

# **单元测试计划书**

**( Unit Test Plan )**

**1450126 陈东仪**

**1452717 徐雯**

**1452719 项安颖**

**1452721 张天人**

# 目录

1 引言	4
1.1 概述	4
1.2 目的	4
1.3 测试策略	4
1.4 范围	4
1.5 参考文献	5
1.6 术语	5
2 测试项目	6
2.1 IO 模块	6
2.1.1 被测函数	6
2.1.2 不被测函数	6
2.2 Reflect 模块	7
2.2.1 被测函数	7
2.2.2 不被测函数	7
2.3 Collect 模块	7
2.3.1 被测函数	7
2.3.2 不被测函数	8
2.4 Base 模块	9
2.4.1 被测函数	9
2.4.2 不被测函数	10
2.5 Math 模块	11

2.5.1 被测函数.....	11
2.5.2 不被测函数.....	11
2.6 Net 模块.....	12
2.6.1 被测函数.....	12
2.6.2 不被测函数.....	12
3 测试方法.....	14
4 测试通过/失败标准.....	15
5 测试挂起/恢复的条件.....	16
6 单元测试交付物.....	17
7 单元测试任务.....	18
8 环境需求.....	20
9 角色和职责.....	21
10 人员及培训.....	22
11 单元测试进度.....	23
12 风险和应急计划.....	25

# 1 引言

## 1.1 概述

Guava 工程包含了若干被 Google 的 Java 项目广泛依赖 的核心库，例如：集合 [collections]、缓存 [caching]、原生类型支持 [primitives support]、并发库 [concurrency libraries]、通用注解 [common annotations]、字符串处理 [string processing]、I/O 等等。所有这些工具每天都在被 Google 的工程师应用在产品服务中。

## 1.2 目的

本文档为 guava 系统以下模块的单元测试活动提供范围、方法、资源和进度方面的指导：base, collect, io, math, net, reflect。本文档的读者主要是开发经理，测试人员和开发人员。

## 1.3 测试策略

以类为单元，采用独立的单元测试策略，通过设计相应的驱动和桩的方法来测试类中的方法。在选择类中被测方法时，根据方法的规模和复杂度进行判定。非空非注释代码行数  $LOC > 20$ ，或者复杂度  $VG > 3$  的方法进行单元测试，其他方法不进行单元测试。

对于子类的测试采用分层增量测试策略，对子类的变化部分设计新的测试用例，与父类相同的部分则重用父类的测试用例。

执行单元测试的次序从最小依赖关系的类开始测试，再逐步扩大到依赖关系较强的类。使用多种测试方法补充用例，减少用例的冗余以及完善用例。

## 1.4 范围

单元测试包含了计划阶段、设计阶段、实现阶段和执行阶段四个阶段。本单元测试计划是整个软件开发项目中的一部分，起始于详细设计阶段，直到单元测试阶段结束后终止。该

计划主要处理与 guava 系统单元测试有关的任务安排、资源需求、人力需求、风险管理、进度安排等内容。

## **1.5 参考文献**

《软件需求规格说明 ( Software Requirement Specification ) 》

Guava 官方 API 文档

## **1.6 术语**

无。

## 2 测试项目

### 2.1 IO 模块

#### 2.1.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-IO-01	BaseEncoding	GV-IO-01-01	encodingStream()	43	7
		GV-IO-01-02	decodingStream()	50	9
GV-IO-02	ByteSource	GV-IO-02-01	size()	25	5
		GV-IO-02-02	contentEquals()	24	4
GV-IO-03	ByteStream	GV-IO-03-01	toByteArray()	30	4
		GV-IO-03-02	copy()	27	6
GV-IO-04	Closer	GV-IO-04-01	close()	22	5
GV-IO-05	LineBuffer	GV-IO-05-01	add()	26	15
GV-IO-06	MoreFiles	GV-IO-06-01	deleteRecursively()	33	9
		GV-IO-06-02	deleteDirectoryContents()	22	6

