

Ch2 Code Unit Testing

Write Code to Test Code(2)



Instructor: **Haiying SUN**

E-mail: hysun@sei.ecnu.edu.cn

Office: **ECNU Science Build B1104**

Available Time: **Wednesday 8:00 -12:00 a.m.**

Agenda



- **Code Test Techniques**
 - Logical Testing & Tools
 - Heuristic Rules
 - Junit & Qualified test scripts
- **Code Test Generation**
 - Control flow based
 - Data flow based
 - Mutation Based
 - Test Automation Tool Development

Example

阿里巴巴 Java 开发手册

9. 【推荐】编写单元测试代码遵守 BCDE 原则，以保证被测试模块的交付质量。

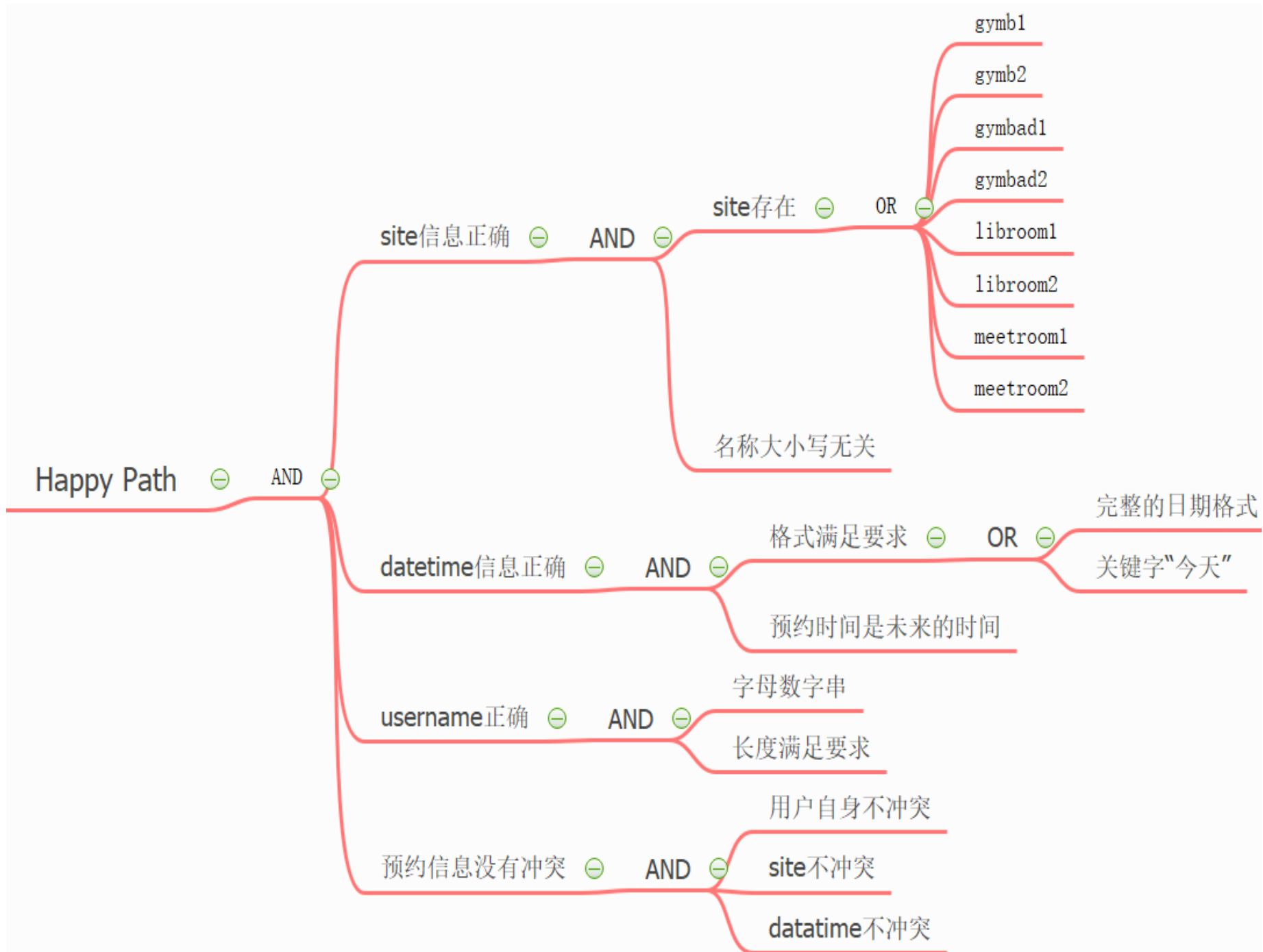
- **B**: Border, 边界值测试, 包括循环边界、特殊取值、特殊时间点、数据顺序等。
- **C**: Correct, 正确的输入, 并得到预期的结果。
- **D**: Design, 与设计文档相结合, 来编写单元测试。
- **E**: Error, 强制错误信息输入 (如: 非法数据、异常流程、非业务允许输入等), 并得到预期的结果。

