

本课程理论考试大题（共70分）
与实验/作业高度对应，不亲自做
一遍实验可能无法答题。
请谨慎对待实验！

总体要求：

1. 独立完成
2. 报告清晰、规范
3. 覆盖实验要求（特别注意参考评价要素）
4. 体现个性化（各人尽量针对不同实验对象来完成实验）
5. 体现分析与思考（对实验中出现的問題、数据等加以评述，对存在的不足与改进方向多做思考。简明扼要，不写废话）

关于编程语言

- 本实验会轻度使用Java和Python两门语言
- 请自学，如有问题，可咨询助教
- 不愿意用这两门语言，也可用C++等任意语言，只要实现同等实验内容接口

实验课评分构成

- 实验一：单元与集成测试20%
- 实验二：测试管理20%
- 实验三：黑盒测试20%
- 实验四：覆盖测试10%
- 实验五：自动化测试10%
- 实验六：性能测试20%

实验一：单元与集成测试

- 请使用JUnit工具编写单元测试，对Apache Commons库中的某一工具类进行测试。实验要求如下：
 - 查阅Apache Common库中的类，选择一个感兴趣的类作为待测对象，查阅该类的接口说明，了解类的功能。下载待测类的代码，将其复制到一个新建的项目工程中，放置在与Apache Common库中同名的包下，使得该代码可以覆盖Apache Common库Jar包中的原有代码。
 - 查阅文献，了解软件缺陷的常见类型；在待测类代码中注入一个不影响编译的缺陷，作为待检测的问题。
 - 实现不依赖于工具的单元测试。编写main方法驱动待测类中至少3个关键方法（覆盖注入错误的方法），实现对模块单元质量的检查。
 - 使用JUnit构造测试单元测试用例，并运行测试，获得测试结果；将多个JUnit将单元测试打包为一个测试集，实现单元测试的批量运行；
 - 对比第3步和第4步的方法，分析其各自有哪些优缺点，讨论为什么一般建议基于JUnit等工具开展单元测试，而不是采用步骤3中的方法；

参考资料：https://en.wikipedia.org/wiki/Software_bug#Common_types_of_computer_bugs

目标知识与能力

知识	能力
(1)· 软件缺陷及其常见类别； (2)· 单元测试的设计方法； (3)· 测试用例与测试集； 	(1)· 问题分析：通过文献研究，了解有哪些缺陷类型。 (2)· 设计/开发解决方案：开发单元测试代码； (3)· 使用现代工具：使用单元测试框架并分析其局限性； (4)· 终身学习：自主学习开源软件、测试工具，阅读他人高质量代码。

评价要素

要素	实验要求
缺陷注入	了解常见缺陷类型，注入符合常规的缺陷
测试代码编写	测试用例包含输入数据和预期结果校验，能够识别失效
方法比较	能够体会不同方法的差别，了解为什么要使用单元测试框架

尽量选择不同的类来测试，同一类在不同同学的实验中出现越多，相应评分越低

表 6-1 Apache Commons 程序库

组件	官方描述	说明
BCEL	Byte Code Engineering Library - analyze, create, and manipulate Java class files.	Java 字节码处理库。
<u>BeanUtils</u>	Easy-to-use wrappers around the Java reflection and introspection APIs.	对 Java 反射等的封装。
BSF	Bean Scripting Framework - interface to scripting languages, including JSR-223.	脚本运行框架，可用于运行 JSP。
Chain	Chain of Responsibility pattern <u>implementation</u> .	一个“责任链”设计模式的实现框架。
CLI	Command Line arguments parser.	命令行解析器。
Codec	General encoding/decoding algorithms (for example phonetic, base64, URL).	常用编解码算法。
Collections	Extends or augments the Java Collections Framework.	Java Collections 容器库的扩展。
Compress	Defines an API for working with tar, zip and bzip2 files.	tar、zip 和 bzip2 压缩包的处理 API。
Configuration	Reading of configuration/preferences files in various formats.	从多种文件格式读取配置的库。
Crypto	A cryptographic library optimized with AES-NI wrapping <u>Openssl</u> or JCE algorithm implementations.	加密算法库。
CSV	Component for reading and writing comma separated value files.	CSV 文件读写库。
Daemon	Alternative invocation mechanism for <u>unix</u> -daemon-like java code.	后台服务程序调用机制。
DBCP	Database connection pooling services.	数据库连接池服务。
<u>DbUtils</u>	JDBC helper library.	JDBC 数据库辅助工具。
Digester	XML-to-Java-object mapping utility.	XML 到 Java 对象的映射工具。
Email	Library for sending e-mail from Java.	Email 发送库。
Exec	API for dealing with external process execution and environment management in Java.	进程调用相关库。
<u>FileUpload</u>	File upload capability for your servlets and web applications.	文件上传工具。