#### 2.1.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-IO-07	FileBackedOutputStream	GV-IO-07-01	reset()	18	7
		GV-IO-07-02	update()	18	3
GV-IO-08	Files	GV-IO-08-01	createTempDir()	19	3
		GV-IO-08-02	createParentDirs()	11	3
		GV-IO-08-03	move()	14	4
		GV-IO-08-04	readLines()	17	1
GV-IO-09	Flushables	GV-IO-09-01	flush()	10	3
GV-IO-10	LineBuffer	GV-IO-10-01	finishLine()	8	1
GV-IO-11	MoreFiles	GV-IO-11-01	equal()	15	3
		GV-IO-11-02	getFileExtension()	9	2

## 2.2 Reflect 模块

### 2.2.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-RE-01	TypeToken	GV-RE-01-01	isSubtypeOf()	25	8
		GV-RE-01-02	method()	28	1
		GV-RE-01-03	constructor()	28	1
		GV-RE-01-04	toGenericType()	25	3
GV-RE-02	TypeVisitor	GV-RE-02-01	visit()	28	14

### 2.2.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-RE-03	Reflection	GV-RE-03-01	initialize()	8	3
GV-RE-04	TypeResolver	GV-RE-04-01	resolveParameterizedType()	10	1
		GV-RE-04-02	resolveType()	14	5

## 2.3 Collect 模块

### 2.3.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-CO-01	GeneralRange	GV-CO-01-01	intersect()	46	17
GV-CO-02	Iterators	GV-CO-02-01	getOnlyElement()	17	4
	Iterators	GV-CO-02-02	contains()	16	6
GV-CO-03	Maps	GV-CO-03-01	subMap()	24	8
GV-CO-04	Multimaps	GV-CO-04-01	filterKeys()	17	5
GV-CO-05	Multisets	GV-CO-05-01	removeOccurrences()	20	4

	Multisets	GV-CO-05-02	retainOccurrencesImpl()	20	4
GV-CO-06	Queues	GV-CO-06-01	drain()	30	4
	Queues	GV-CO-06-02	drainUninterruptibly()	39	11
GV-CO-07	Range	GV-CO-07-01	containsAll()	21	5
GV-CO-08	Sets	GV-CO-08-01	subset()	23	8
	Sets	GV-CO-08-02	symmetricDifference()	46	4
GV-CO-09	Streams	GV-CO-09-01	concat()	22	2
	Streams	GV-CO-09-02	zip()	28	2
GV-CO-10	Tables	GV-CO-10-01	toTable()	27	2

### 2.3.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-CO-02	Iterators	GV-CO-02-03	pollNext()	9	2
	Iterators	GV-CO-02-04	checkNonnegative()	5	2
	Iterators	GV-CO-02-05	advance()	10	2
GV-CO-11	Lists	GV-CO-11-01	reverse()	11	4
	Lists	GV-CO-11-02	addAllImp()	9	2
GV-CO-03	Maps	GV-CO-03-02	unmodifiableNavigableMap()	11	2
	Maps	GV-CO-03-03	containsEntryImpl()	6	2
GV-CO-05	Multisets	GV-CO-05-03	Filter()	10	2



	Multisets	GV-CO-05-04	setCountImpl()	11	3
GV-CO-06	Queues	GV-CO-06-03	newPriorityQueue()	9	2
	Queues	GV-CO-06-04	newLinkedBlockingDeque()	8	2
GV-CO-07	Range	GV-CO-07-02	canonical()	6	1
	Range	GV-CO-07-03	readResolve()	7	2
GV-CO-08	Sets	GV-CO-08-03	newHashSet()	5	1
	Sets	GV-CO-08-04	makeComplementByHand()	6	1

## 2.4 Base 模块

### 2.4.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-BA-01	Ascii	GV-BA-01-01	precomputedInternal()	23	2
		GV-BA-01-02	truncate()	21	3
		GV-BA-01-03	equalsIgnoreCase()	21	6
GV-BA-02	Utf8	GV-BA-02-01	encodedLength()	32	3
		GV-BA-02-02	isWellFormedSlowPath()	40	2
GV-BA-03	Strings	GV-BA-03-01	repeat()	21	3
GV-BA-04	Preconditions	BA-11	format()	25	4
GV-BA-05	Splitter	GV-BA-05-01	computeNext()	43	8
GV-BA-06	Joiner	GV-BA-06-01	skipNulls()	33	1