Testing Strategy

- Right—BICEP
 - Right: Are the results right?
 - B: Are all the boundary conditions correct?
 - I: Can you check inverse relationships?
 - C: Can you cross-check results using other means?
 - E: Can you force error conditions to happen?
 - P: Are performance characteristics within bounds?

Right-BICEP

- **Right**—BICEP
 - happy path tests: tests should **first and foremost** validate that the code produces expected results what the users want.
 - If the code ran correctly, how would I know?
- **Exercise**
 - MeetCalendar.addReservation的Happy Path?



Right-BICEP

- Right—BICEP: Boundary conditions
 - Bogus or inconsistent input values, a filename like:
"!*W:X\&Gi/w\$→>\$g/h#WQ@.
 - Badly formatted data, bad phone number,
hysun@ecnu.edu.cn
 - Computations that can result in numeric overflow.
 - Empty or missing values: 0, 0.0, "", null.

Right-BICEP

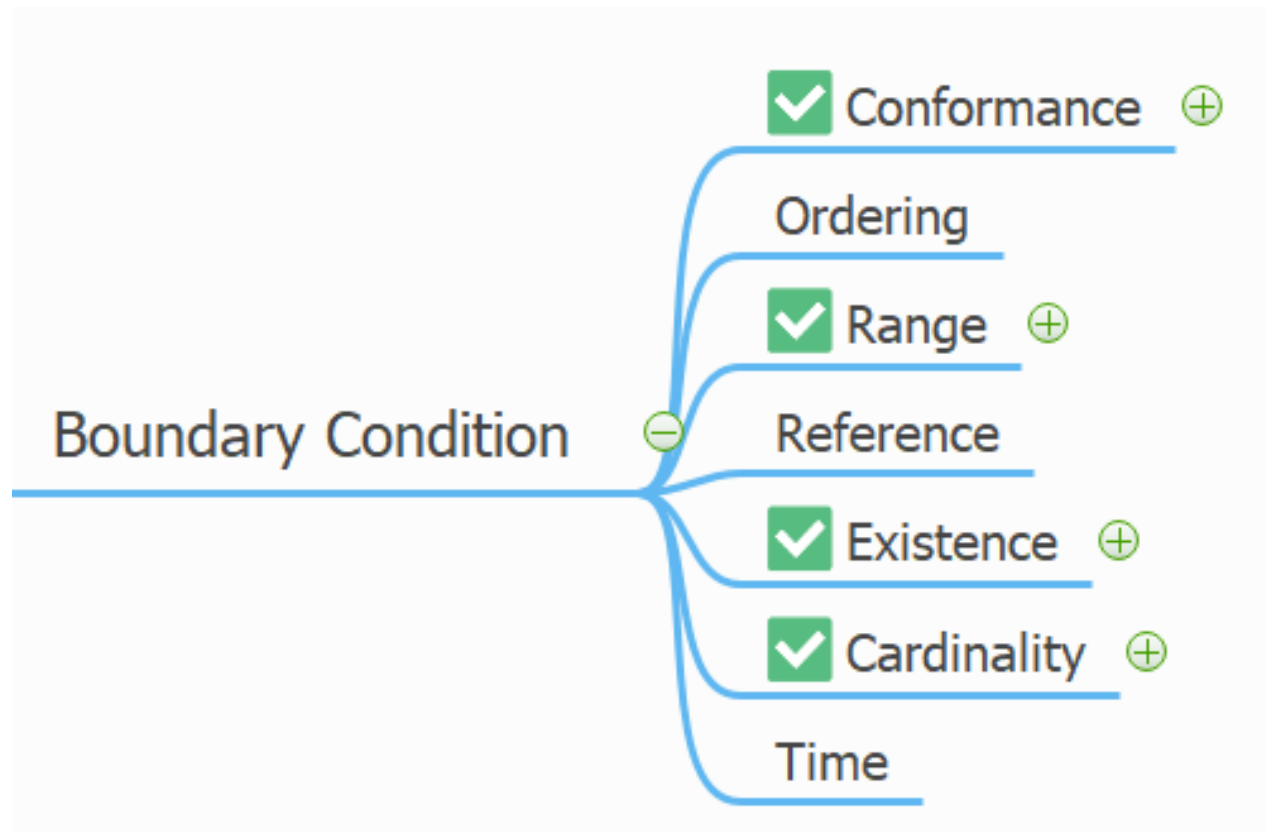
- Values far in excess of reasonable expectations
- Duplicates in lists that shouldn't have duplicates
- Ordered lists that aren't, and vice versa. Try handing a presorted list to a sort algorithm.
- Things that happen out of expected chronological order.

