

2021 软件体系结构 大作业要求

软件体系结构课程组

2021 年 5 月 20 日

1 作业内容

主题：分析开源软件系统的体系结构

内容：从课程组提供的开源软件系统中选择一个，组队分析并恢复该软件系统的体系结构

基本要求：提交恢复后的软件体系结构模型和恢复过程文档，需要包含以下内容

1. 软件体系结构模型包括对软件系统各模块的**分解和功能描述**，以及用**文件目录树**形式给出各模块相对应的包、类或源文件，和以**模块关系图**形式给出的高层模块间的依赖关系图，具体见 2.2.3 中的要求
2. 恢复过程文档主要包括软件系统简介、需求分析、软件体系结构分析与恢复、设计特色分析等部分，具体见 2.2 中的要求
3. 恢复过程文档需详细描述组内分工情况，包括小组成员、分工方式、工作内容、工作量占比等

2 详细要求

2.1 组队与软件系统选择

- 本次作业可选择的开源软件系统见附录 A
- 本次作业每个小组最多有 **3** 个成员，同一个软件系统可以有不多于 **5** 个小组同时选择
- 在完成作业的过程中，组内成员应做到互相商讨后再进行决策，避免个人偏见
- 小组之间不得互相抄袭，一经发现，严肃处理

2.2 恢复过程文档要求

恢复过程文档至少需要包含软件系统简介、需求分析、软件体系结构分析与恢复、设计特色分析和组内分工情况五个部分，各部分的要求如下：

2.2.1 软件系统简介

软件系统简介部分至少需要包括以下内容：

1. 软件系统的主要功能介绍
2. 软件系统的代码行数（对于多语言的软件系统，需要标注各语言的代码行数）
3. 软件系统中包和类的数量

2.2.2 需求分析

由于需求是软件体系结构设计的重要依据，因此在分析并恢复软件系统的体系结构之前，需要先对软件系统的需求进行详细分析。需求分析部分至少需要包括以下内容：

1. 软件系统的功能需求，使用用例图、活动图等 UML 模型描述
2. 软件系统的主要质量属性（非功能需求），以分析表的形式给出主要质量属性场景
3. 其它对该软件系统进行体系结构设计时需要考虑的需求，如商业质量属性等

2.2.3 软件体系结构分析与恢复

1. 软件体系结构的恢复和建模

查阅相关资料并结合软件系统源代码，以需求为依据，恢复出软件体系结构的模型。具体要求如下：

- 1) 给出**软件体系结构的模块及其功能描述**，采用自上而下逐层分解、自下而上逐层聚合或两者结合的方法，得到软件体系结构的模块，并将**源代码中的包、类或源文件等放入对应的模块中**。模块之间存在包含关系，上层模块包含下层模块，**每个模块所包含的子模块原则上不超过 12 个**，最底层的模块直接包含不可再拆分的包、类或源文件。每个模块的命名要逻辑合理、并**列表给出所有模块的功能描述**。需要说明的是，部分包可以直接作为模块，但由于体系结构模块和包/目录结构并不完全对应，因此应避免将所有的包/目录结构直接作为体系结构的模块。另外，如果某个模块中包、类或源文件的数量过多，需要对该模块再次分层，并给出合适的命名
- 2) 要求以**文件目录树**形式给出各模块相对应的包、类或源文件，每个模块、包、类或源文件均需要**合理命名**。文件目录树中各元素的命名方式如表1所示。对于文件目录树，提交时需要提交**完整的树形文件目录**，其中包含体系结构模型各层各个模块对应的文件夹，及各个模块内元素所对应的代码实体（包、类、源文件等）
- 3) 画出软件系统体系结构模型中第 1~3 层各模块的**模块关系图**。例如，软件系统 A 有两个顶层模块 M_1 和 M_2，其中 M_1 有四个子模块 M_1_{1~4}，M_2 有五个子模块 M_2_{1~5}，则在画第 1 层的模块关系图时需要画出 M_1 和 M_2 之间的模块关系图，在画第 2 层的模块关系图时需要画出 M_1_{1~4}