组件	官方描述	说明
<u>Functor</u>	A <u>functor</u> is a function that can be manipulated as an object, or an object representing a single, generic function.	“函子”程序语言机制编程框架
Geometry	Space and coordinates.	几何学工具集
Imaging	A pure-Java image library.	图像处理库
IO	Collection of I/O utilities.	IO 输入输出工具库
JCI	Java Compiler Interface	Java 编译器访问接口
JCS	Java Caching System	Java 缓存系统
Jelly	XML-based scripting and processing engine.	可执行 XML 脚本处理引擎
<u>Jexl</u>	Expression language which extends the Expression Language of the JSTL.	JEXL 表达式语言支撑实现
<u>JXPath</u>	Utilities for manipulating Java Beans using the XPath syntax.	以 XPath 语法处理 Java Beans 的库
Lang	Provides extra functionality for classes in <u>java.lang</u> .	<u>java.lang</u> 包的扩展
Logging	Wrapper around a variety of logging API implementations.	日志库
Math	Lightweight, self-contained mathematics and statistics components.	数学和统计库
Net	Collection of network utilities and protocol implementations.	网络和协议工具
Numbers	Number types (complex, quaternion, <u>fraction</u>) and utilities (arrays, combinatorics).	数值处理工具

OGNL ^o	An Object-Graph Navigation Language ^o	以表达式语言访问对象属性的库 ^o
Pool ^o	Generic object pooling component. ^o	对象池支撑库 ^o
Proxy ^o	Library for creating dynamic proxies. ^o	支持“代理”设计模式的库 ^o
RDF ^o	Common implementation of RDF 1.1 that could be implemented by systems on the JVM. ^o	RDF 实现 ^o
RNG ^o	Implementations of random numbers generators. ^o	随机数生成器 ^o
SCXML ^o	An implementation of the State Chart XML specification aimed at creating and maintaining a Java SCXML engine. ^o	State Chart XML 实现 ^o
Statistics ^o	Statistics. ^o	统计相关工具 ^o
Text ^o	Apache Commons Text is a library focused on algorithms working on strings. ^o	字符串算法库 ^o
Validator ^o	Framework to define validators and validation rules in an xml file. ^o	XML 文件验证工具 ^o
VFS ^o	Virtual File System component for treating files, FTP, SMB, ZIP and such like as a single logical file system. ^o	虚拟文件系统实现 ^o
Weaver ^o	Provides an easy way to enhance (weave) compiled bytecode. ^o	字节码编织相关工具 ^o

实验二： 测试管理

- 在实验一的基础上开展测试管理，要求：
 1. 搭建一个包含以下模块的开发环境
 - Git代码仓库（如Github、Gitee、本地Gitlab等）
 - Jenkins或其它持续集成与测试工具
 - issue tracking模块（例如Github、JIRA、MentisBT等）
 2. 采用Ant、Maven等配置工具配置软件项目，项目中必须包含build和test两个目标，build用来导出程序的完整jar包，test用来对程序进行测试。
 3. 在持续集成（如Jenkins）中应用项目配置中设定的test任务，完成冒烟测试。（Jenkins日志中可以看到执行测试用例的提示）
 4. 调研缺陷提交的一般性流程和格式要求，在缺陷管理工具中登记发现的bug，填写缺陷信息。
 5. 按项目将要给其它人接手的角度，为其编写一个简要的README.md来说明项目开发、测试与缺陷提交步骤、方法（应具体化，长度不限，交代清晰即可）