Corner Case Testing

- 每个被测对象使用CORRECT启发式规则：
 1. 是否涉及这些条件
 2. 当条件被违背时会发生什么
- Conformance: 数据格式是否与期望的一致
- Ordering: 数据之间的顺序是否满足要求
- Range: 数据是否在合理的最大值和最小值之间
- Reference: 被测代码是否使用了无法控制的外部引用
- Existence: 数据是否被要求存在, 例如非空, 非0, 必须在集合中
- Cardinality: 数据数量是否满足要求
- Time: 每件事情是否按顺发生? 是否在正确的时间发生? 发生是否及时?

Corner Case Testing

- **Exercise:** addReservation 涉及哪些规则?



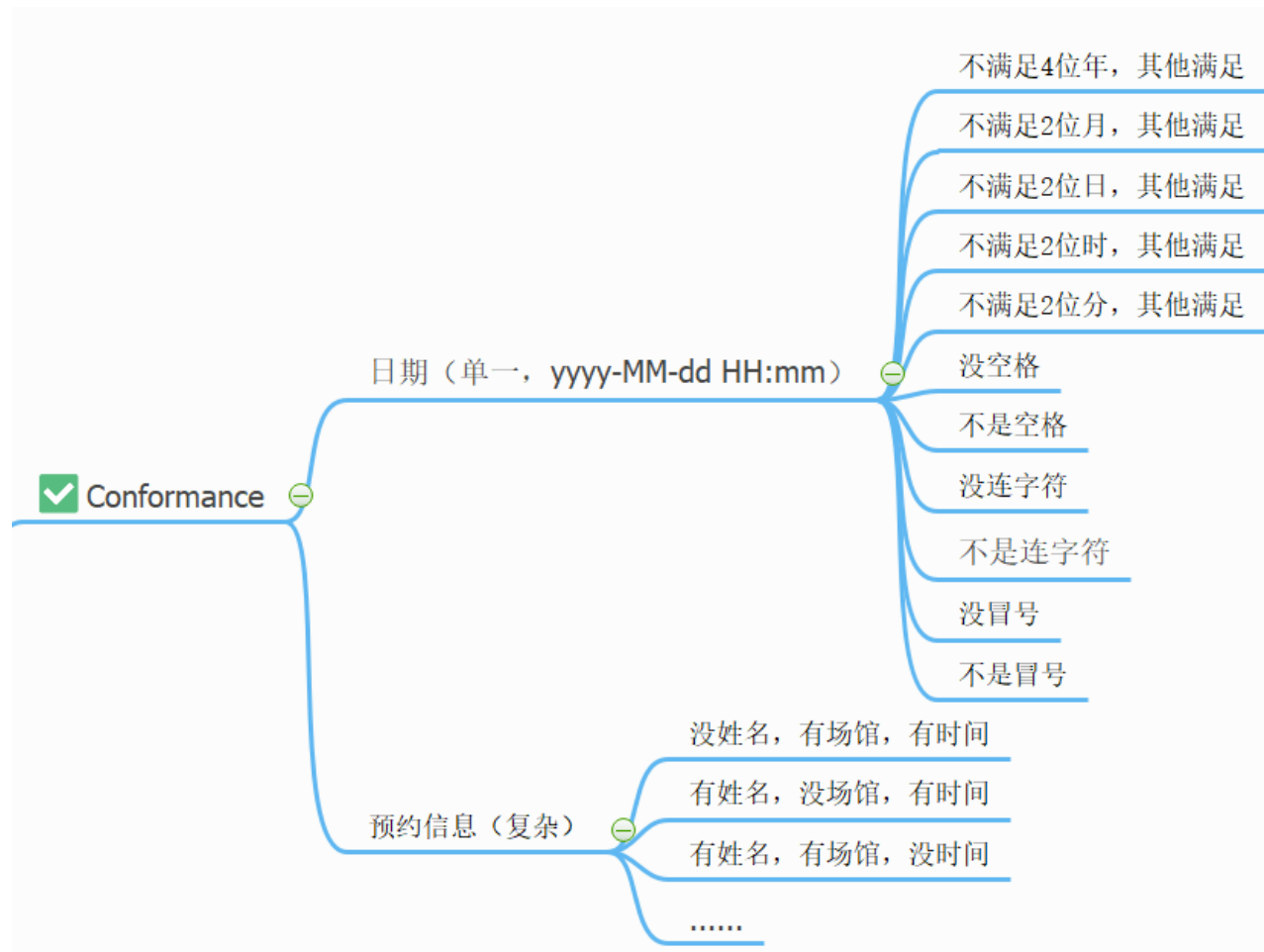
What else can go wrong

- **CORRECT: Conformance**

- 单一结构的数据
 - E-mail, phone number.....
- 复合结构的数据
 - （姓名，场地，时间）：只没有姓名/场地/时间，只有姓名/场地/时间.....
- 确定数据什么时候进入系统，有利于测试设计
 - UI层就确定了，那么测试就会简单

Corner Case Testing

- **Exercise:** addReservation的Conformance



What else can go wrong

- CORRECT: Ordering
 - the position of one piece of data within a larger collection
- CORRECT: Range
 - Java primitive types (primitive obsession: age, salary, score)
 - customized range
 - Invariant assertion

Corner Case Testing

- **Exercise:** addReservation的范围?



What else can go wrong

- CORRECT: Reference

- May consider:

1. What method under test (MUT) references outside its scope
2. What external dependencies MUT has
3. Whether MUT depends on the object being in a certain state
4. Any other conditions that must exist

- precondition
 - postcondition
 - side effect

What else can go wrong

- CORRECT: Existence: Dose some given thing exist?
 - what will happen if the value is null, zero, or otherwise empty, especially check value returned
 - a special case of cardinality
- CORRECT: Cardinality (护栏柱测试)
 - the count of some set of values is only interesting in these three cases (0-1-n rule)
 - Zero
 - One
 - Many (more than one)

