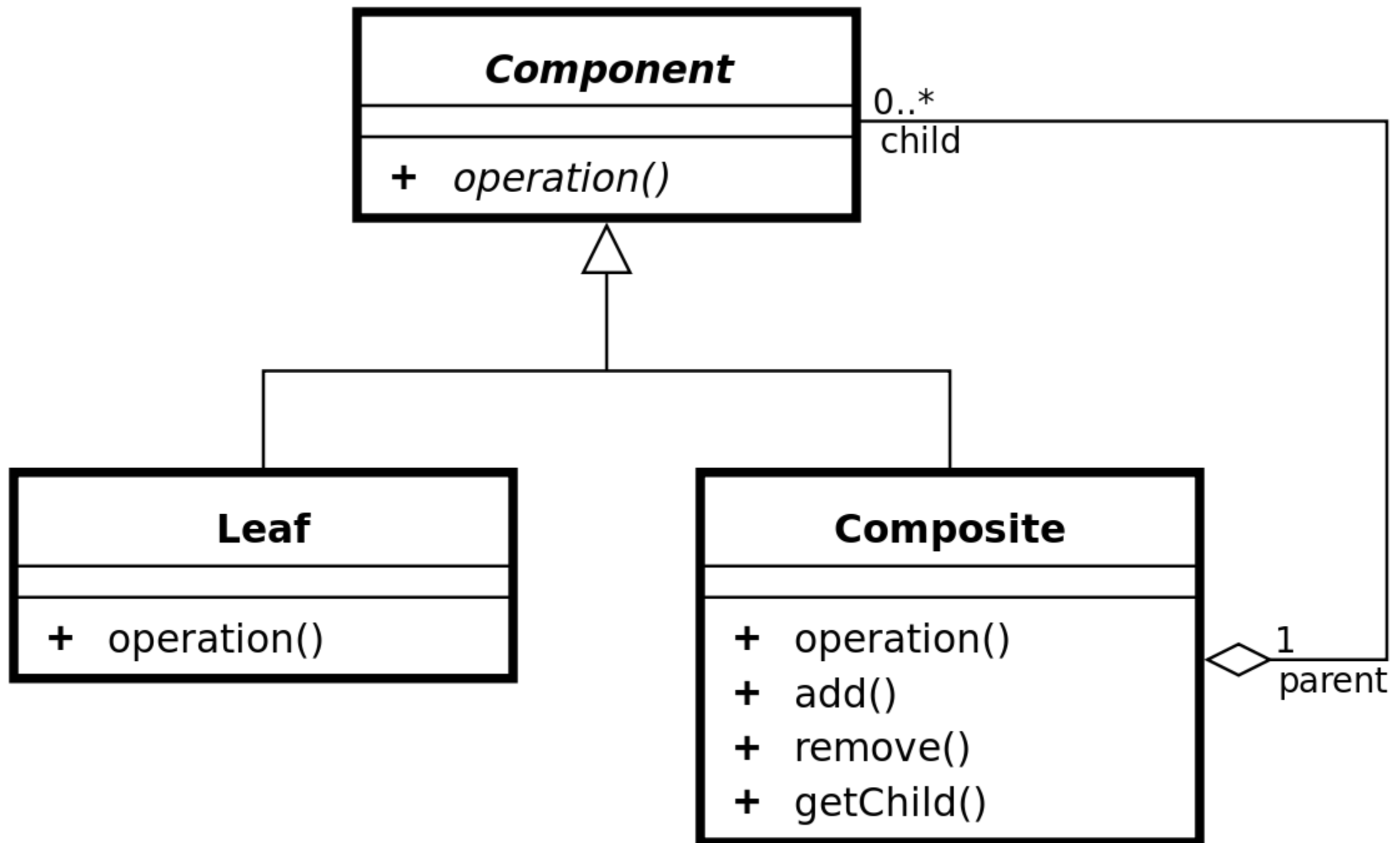
Copyright 2003-2008 Scott W. Ambler



EXERCISE 2

- ~~重构当前代码，支持增加新的规则~~
- 增加新的规则

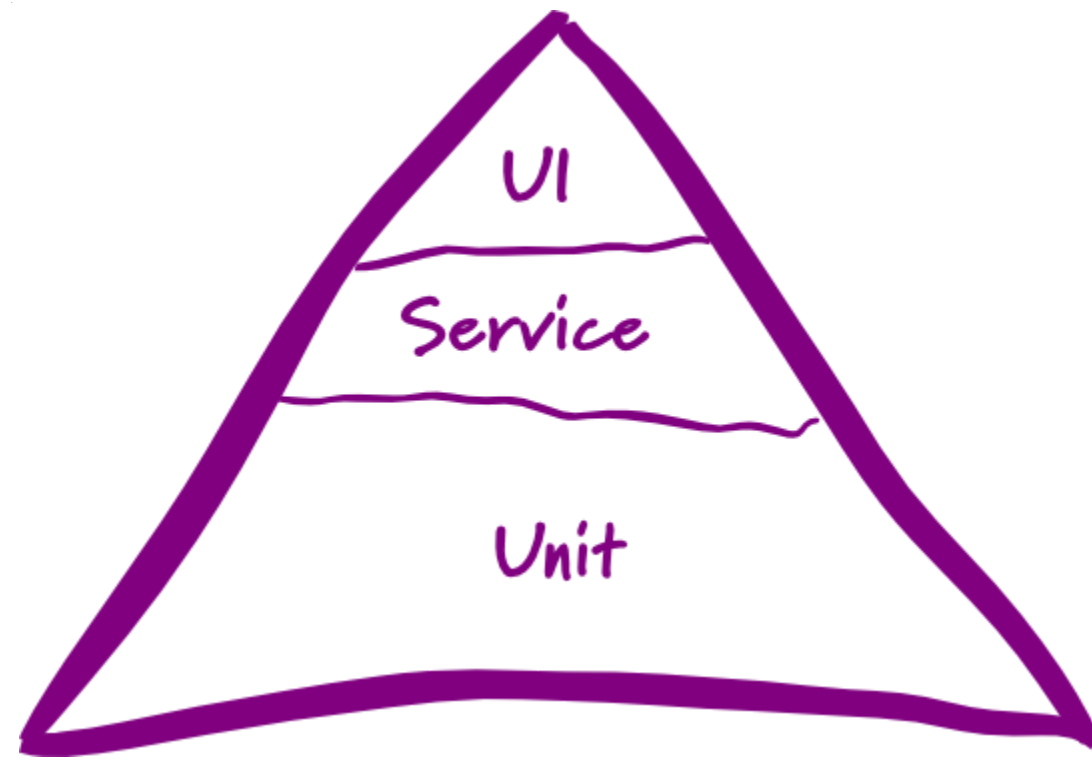
COMPOSITE PATTERN



EXERCISE 3

- Refactor Student

测试金字塔



回顾

`mock(GameInput.class)`

MOCK IS NOT THE SILVER-BULLET

测试是需要维护的

绝大部分问题都是人的问题

课后练习

INPUT

- when sbt "run 3 5"
 - it should not throw RuntimeException
 - it should print `invalid input` in stderr (using System Rules)
 - it should return 1 `System.exit(1)` (using System Rules)
- 其他情况

参考资料

- [Test Driven Development: By Example](#)
- [UnitTest](#)
- [Introduction to Test Driven Development](#)
- [Introducing BDD](#)
- [Stop mocking, start newing](#)
- [Java refactoring features of IntelliJ IDEA](#)