

DPF EXCERCISE #10

2

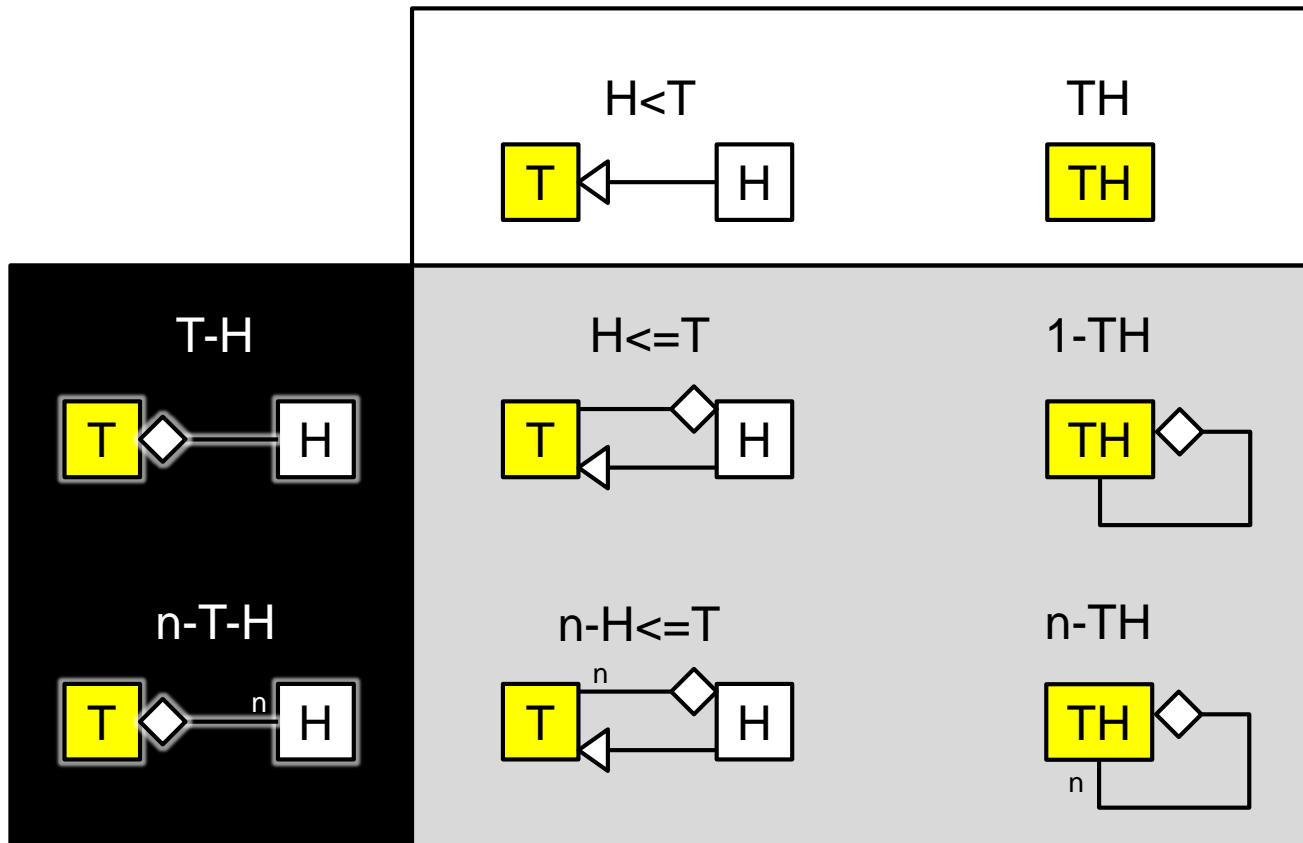
Hook Fundamentals

1a) Framework Hook Patterns

1b) Black- & White-Box Frameworks

Task #1 – Framework Hook Patterns

3



Task #2 – Log4J

4

- 2a) TH-Pattern of **Appender** and **Layout**
- 2b) TH-Pattern of **Logger** and **Appender**
- 2c) Is Log4J a white- or black-box framework?

Task #2 – Log4J

5

- Appenders reference layouts
- Loggers reference appenders
- Usually fully configured via files

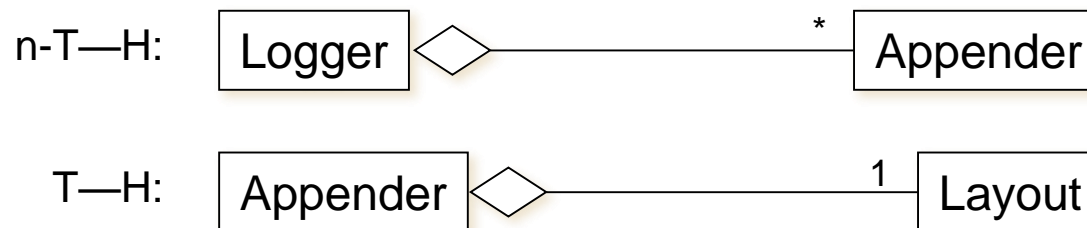
```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="warn" name="MyApp" packages="">
  <Appenders>
    <Console name="STDOUT" target="SYSTEM_OUT">
      <PatternLayout pattern="%m%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Root level="error">
      <AppenderRef ref="STDOUT"/>
    </Root>
  </Loggers>
</Configuration>
```

[<http://logging.apache.org/log4j/2.x/manual/appenders.html>]

Task #2a – Log4J

6

Log4J:



Log4J is a typical black-box framework.

Task #3 – JUnit

7

- 2a) TH-Pattern of **TestCase**
- 2b) TH-Pattern of **TestSuite** and **Test**
- 2b) TH-Pattern of **TestCase** and **TestResult**
- 2c) Is JUnit a white- or black-box framework?

Task #3 – JUnit

8

□ JUnit 3.8.1 (pretty old)

```
class MyTest extends TestCase {  
    ...  
    setup() { ... }  
    tearDown() { ... }  
    testABC() { ... }  
    ...  
}
```

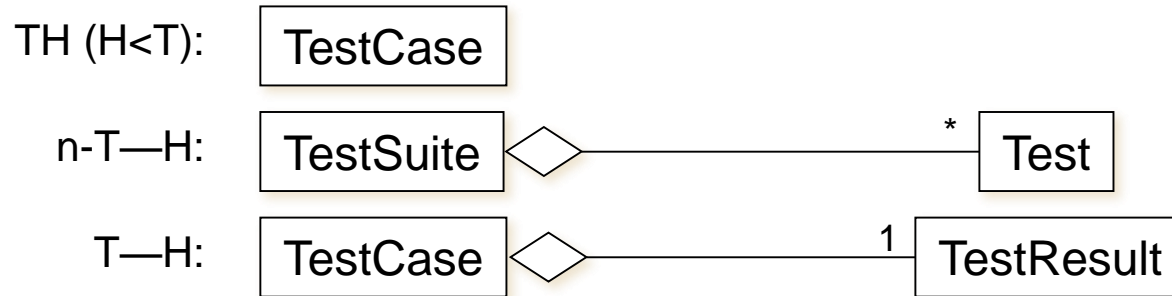
□ JUnit 4

```
class MyTest {  
    ...  
    @Before setup() { ... }  
    @After tearDown() { ... }  
    @Test abc() { ... }  
    ...  
}
```


Task #3 – JUnit

9

JUnit:



- Central class **TestCase** indicates white-box.
- Remaining classes indicate black-box.

10

End