目标知识与能力

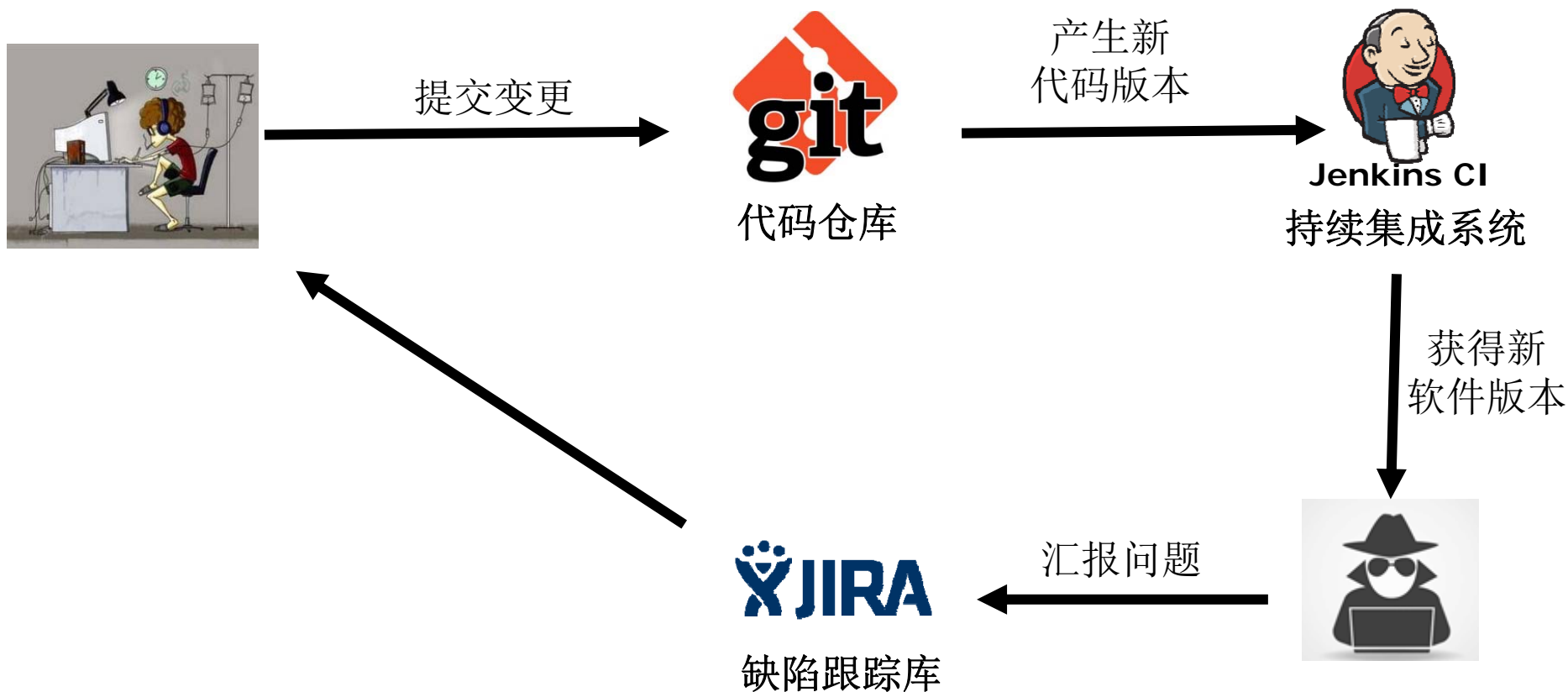
知识	能力
(1) 软件配置管理方法； (2) 软件自动化构建与测试； (3) 持续集成； (4) 缺陷的常见属性； (5) 缺陷管理的基本流程。	(1) 使用现代工具：使用配置管理、持续集成、缺陷管理工具； (2) 项目管理：能够应用持续集成方法管理软件开发与测试相关过程，能够围绕缺陷实施项目管理。