之间的模块关系图和 M_2_{1~5} 之间的模块关系图，以此类推。模块关系图中展现出来的模块结构需要能够与通过文件目录树给出的体系结构模型完全对应。在提交时，各个模块关系图以模块所在的层级和该图中模块的公共父模块命名。例如，展示 M_1 和 M_2 间关系的图应命名为 1-A-模块关系图，展示 M_1_{1~4} 间关系的图应命名为 2-M_1-模块关系图，展示 M_2_{1~5} 间关系的图应命名为 2-M_2-模块关系图。提交的模块图应为矢量图或高分辨率的位图

表 1: 软件体系结构模型元素命名方式

元素	命名方式
模块	M_ 模块名称
包	P_ 包名称
类	C_ 类名称
源文件	S_ 源文件名称

2. 软件体系结构的分析

- 1) 以表格的形式说明所分析的需求和得到的模块、软件系统源代码之间的对应关系，即哪些模块对应哪些需求，哪些源代码的文件对应哪些模块
- 2) 结合相关资料，以需求为依据，追踪到代码，详细说明每层、每个模块的分解或聚合理由、考虑的问题与角度（例如：这样划分模块提高了软件系统的可维护性）
- 3) 分析每层、每个模块采用的软件体系结构风格、策略/战术、设计模式、领域框架等，反向推测该层或该模块采用该技术的理由、优势和问题，分析其特点和优势，并讨论是否存在更好的方法或技术。可以使用 UML 模型进行描述，如类图、顺序图、活动图等
- 4) **发现体系结构模型中的异味 (Bad Smell)**：如果在体系结构模型的恢复过程中，发现体系结构模型中存在异味，即可能对软件系统的架构带来负面影响的情况，则应列出并给出说明和你的建议。可能存在但不限于以下几种异味：
 - i. 一个包跨越了多个模块（即包中的元素在不同的模块中），同时多个模块中除了该包中的元素以外还有其它的包、类或源文件存在
 - ii. 代码元素无法与任何一个模块对应
 - iii. 模块中存在无对应代码实现的元素
- 5) 对软件体系结构进行建模分析时，如果采用 UML，请按照 UML 2.0 以上标准建模；如果不采用 UML 标准，则需要给出图例和说明

2.2.4 设计特色分析

- 1) 分析该开源软件系统的体系结构设计中你认为最值得借鉴的地方、与其它同类产品相比之下更具有优势的设计特色，说明这些优势体现在哪里、支撑哪些功能需求与非功能需求
- 2) 分析该开源软件系统存在哪些可以改进的地方

2.2.5 组内分工情况

在这部分中需要说明本次作业的分工情况，包括小组成员、分工方式、工作内容、工作量占比等。

2.2.6 参考文献

此部分中需要给出在分析和恢复软件系统的体系结构期间所参考的文档或文献，如官方设计文档、博客、专著、论文等，按照标准格式列出，且需要在正文中有所引用。

3 其它要求

- 提交作业时，各组组长提交即可，其他成员无需重复提交
- 本次作业需要提交的文档一共有三个部分，分别为用文件目录树表示的软件体系结构模型、存放所有模块关系图的文件夹和软件体系结构恢复过程文档。将这三部分文档放入一个文件夹中，命名该文件夹为所选软件系统名-组号-组长姓名-组长学号-大作业，将该文件夹压缩为.zip 文件后提交
- 作业提交网址：[6 系教辅平台 \(Judge 系统\)](#)，校外可通过北航 VPN 访问
- 作业提交截止时间：**6 月 9 日晚 23:55**
- 及时提交作业：未在规定时间内提交，则根据补交时间扣除相应分数，迟交 1 天扣除所得分数的 10%，迟交 2 天扣除所得分数的 20%，以此类推，扣完为止

附录

A 可选开源软件系统

A.1 PyTorch

- Python 机器学习库
- 源码地址: [GitHub - pytorch/pytorch](#)

