

软件工程专业纸(2)

学号：221220144

姓名：张路远

邮箱：craly199@foxmail.com

1. 在你做过的项目中：

a. 做过哪些事情用于保障软件的质量？

模块化设计、编码规范、注释与文档、日志记录与监控、版本控制和静态代码分析等

b. 还可以做哪些事情用于保障软件的质量？

需求文档化、设计模式应用、单元测试、自动化构建与部署、性能测试、规范化错误处理和资源优化等

2. 请用等价类划分法设计测试用例

某编译程序的规约中关于标识符的规定如下：

a. 标识符是由字母开头，后跟字母或数字的任意组合构成；

b. 标识符的字符数为1~8个；

c. 标识符必须先说明后使用；

d. 一个说明语句中至少有一个标识符；

e. 保留字不能用作变量标识符。

有效测试样例：

样例类A：无数字有效标识符，例如 `var`

样例类B：含数字有效标识符，例如 `var1`，`var1test`

样例类C：单标识有效声明语句，例如 `int var1;`

样例类D：多标识有效声明语句，例如 `int var1, var1test;`

无效测试样例：

样例类E：数字开头的标识符，例如 `1test`

样例类F：含非法字符的标识符，例如 `var@1`，`var 1`

样例类G：字符数不合法的标识符，例如 `abcd12345`，`<空字符串>`

样例类H：未声明的标识符，例如 `var=5; int var;`

样例类I：保留字标识符，例如 `int return;`

样例类J：说明语句无标识符，例如 `int;`

3. 一个计算第二天日期的 NextDate 程序，规定输入的年、月、日的变量分别为 month、day、year，相应的取值范围为 $year \in [1950, 2050]$ ， $month \in [1, 12]$ ， $day \in [1, 31]$ ，请使用边界值

分析方法设计测试用例

用例ID	年份	月份	日期	预期输出	说明
TC1	1950	1	1	1950年1月2日	最小合法值，测试年份、月份、日期的下边界
TC2	2050	12	31	2051年1月1日	最大合法值，测试年份、月份、日期的上边界
TC3	1949	1	1	无效输入	年份小于1950（无效）
TC4	2051	1	1	无效输入	年份大于2050（无效）
TC5	1950	0	1	无效输入	月份小于1（无效）
TC6	1950	13	1	无效输入	月份大于12（无效）
TC7	1950	2	29	1950年3月1日	非闰年2月29日（无效）
TC8	2000	2	29	2000年3月1日	闰年2月29日（有效）
TC9	1950	2	28	1950年3月1日	非闰年2月28日（有效）
TC10	1950	4	30	1950年5月1日	30天的月份（有效）
TC11	1950	4	31	无效输入	4月没有31号（无效）
TC12	1950	1	31	1950年2月1日	31天的月份（有效）
TC13	2050	12	31	2050年1月1日	最大合法日期，12月31日（有效）
TC14	1950	12	32	无效输入	12月没有32号（无效）
TC15	2050	2	29	2050年3月1日	非闰年2月29日（无效）

4. 谈谈：

a. 你对单元测试、集成测试、确认测试的理解

单元测试：用于验证代码中最小的单元，例如函数或方法，是否按预期工作。通常通过自动化测试框架进行，可以快速执行并频繁运行，这样在开发阶段就能尽早发现问题，具有高效、精确、成本低的优点

集成测试：用于验证不同模块或系统组件之间的交互是否正常，通过测试模块之间的接口，确保不同部分组合起来时能够顺利协作。测试通常关注数据流、功能流、外部系统的交互等，可

能涉及数据库、网络或第三方服务。可以通过自动化或者手动测试进行，但自动化测试比较适合那些接口清晰的集成模块

确认测试：用于确认软件是否符合用户需求或业务需求，由最终用户、客户或质量保证团队执行，重点是验证软件功能是否满足最初的需求文档或业务目标。测试通常是黑盒测试，不关心内部实现，只关注软件是否符合用户预期。测试可以通过模拟用户的操作流程，确认软件是否可以交付使用

b. 如何在你的项目应用这三个测试策略

单元测试：作为一项基础工作。在开发阶段，我会尽可能对每个独立的模块或功能进行**单元测试**，在每编写完一个功能或模块之后，立即编写单元测试来验证其正确性。如有可能，我会使用自动化测试工具，例如 `pytest`

集成测试：作为阶段性工作。当项目中的多个模块或功能实现后，我会进行集成测试，确保这些模块能够协同工作。测试时，我会先将多个功能模块集成在测试框架中，确保它们能够顺利交互；对于使用外部服务或数据库的模块，我会尝试使用模拟技术来模拟这些外部系统，这样测试时就可以不依赖真实环境

确认测试：作为最后一项工作。在功能开发完成后，由我（团队），或是用户进行测试，确保项目满足用户需求。在开发开始前，我会先明确项目需求和业务目标；而项目接近完成时，我会手动执行确认测试，模拟用户使用场景，测试软件的核心功能是否达到需求，这可能会用到自行编写的测试样例