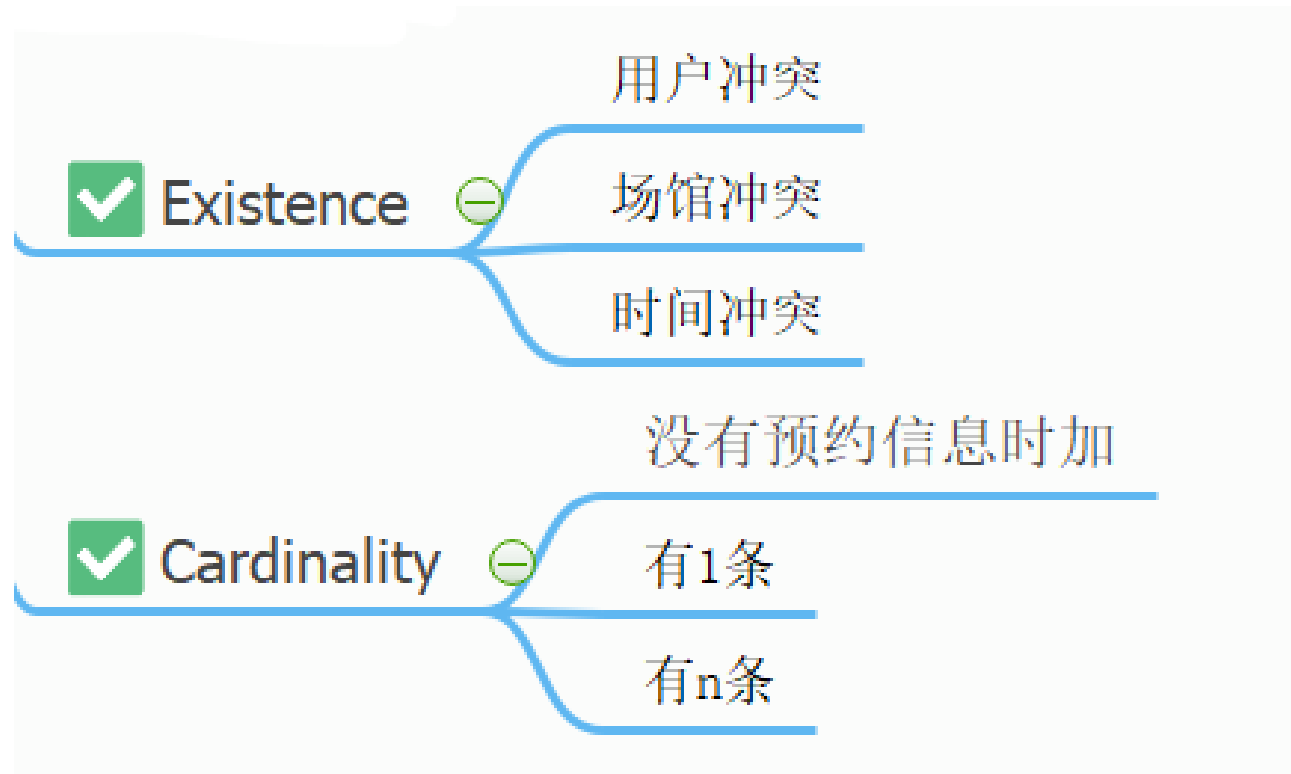
Defect Patterns Ten Top List

Defect Type	2008 Frequency	2009 Frequency	% Difference	Ranking Change
NULL Pointer Deference	27.95%	27.81%	0.14% ↓	0
Resource Leak	25.73%	23.34%	2.39% ↓	0
Unintentional Ignored Expressions	9.76%	9.71%	0.05% ↓	0
Use Before Test (NULL)	8.09%	8.35%	0.25% ↑	-1
Use After Free	6.46%	5.91%	0.34% ↓	-1
Buffer Overflow (statically allocated)	6.14%	5.79%	0.55% ↓	-1
Unsafe Use of Returned NULL	5.85%	5.30%	0.55% ↓	-1
Uninitialized Values Read	5.50%	8.41%	2.91% ↑	+4
Unsafe Use of Returned Negative	3.72%	3.90%	0.18% ↑	0
Type and Allocation Size Mismatch	0.62%	1.10%	0.48% ↑	0
Buffer Overflow (dynamically allocated)	0.31%	0.21%	0.10% ↓	0
Use Before Test (negative)	0.21%	0.18%	0.03% ↓	0

2009 COVERITY SCAN [OPEN SOURCE](#) REPORT

Corner Case Testing

- **Exercise:** addReservation的Existence?



What else can go wrong

- CORRECT: Time

- Relative time (ordering in time) : if methods are called out of order
- Absolute time (elapsed and wall clock): test any time-sensitive code on boundary days
- Concurrency issues
 1. what will happen if multiple threads access the same object at the same time?
 2. Do you need to synchronize any global or instance-level data or methods?
 3. How about external access to files or hardware?

Exercise

- 请完成完整的addReservation测试代码



Right-BICEP

- Right—BICEP: **Checking Inverse Relationship**
 - Seek an independent means of verification.
 - 用“逆行为”测试被测试代码