A.2 Django

- Web 应用框架
- 源码地址: [GitHub - django/django](#)

A.3 Tomcat

- Servlet 容器与 Web 应用服务器
- 源码地址: [GitHub - apache/tomcat](#)

A.4 V8

- 高性能 JavaScript 引擎
- 源码地址: [Google Git - chromium/v8](#)

A.5 Qt Base

- 跨平台 C++ 图形用户界面应用程序开发框架 Qt 的基础核心模块
- 源码地址: [GitHub - qt/qtbase](#)

A.6 PostgreSQL

- 一个开源的数据库管理系统
- 源码地址: [GitHub - postgres/postgres](#)

A.7 React

- 一个将数据渲染为 HTML 视图的开源 JavaScript 库
- 源码地址: [GitHub - facebook/react](#)

A.8 Visual Studio Code

- 一个轻量级代码编辑器
- 源码地址: [GitHub - microsoft/vscode](#)

A.9 Eclipse Sirius

- 基于 Eclipse 的图形化建模框架
- 源码地址: [Eclipse - org.eclipse.sirius.git](#)

A.10 Mainflux

- 使用 Go 语言编写的工业物联网消息传递和设备管理平台
- 源码地址: [GitHub - mainflux/mainflux](#)

A.11 OpenCASCADE

- CAE 几何造型基础软件平台 (FreeCAD 的核心模块)
- 源码地址: [GitHub - tpaviot/oce](#)

A.12 Gmsh

- 三维有限元网格生成器
- 源码地址: [GitLab - Gmsh](#)

A.13 MeshLab

- 三维立体网格处理软件
- 源码地址: [GitHub - cnr-isti-vclab/meshlab](#)

A.14 OpenFOAM

- 求解流体力学相关问题的 CAE 软件
- 源码地址: [GitHub - OpenFOAM/OpenFOAM-7](#)

B 软件体系结构建模及提取工具

B.1 Papyrus

Papyrus 是一款开源的 UML 建模工具, 用户能够扩展模型的语义规则, 作为建模工具具有较高的灵活性。Papyrus 同样提供了逆向工程能力, 能够将 Java 项目的源码转

换为 UML 模型显示。

B.2 PlantUML

PlantUML 是一个可以使用简单的语法描述 UML 图的开源工具，也可以简单有效的定制 UML 图的样式。PlantUML 在 VSCode、IntelliJ Idea、Eclipse 等多款软件中均有插件支持。

B.3 StarUML

StarUML 是一款开源的 UML 建模工具，能够根据类图正向生成编程语言代码，也能够从源码逆向获取 UML 类图，支持 Java、C++ 等编程语言。

B.4 Enterprise Architect (EA)

EA 是面向企业级开发的建模软件，能够支撑开发团队从需求分析到系统设计的各个阶段。EA 同样支持 UML 图和编程语言代码的正、逆向转换，能够根据系统设计人员创建的模型生成框架。用户将 Java 工程导入作空间时，可以自动分析代码结构，为每一层包自动逆向生成类图和包图，展示源码中包含的类元素、方法和关系。

B.5 Scientific Toolworks Understand

Understand 是一款定位于代码阅读的软件，支持多语言如 Ada、C、C++、C#、Java、FORTRAN、Delphi、Jovial 和 PL/M 等。支持代码语法高亮、代码折迭、交叉跳转、书签等基本阅读功能。另外，可以对整个项目的架构、度量等指标进行分析并输出报表。支持对代码生成多种图 (butterfly graph、call graph、called by graph、control flow graph、UML class graph 等)，在图上点击节点可以跳转到对应的源代码位置。

B.6 Source Insight

Source Insight 支持多种语言的代码分析，包括 C，C++，ASM，HTML 等，能够自动创建并维护它自己高性能的符号数据库，包括函数、方法、全局变量、结构、类和工程源文件里定义的其它类型的符号，对于大工程的源码阅读非常方便。