评价要素

要素	实验要求
环境部署	有效构建环境，体现软件应用能力
持续集成中的冒烟测试	能够编写项目配置，使得可以在持续集成中运行测试，并通过测试发现问题
缺陷提交	提交有价值的缺陷报告，尽量包含缺陷复现与修复所需信息，同时避免冗长而罗嗦
README	简洁地描述项目的必要开发管理信息

尽量用体现个人特征的账号等

(环境可通过虚拟机拷贝等方式共享，也可使用云服务商提供的
相关功能。推荐自行安装一两个工具，增进对测试工具了解)



候选工具:
Redmine、MantisBT
Github等上的各种issue tracking功能

提示：可以从bitnami下载软件配置好的版本进行安装。<https://bitnami.com/stacks>
尽量在虚拟机(VMWare/VirtualBox)中安装，注意虚拟机内存应设置2G以上，一般为Linux系统，否则容易有奇怪错误。

实验三：黑盒测试

- 采用黑盒测试方法（例如用例场景法、等价类法、边界值法、决策表法、正交实验设计法等）为以下软件之一设计测试用例。
 1. 分析待测对象的输入、输出结构和行为特征，分析其适合采用哪些方法来进行测试，说明如何将多种测试方法串联使用；
 2. 按步骤(1)给出的测试思路开展测试用例设计，并执行测试，获得测试结果；
 3. 分析测试结果，列出所发现的软件缺陷，并尝试对缺陷的可能成因进行分析。

（Word画表格较复杂，可用Excel画表后贴入到Word中）

计算日期间隔服务. <http://rili.160.com/tools/datecalculator.html>

房贷计算器. <https://www.fangdaijisuanqi.com/>

所得说计算. <http://www.gerensuodeshui.cn/index.html>

外汇储蓄计算器. <https://www.itouzi.com/whcx-jsq>

电脑功耗计算器. <http://tools.jb51.net/jisuanqi/computerjsq>

瓷砖用量计算器. <http://tool.520101.com/changyong/cizhuanmianji/>

涂料用量计算. <http://www.zx123.cn/zxjsq/tuliao/>

如果上述软件无法访问，也可百度搜索个类似的，并在
实验报告中说明具体实验对象地址

目标知识与能力

知识	能力
(1) 各类黑盒测试方法；	(1) 问题分析：分析软件的结构和行为特征，形成测试方法选择的结论； (2) 设计/开发解决方案：黑盒测试用例设计；

评价要素

要素	实验要求
软件特征分析	是否有效总结软件在输入和输出结构、行为等多方面特征；
测试方法设计	是否有清晰的测试思路，并合理衔接多种测试方法； 有否考虑软件的功能特征和失效风险
测试实施	能否根据被测软件特征，灵活地应用测试技术，将测试思路落实为具体的实施过程，并体现清晰的测试设计
结果分析	能否在测试结果数据的基础上对软件的质量问题进行概括，形成测试结论
缺陷成因推测	能否合理推测缺陷成因，为后续测试和开发吸收经验

实验四：覆盖测试

- 要求：

选择你过去编写过的任意一个**50**行以上程序，尝试为其生成测试用例，实现语句覆盖、分支覆盖。

- 请保持程序原生态，尽量不要为测试目的而去改变程序。
- 允许未实现**100%**覆盖，但须解释为什么目前无法实现，讨论可能的解决途径。
- 可使用工具去度量覆盖情况（如Eclipse CodeCover、Emma、VC、Intelij Idea等），手工分析亦可

实验五： 自动化测试

任选一个Web应用开展测试，要求：

1. 分别使用Selenium工具和Jmeter工具编写或录制测试脚本，**解释脚本结构**，执行脚本，**并分析比较两种脚本的差异**
(**Selenium工具请录制Python脚本**。Jmeter脚本可使用Badboy等第三方工具录制，具体请自行百度。JMX脚本可用文本编辑器打开查看)