GV-BA-07	Converter	GV-BA-07-01	convertAll()	23	1
		GV-BA-07-02	replaceFrom()	28	4
GV-BA-08	SmallCharMatcher	GV-BA-08-01	matches()	23	2
GV-BA-09	CharMatcher	GV-BA-09-01	precomputedInternal()	23	2
		GV-BA-09-02	precomputedPositive()	16	4
		GV-BA-09-03	removeFrom()	24	2

## 2.4.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-BA-09	CharMatcher	GV-BA-09-04	trimFrom()	16	2
		GV-BA-09-05	trimLeadingFrom()	9	2
		GV-BA-09-06	trimTrailingFrom()	9	2
		GV-BA-09-07	apply()	3	2
GV-BA-10	Enums	GV-BA-10-01	doForward()	3	1
		GV-BA-10-02	doBackward()	3	1
		GV-BA-10-03	hashCode()	3	1
		GV-BA-10-04	toString()	4	1
GV-BA-01	Ascii	GV-BA-01-04	toUpperCase()	15	1
		GV-BA-01-05	getAlphaIndex()	3	1
		GV-BA-01-06	isUpperCase()	3	1

GV-BA-11	JdkPattern	GV-BA-11-01	find()	3	1
		GV-BA-11-02	replaceAll()	3	1
		GV-BA-11-03	end()	3	1
		GV-BA-11-04	start()	3	1

## 2.5 Math 模块

### 2.5.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-MA-01	BigIntegerMA	GV-MA-01-01	log2()	29	4
GV-MA-01	BigIntegerMA	GV-MA-01-02	sqrt()	25	3
GV-MA-01	BigIntegerMA	GV-MA-01-03	log10()	20	4
GV-MA-02	IntMA	GV-MA-02-01	pow()	26	7
GV-MA-02	IntMA	GV-MA-02-02	gcd()	20	3
GV-MA-02	IntMA	GV-MA-02-03	binomial()	22	4
GV-MA-02	IntMA	GV-MA-02-04	checkedMultiply()	18	3
GV-MA-02	IntMA	GV-MA-02-05	checkedPow()	32	6
GV-MA-02	IntMA	GV-MA-02-06	factorial()	31	2
GV-MA-02	IntMA	GV-MA-02-07	saturatedPow()	34	6
GV-MA-03	LongMA	GV-MA-03-01	isPrime()	20	4

### 2.5.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-MA-01	BigIntegerMA	GV-MA-01-04	ceilingPowerOfTwo ()	2	1
GV-MA-01	BigIntegerMA	GV-MA-01-05	divide()	5	1
GV-MA-01	BigIntegerMA	GV-MA-01-06	isPowerOfTwo ()	4	1
GV-MA-04	DoubleMA	GV-MA-04-01	fuzzyCompare()	10	2
GV-MA-04	DoubleMA	GV-MA-04-02	mean()	11	2
GV-MA-03	LongMA	GV-MA-03-02	checkedAdd()	5	1
GV-MA-03	LongMA	GV-MA-03-03	checkedSubtract()	5	1
GV-MA-03	LongMA	GV-MA-03-04	saturatedMultiply()	36	7
GV-MA-04	Quantiles	GV-MA-04-01	Partition()	10	2
GV-MA-04	Quantiles	GV-MA-04-02	movePivotToStartOfSlice()	9	2

## 2.6 Net 模块

### 2.6.1 被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-NE-01	HostAndPort	GV-NE-01-01	fromString()	26	4
GV-NE-01	HostAndPort	GV-NE-01-02	toString()	15	3
GV-NE-02	InetAddresses	GV-NE-02-01	toAddrString()	19	3
GV-NE-02	InetAddresses	GV-NE-02-02	convertDottedQuadToHex ()	16	3

### 2.6.2 不被测函数

类标识符	类名	方法标识符	方法名称	LOC	VG
GV-NE-01	HostAndPort	GV-NE-01-03	getPort ()	4	1
GV-NE-01	HostAndPort	GV-NE-01-04	requireBracketsForIPv6 ()	4	1
GV-NE-02	INEAddresses	GV-NE-02-03	fromLittleEndianByteArray ()	7	2
GV-NE-02	INEAddresses	GV-NE-02-04	decrement ()	10	2
GV-NE-02	INEAddresses	GV-NE-02-05	increment ()	10	2

