对Java代码的基本查询(Basic query for Java code)

你可以使用本地的Codeql或者LGTM平台进行对Java代码查询的学习

关于查询

我们将要运行的查询对代码进行基本搜索，以查找是否有多余的if语句（如果它们具有空的then分支）。例如，如下代码：

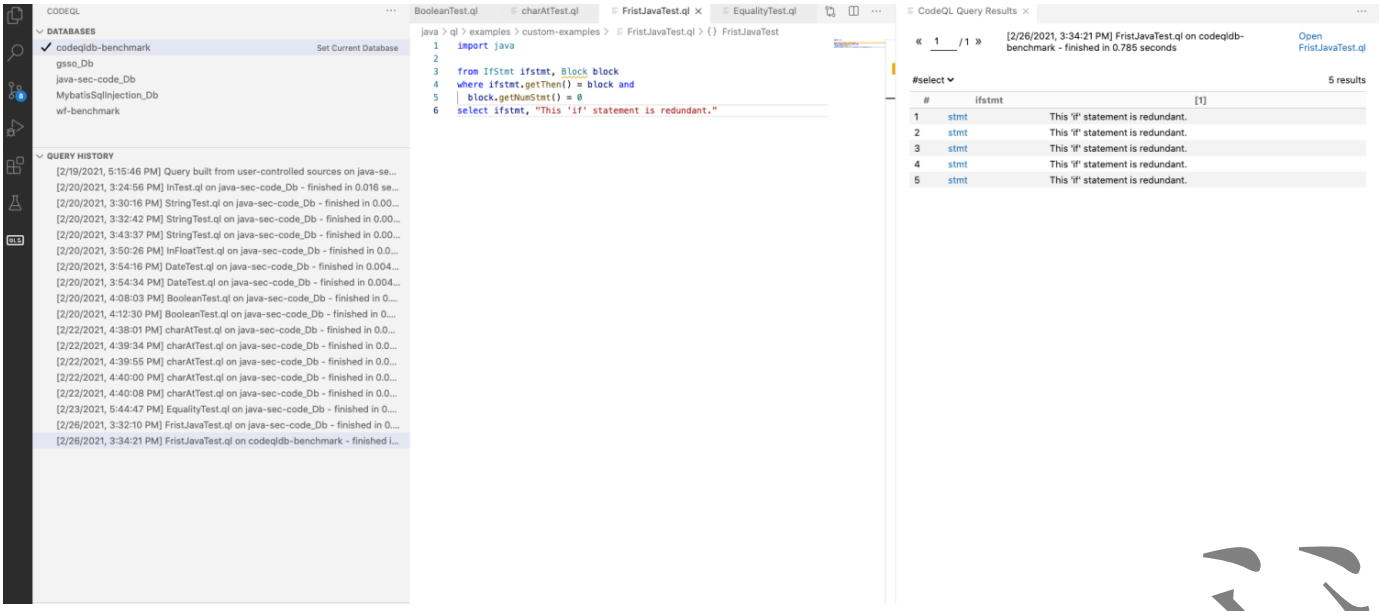
```
1 if (error) { }
```

运行查询语句

在VScode里运行

```
1 import java
2
3 from IfStmt ifstmt, Block block
4 where ifstmt.getThen() = block and
5   block.getNumStmt() = 0
6 select ifstmt, "This 'if' statement is redundant."
```

我们选取owasp的benchmark来进示例查询



查询将需要一些时间来返回结果。查询完成后，结果将显示在项目名称下方。查询结果列在两列中，分别对应于查询的select子句中的两个表达式。第一列对应于表达式ifstmt，并链接到项目源代码中ifstmt发生的位置。第二列是警报消息。

关于该查询语句的结构

在初始`import`语句之后，此简单查询包含三个部分，这些部分的作用与SQL查询的FROM，WHERE和SELECT部分相似。

查询部分	目的	细节
`import java`	导入Java的标准CodeQL库。	每个查询都以一个或多个`import`语句开头。
`from IfStmt ifstmt, Block block`	定义查询的变量。声明的形式为：`<type> <variable name>`	我们用： • 一个`IfStmt`变量`if`声明 • `Block`then块的变量
`where ifstmt.getThen() = block and block.getNumStmt() = 0`	在变量上定义条件。	`ifstmt.getThen() = block`关联两个变量。该块必须是`then`该`if`语句的分支。 `block.getNumStmt() = 0`指出该块必须为空（即，它不包含任何语句）。

查询部分	目的	细节
`select ifstmt, "This 'if' statement is redundant."`	定义报告每个匹配项的内容。`select` 用于查找编码实践不佳的实例的查询的语句始终为以下形式：`select <program element>, "<alert message>"`	`if`使用解释问题的字符串报告结果语句。

扩展语句

查询编写是一个固有的迭代过程。您编写了一个简单的查询，然后在运行它时发现了以前未曾考虑过的示例或改进的机会。

删除误报

浏览我们的基本查询的结果表明可以改进它。在结果中，您可能会发现`if`带有`else`分支的语句示例，其中空`then`分支确实可以达到目的。例如：

1、扩展where子句以包括以下额外条件：

```
1 and not exists(ifstmt.getElse())
```

该`where`子句现在是：

```
1 where ifstmt.getThen() = block and
2   block.getNumStmt() = 0 and
3   not exists(ifstmt.getElse())
```

2、点击**运行**。

现在有更少的结果，因为不再包含`if`带有`else`分支的语句。