实验六：性能测试

任选一个Web应用（推荐选择自己编的系统），利用JMeter对其开展性能测试，具体步骤要求如下：

- 分析待测应用的主要功能场景和性能需求；
- 针对至少三个典型功能场景，手工操作测试过程，获得测试结果；将上述功能场景分别录制为初始测试脚本；
- 在脚本中增加检查点，利用检查点判断测试结果是成功还是失败；
- 对脚本中的用户名、密码等信息进行参数化，似的脚本能够模拟不同虚拟用户同时登录的情况；
- 在以上脚本设定的基础上，定义多种负载规模 and 变化策略，以模拟不同软件工作场景；
- 执行性能测试，分别获得响应时间、吞吐量等性能指标，解读这些性能指标，对照步骤(1)中的性能需求，形成关于应用性能的评价。

目标知识与能力

知识	能力
(1) 功能与性能需求； (2) 并发性能测试的基本方法； (3) 性能测试脚本，以及脚本的参数化、检查点、集合点、事务等； (4) 性能指标。	(1) 设计/开发解决方案：能够识别系统性能需求，能够对 Web 应用实施并发性能测试； (2) 研究：在多种环境下开展性能测试实验，获得性能表现数据，综合数据，形成关于应用性能表现的结论； (3) 使用现代工具：使用并发性能测试工具；

评价要素

要素	实验要求
性能需求分析	应能够分析被测应用的性能需求，并围绕 <u>该需求</u> 开展整个性能测试活动。
性能测试设计	应能够围绕性能需求开展测试设计，包括确定测试中采用的负载规模、负载变化策略等。
性能测试实施	掌握实施性能测试的方法，包括录制脚本、设置检查点、开展测试过程参数化等； 能够解读性能测试脚本，并由此形成对性能测试工具运行原理的初步认识。
性能数据分析	能够基于多种指标数据，分析和概括被测软件的性能表现，形成对软件的性能评价。

关于参加全国大学生软件测试竞赛

- 自愿参加，不强求
- 参加练习赛，且工作量基本相当，可替代一个接近的实验
应按实验方式撰写参赛报告，说明怎么参赛的，做了什么样的题，怎么做的，结果如何。将按题目和完成成绩评分
- 通过预选赛，成功入围省赛可替代两个实验，该项实验评为优
需撰写形式完整的参赛报告来代替实验报告
- 取得省赛二等奖以上成绩实验课程评为优，理论课平时成绩评为优，可用参赛报告代替三个实验(<50%); 其它获奖视获奖比例加分。
- 获全国奖(三等以上) 理论课加1-5分。

(以上为一般情况，如获奖面过大，奖励可能适当降低，而获奖面小，则奖励也可能提高。)

2021年全国大学生软件测试大赛

<http://www.moocetest.org/#/NoticeDetail/146>

大赛报名通道将于8月10日开启，届时请登录大赛官网慕测平台（moocetest.org）报名。

预选赛：10月22-23日。互联网比赛形式。参赛选手可任选赛项参赛，在时间不冲突的前提下可参加多个分项赛。

省赛：11月6-7日，现场赛

总决赛：11月20-21日，现场赛，陕西西安、湖南衡阳（学院可资助）

时间 赛 项	10 月 23 日		10 月 24 日
	8: 30-12: 30	13: 00-16: 00	9: 00-12: 00
嵌入式测试	✓		
Web 应用测试		✓	
Web 安全测试		✓	
移动应用测试	✓		
开发者测试			✓

138	计算机科学与技术学院	全国大学生软件测试大赛	教育部软件工程专业教学指导委员会、全国高等院校计算机基础教育研究会、中国计算机学会软件工程专业委员会、中国软件测评机构联盟等	学科知识类	Ⅱ级甲等	2020 年认定
-----	------------	-------------	--	-------	------	----------

完成时间要求

- 理论课考核在第**10**周左右，如有人进入全国赛，则将在全国赛结束完成评分
- 实验课考核在**13-14**周左右

机房安排

- 待定