### 3 测试方法

根据类规约和操作规约构建测试用例，利用传统等价类划分法、边界值分析法、判定表法等黑盒测试技术对边界值、正常值、错误值等情况进行全面测试，以覆盖所有前置条件和后置条件组合。

对具有特殊需求的类辅以下两种方法设计测试用例：

（1）根据状态转换图构建测试用例。该方法根据被测试的类的对象所处的状态以及状态之间的转移来构造测试用例，对状态之间和状态内部的每一转换及其可能发生的异常转换、转换的监护条件等进行全面测试。

（2）基于实现构建测试用例。该方法利用传统逻辑覆盖法、数据流分析法等白盒测试技术对程序的逻辑结构或数据流进行测试，以达到一定的代码覆盖率。

更详细的测试策略描述参考《单元测试说明》。

## 4 测试通过/失败标准

测试通过的标准表述如下：

- 所有单元测试的用例都被执行并通过；
- 所有发现的缺陷都被修正并回归测试过；
- 所有被测对象的前置条件和后置条件组合覆盖率达到 100% ,或能明确给出不需要达到的理由；
- 单元测试报告被权签人批准。

测试失败标准表述如下：

- 严重缺陷密度大于 15 个/kLOC；
- 发现软件结构有重大设计问题，其修改会导致 20%以上的接口、功能、数量的变化，进一步测试相关特性已经无意义；
- 发现关键功能未被设计，该功能的设计会导致 20%以上的接口、功能、数量的变化，进一步测试相关特性已经无意义；

测试结果审批过程：

开发人员提交单元测试报告→开发经理签字并提交 SQA→SQA 对报告进行评审并签字

( 测试经理参与 ) →产品经理签字。

## 5 测试挂起/恢复的条件

测试挂起的条件有：

- 当某个类在单元测试执行过程中发现有阻塞用例的时候 ,该类的单元测试被挂起。
- 当有 20%以上的被测类都遇到有阻塞用例的时候 ,所有类的单元测试被挂起。
- 当出现有新增需求的时候 ,与该需求相关的所有类的单元测试被挂起。
- 当开发人员提出要进行设计变更的时候 ,相关类的单元测试将被挂起。

测试恢复的条件有：

- 测试被挂起的条件已经被解决。
- 需要恢复测试的对象达到单元测试入口条件 ,在这里要求这些被测对象已经通过代码走读（要提交走读报告）和语法检查（要提交检查结果）。



## 6 单元测试交付物

- 单元测试计划 ( Unit Test Plan ) ；
- 单元测试设计规格 ( Unit Test Design Specification ) ；
- 单元测试用例规格 ( Unit Test Case Specification ) ；
- 单元测试用例脚本 ；
- 单元测试驱动和桩代码 ；
- 单元测试执行日志 ( Unit Test Log ) ；
- 单元测试报告 ( Unit Test Report ) 。

## 7 单元测试任务

单元测试任务表

任务标识	任务描述	责任人	优先级	依赖关系
UT-TASK-001	单元测试计划写作	开发经理	高	
UT-TASK-003	单元测试计划评审	SQA	中	UT-TASK-001
UT-TASK-005	单元测试计划修改	开发经理	中	UT-TASK-003
UT-TASK-007	单元测试设计规格写作	某开发人员	中	UT-TASK-003
UT-TASK-009	单元测试设计规格评审	SQA	中	UT-TASK-007
UT-TASK-011	单元测试设计规格修改	某开发人员	中	UT-TASK-009
UT-TASK-013	单元测试用例规格写作	某开发人员	高	UT-TASK-009
UT-TASK-015	单元测试用例规格评审	SQA	中	UT-TASK-013
UT-TASK-017	单元测试用例规格修改	某开发人员	中	UT-TASK-015
UT-TASK-019	单元测试驱动、桩、用例脚本代码实现	某开发人员	中	UT-TASK-015
UT-TASK-021	驱动、桩、脚本代码走读	SQA	低	UT-TASK-019

UT-TASK-023	驱动、桩、脚本代码修改	某开发人员	低	UT-TASK-021
UT-TASK-025	单元测试执行及回归	某开发人员	高	UT-TASK-023
UT-TASK-027	单元测试报告	开发经理	高	UT-TASK-025
UT-TASK-029	单元测试报告审批	产品经理	高	UT-TASK-027

