

# 杭州电子科技大学

## 硕士学位论文

题目：基于  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$  开发的杭州电子科技大学硕士学位  
论文  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  模板

研究生 申智能

专业 凝聚态物理

指导教师 叶芷晴 教授

完成日期 2025 年 1 月

杭州电子科技大学硕士学位论文

基于  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$  开发的杭州电子科技大学硕士  
学位论文  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  模板

研 究 生：申 智 能

指导教师：叶 芷 晴 教授

2025 年 1 月

**Dissertation Submitted to Hangzhou Dianzi University  
for the Degree of Master**

**$\text{\LaTeX}$  Template for Master's Thesis at  
Hangzhou Dianzi University that was  
Developed Based on  $\text{\LaTeX}3$**

**Candidate: SAN Chi Nan**

**Supervisor: Prof. YIP Tsz Ching**

**January, 2025**

# 杭州电子科技大学

## 学位论文原创性声明和使用授权说明

### 原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：



日期：2025 年 05 月 31 日

### 学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署单位名称仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：



日期：2025 年 05 月 31 日

指导教师签名：



日期：2025 年 06 月 01 日

## 摘要

设计模式被广泛用于解决软件设计和开发过程中反复出现的设计问题。设计模式在软件源码中的使用信息（即设计模式实例）能够在较大程度上反映软件系统的设计思路。准确地从软件源码中识别和挖掘设计模式实例能够帮助软件开发和维护人员快速理解软件系统的原始设计和实现，从而对软件系统的维护、升级更新和二次开发等工作提供方便。

近些年来软件工程领域提出了多种自动化或半自动化地挖掘设计模式实例的方法。其中部分方法将设计模式和软件源码转化为某种特定的图的形式，然后通过挖掘同构子图的方式挖掘设计模式实例。然而，由于同构子图挖掘问题是一个 NP 完全问题，这些方法的执行效率通常较低。

为了有效解决基于子图同构的设计模式挖掘方法的效率低下问题，本文提出了一种高效的基于快速搜索序列和行为模板的结构型设计模式挖掘方法。该方法首先将设计模式和软件源码转化为一种带权有向图，即类关系图，然后从软件源码的类关系图中找出所有与设计模式的类关系图相同构的子图，每个子图的顶点所表示的类组成了一个候选设计模式实例。在挖掘过程中，本文方法为每种设计模式构建了一个快速搜索序列，该快速搜索序列描述了设计模式的结构信息并指定了一个高效的搜索设计模式角色的次序，从而大大减少了挖掘过程的搜索空间。此外，本文为每种设计模式制定了一个特定的行为模板，用于过滤候选实例集合中不满足行为特征的错误实例，进一步提高了本文方法的准确率。最后，本文选取 4 个常用的开源软件系统进行验证实验。实验结果表明，针对本文制定的结果基准，该方法不仅能够达到 100% 的召回率和相对较高的准确率和 F-measure 值，而且显著提高了执行效率。

**关键词：**设计模式，设计模式挖掘，快速搜索序列，行为模板，子图同构

## Abstract

Design patterns are widely used to address the recurring design problems during the process of software design and development. The information about design patterns used in software source code-design pattern instances reveal much about the high-level abstract design ideas. Accurately identifying and detecting design pattern instances in a software system can help developers and maintainers to understand its original design and implementation, facilitating maintenance, update and re-development.

In recent years, many approaches have been proposed to automatically or semi-automatically detecting design pattern instances from software source code. Among these approaches, some approaches transform software source code and design patterns into certain graphs, and then exploit subgraph isomorphism techniques to detect design pattern instances. However, as mining isomorphic subgraphs is an NP-complete problem, those approaches usually fail to achieve satisfactory efficiency.

In order to effectively solve the inefficiency of the approaches which are based on subgraph isomorphism, we propose an efficient approach to detect structural design pattern instances based on quick-search sequences and behavior templates. In the proposed approach, we transform software source code and design patterns into a certain form of weighted and directed graph namely Class Relationship Graph, and then detect all the subgraphs from the source code graph which are isomorphic to the Class Relationship Graph of a certain design pattern. The classes that each isomorphic subgraph's vertices represent then consist of a candidate instance. During the process of detecting design pattern instances, we construct a quick-search sequence for each design pattern, which describes the structural characteristics of design pattern and specify an order to efficiently search the roles of each design pattern. According to the quick-search sequence, we can greatly reduce the search space. Afterwards, we construct a specific behavior template, which is used to filter the false positives in the candidate instance set and further improve the processing efficiency. Finally, we exploit our approach on four well-known open-source software systems. The results demonstrate that our approach not only achieves nearly 100% recall and relatively high precision and F-measure on the benchmark we construct, but also significantly.

**Keywords:** Design Pattern, Design Pattern Detection, Quick-Search Sequence, Behavior Template, Sub-graph Isomorphism

目录

摘要 ..... I

**Abstract** ..... II

第一章 绪论 ..... 1

    1.1 研究背景与意义 ..... 1

    1.2 国内外研究现状 ..... 1

## 第一章 绪论

### 1.1 研究背景与意义

随着信息技术的飞速发展和广泛应用，各行各业都积累了大量的软件系统。

### 1.2 国内外研究现状

设计模式挖掘的相关研究工作伴随着设计模式在软件工程领域的提出而应运而生，并得到了广泛的关注和研究。正如 Apostolos Ampatzoglou 的描述，关于挖掘软件源码中设计模式实例的研究是设计模式相关的所有研究工作中最受关注的一项研究课题。

#### 1.2.1 结构分析

结构分析方法是最常见的挖掘设计模式实例的方法。