 - 在数据库中插入一条记录后，查询该记录
 - 已经使用的款项总数 = 款项总数 - 剩余的款项数
- Right—BICEP: **Cross checking**
 - 使用不同数据之间的关系进行测试
 - 已经使用的款项总数 = 款项总数 - 剩余的款项数

Exercise

- addReservation中是否存在inverse relationship或者cross-check?
 - 每个场地每天的预定数是有上限的

Right-BICEP

- Right—BICEP: Forcing Error Condition
 - think about what kinds of errors or other environmental constraints :
 1. destroy business rules
 2. Running out of memory
 3. Running out of disk space
 4. Network availability and errors
 5. System load
 6. Very high or very low video resolution

.....
 - 使用Mock对象模拟各种异常

Right-BICEP

- Right—BICEP: Performance characteristics
 - Usually applied to E2E testing
 - Take as baseline information for changing
 - Junit 5 @Timeout

```
@Test
@Timeout(value = 100, unit = TimeUnit.MILLISECONDS)
void failsIfExecutionTimeExceeds100Milliseconds() {
    // fails if execution time exceeds 100 milliseconds
}
```


Summary

- Unit is defined by function, size and time
- Unit testing is to write code to test code
- Logical code coverage criteria are intended to detect logical bugs
- Different coverage criterion has different defect-detective ability
- Statement coverage is the weakest while Multiple coverage is the strongest but need more test cases
- Coverage tools are the practical implementation of logical coverage criteria theory. Because of different coverage data collection strategies, one should check the tools coverage definitions before using them.

The End