# （一）基于API知识图谱的代码检测和修复技术

## 系统背景:

随着软件系统规模和复杂性的增加，软件开发成本和质量问题已经成为制约软件发展的关键因素。为确保代码质量、可读性和可维护性，开发人员制定了一系列代码规软件库通过应用程序编程接口（API）提供可重用的功能。API 通常附带使用规范来阐述一些契约、约束和准则，例如对调用顺序、特殊值和状态条件的约束。开发人员在学习和使用API时，经常会发现开发需求和可访问的API知识之间存在鸿沟。如果不深入了解API的使用知识，开发人员很可能会写出有缺陷或脆弱的代码。现有的与API误用相关的代码检测和修复技术仍然存在巨大的提升空间。例如，现有的检测工具覆盖率低、质量无法保证以及不可解释；自动程序修复工具无法为API误用提供正确且高效的修复补丁。鉴于此，本项目聚焦于基于API知识图谱的代码检测和修复技术。旨在通过构建高质量的API知识图谱，帮助开发人员理解API知识，提升自动化代码检测的精确度和修复的可用性，以达到辅助开发人员提高编码效率和代码质量。

## 主要功能

基于API知识图谱的代码检测和修复技术主要包含三个功能：

（1）静态API误用检测

API误用会导致软件崩溃、错误和漏洞。现有方法根据从代码库中挖掘的频繁API使用模式来检测API误用，这与真实的API使用规范在误用检测的覆盖率、检测质量以及可解释性方面存在很大的缺陷。因此，本项目拟构建一个较大规模的Java API误用数据集，并探究如何利用基于官方文档构建的API知识图谱中检测代码中的API误用。

（2）动态API误用检测

静态API误用检测可以成功识别特定类型的API误用，但这些方法的误报率很高，需要开发者手动检查分析产生的大量候选静态API误用。而基于搜索的软件测试（SBST）技术在检测程序错误方面存在搜索效率低，代码覆盖率低的局限性，不足以有效地检测程序中的错误。本项研究拟结合动静态检测的优点，通过构建API知识图谱，将其用于指导搜索测试，从而弥补上述局限性。因此，本项目拟研究如何利用API知识图谱静态识别隐性错误，发现检测入口，缩小搜索范围，从而有效地检测程序中的错误。

（3）API误用修复

检测到代码中的API误用后，本项目拟研究如何根据API知识图谱中的规范生成代码补丁来修复API误用，并提供相应的修复解释，进而开发出集API误用检测和修复一体的IDE插件。

## 开发环境

操作系统及版本：Windows 10 专业版

支撑软件及版本列表：PyCharm + Python-3.9 + IntelliJ + jdk-15.0.2

数据库环境：MongoDB

## **3.3.1**基本的架构体系

图 1 功能结构图

（1）API知识图谱的构建。具体来说，从官方文档和众智平台中收集高质量的Java SE&JDK API使用知识，通过自然语言处理等方式提取知识并构建知识图谱；

（2）静态API误用检测：研究利用知识图谱静态检测代码中的API误用。具体来说，通过静态代码分析技术，抽取代码信息，并与知识图谱中的API规范进行比对，进而检测是否包含潜在的误用；

（3）动态API误用检测：研究利用知识图谱动态检测代码中的API误用。具体来说，在静态检测发现API误用的地方注入相应异常处理代码，通过动态测试算法生成测试用例进行动态验证，发现真实存在的API误用；

（4）API误用修复：研究修复API误用。具体来说，利用静态代码分析技术，对比分析代码片段和API知识图谱中的使用规范，抽取误用片段。结合代码内容，将知识图谱中的使用规范转化为修复补丁并提供相应的解释；基于项目提出的理论模型和关键技术方法，研制开发者编程协助工具原型，并结合开发者日常应用场景，完成实际应用、验证和评估。