## 8 环境需求

### 8.1 硬件需求

硬盘大于 20G ;

内存大于 1G。

### 8.2 软件需求

Windows 7 以上 ;

JDK 8 ;

IntelliJ IDEA ;

Guava-V21.0。

### 8.3 测试工具

JUnit 4.0。

### 8.4 其他

无

## 9 角色和职责

角色	职责	负责人
产品经理	解决资源（包括人、工具等）需求，对单元测试结果进行监督	项安颖
开发经理	制定单元测试计划，安排单元测试任务	陈东仪
测试经理	参与单元测试结果验收	徐雯
SQA	对单元测试过程（包括代码走读、正规检视活动）进行监控	张天人
开发人员	完成单元测试需要的输入，并完成单元测试设计规格、单元测试用例规格、单元测试规程的制定，执行单元测试，记录发现问题，修改问题，并负责问题的回归测试。与此同时，负责定位问题和解决问题	项安颖 陈东仪 徐雯 张天人

## 10 人员及培训

- 需要 2 名一年以上工作经验的开发人员，并且他们应在详细设计开始之后全职投入到单元测试项目组中；
- 在详细设计完成之前，需要完成对项目需求、系统设计、详细设计、单元测试技术、单元测试脚本技术方面的培训；
- 在编码完成之前要完成缺陷电子流使用、测试日志表格使用、测试工具使用的培训；
- 以上培训大约需要花费每人 20 人时的工作量。

## 11 单元测试进度

任务标识	任务描述	起始日期	周期/天
UT-TASK-001	单元测试计划写作	2017/4/27	3
UT-TASK-003	单元测试计划评审	单元测试计划完成后 2 天内	2
UT-TASK-005	单元测试计划修改	单元测试计划评审完成后 2 天内	4
UT-TASK-007	单元测试设计规格写作	单元测试计划评审完成后 4 天内	4
UT-TASK-009	单元测试设计规格评审	单元测试设计规格完成后 3 天内	3
UT-TASK-011	单元测试设计规格修改	单元测试设计规格评审完成后 2 天内	2
UT-TASK-013	单元测试用例规格写作	单元测试设计规格评审完成后 5 天内	5
UT-TASK-015	单元测试用例规格评审	单元测试用例规格完成后 2 天内	2
UT-TASK-017	单元测试用例规格修改	单元测试用例规格评审完成后 1 天内	1
UT-TASK-019	单元测试驱动、桩、用例脚本代码实现	单元测试用例规格评审完成后 7 天内，并且编码阶段已经开始	7
UT-TASK-021	单元测试驱动、桩、脚本代码走读	单元测试驱动、桩、用例脚本代码完成后 3 天	3
UT-TASK-023	单元测试驱动、桩、脚本代码修改	单元测试驱动、桩、脚本代码走读后 2 天	2
UT-TASK-025	单元测试执行及回归	单元测试驱动、桩、脚本代码走读后 5 天，并且编码阶段已经结束	5

UT-TASK-027	单元测试报告	单元测试执行及回归完成后 1 天内	
UT-TASK-029	单元测试报告审批	单元测试报告完成后 1 天内	
	风险预留时间	单元测试阶段工作中任意时候	
	单元测试阶段里程碑时 间点	2009-11-15	



## 12 风险和应急计划

风险 ID	风险描述	责任人	优先级	规避措施	应急计划
1	人员无法及时到位	开发经理	高	1. 在产品的预算中体现这部分需求 2. 定期催促人力资源部进行资源协调 3. 从可能空闲的产品部中物色人员	1. 推迟进度计划 2. 进行招聘 3. 考虑工作外包
2	人员技能不符合要求	开发经理	中	1. 在人力预算中给出人员技能要求 2. 对提供的人员进行技能面试 3. 从其他产品部门协调有能力的人员	1. 提高培训的强度 2. 加强培训效果监控 3. 对工作输出加强检视
3	测试工具无法到位	某员工	低	1. 在产品预算中尽早体现这部分需求 2. 联系其他产品，看是否有空闲的 License 3. 尽早联系工具代理商，洽谈采购事宜	1. 采用人工分析程序规模 2. 允许使用规模估计值 3. 自行开发测试工具