

模块 (Modules)

模块 (Modules)

定义模块(Defining a module)

模块种类(Kinds of modules)

文件模块(File modules)

库模块(Library modules)

查询模块 (Query modules)

明确的模块 (Explicit modules)

模块主体(Module bodies)

导入模块(Importing modules)

import语句

小结

模块 (Modules)

模块通过将相关类型，谓词和其他模块组合在一起，提供了一种组织QL代码的方法。

您可以将模块导入其他文件，从而避免重复，并有助于将代码结构化为更易于管理的部分。

定义模块(Defining a module)

有多种方式来定义模块，这里是最简单的方法的例子，声明一个 [明确的模块](#) 名为 `Example` 含有类

`OneTwoThree`：

```
1 module Example {
2   class OneTwoThree extends int {
3     OneTwoThree() {
4       this = 1 or this = 2 or this = 3
5     }
6   }
7 }
```

- 模块的名称可以是任何 以大写或小写字母开头的 [标识符](#)。
- `.ql` 或 `.qll` 文件也隐式定义模块。有关更多信息，请参见“[模块种类](#)”。

您也可以注释模块。有关更多信息，请参见“[注释概述](#)”。

注意，您只能注释[显式模块](#)。无法注释文件模块。

模块种类(Kinds of modules)

文件模块(File modules)

每个查询文件（扩展名 `.ql`）和库文件（扩展名 `.qll`）都隐式定义了一个模块。模块的名称与文件相同，但文件名中的所有空格均由下划线（`_`）代替。文件的内容构成模块的[主体](#)。

库模块(Library modules)

库模块由 `.qll` 文件定义。除select子句外，它可以包含下面[模块主体](#)中列出的任何元素。

例如，考虑以下QL库：

```
1 class OneTwoThree extends int {
2   OneTwoThree() {
3     this = 1 or this = 2 or this = 3
4   }
5 }
```

该文件定义了名为 `OneTwoThreeLib` 的库模块。该模块的主体定义了class `OneTwoThree`。

查询模块 (Query modules)

查询模块由 `.ql` 文件定义。它可以包含下面的[模块主体](#)中列出的任何元素。

查询模块与其他模块略有不同：

- 查询模块无法导入。
- 一个查询模块在其[名称空间](#)中必须至少有一个查询。这通常是[select子句](#)，但也可以是[查询谓词](#)。

例如：

```
1 import OneTwoThreeLib
2
3 from OneTwoThree ott
4 where ott = 1 or ott = 2
5 select ott
```

该文件定义了名为的查询模块 `OneTwoQuery`。该模块的主体由 [import语句](#)和[select子句](#)组成。

明确的模块 (Explicit modules)

您也可以在另一个模块中定义一个模块。这是一个明确的模块定义。

显式模块定义为关键字，`module` 后跟模块名称，然后使用大括号将模块主体括起来。除select子句外，它可以包含下面“[模块主体](#)”中列出的任何元素。

例如，您可以将以下QL代码段添加到[上面](#) 定义的库文件`OneTwoThreeLib.qll`中：

```
1 ...
2 module M {
3     class OneTwo extends OneTwoThree {
4         OneTwo() {
5             this = 1 or this = 2
6         }
7     }
8 }
```

这定义了一个名为的显式模块 `M`。该模块的主体定义了class `OneTwo`。

模块主体(Module bodies)

一个模块的主体是模块定义内的代码，例如，类 `OneTwo` 中的[显式模块](#) `M`。

通常，模块的主体可以包含以下构造：

- [导入语句](#)
- [谓词](#)
- [类型](#)（包括用户定义的[类](#)）
- [别名](#)
- [显式模块](#)
- [Select子句](#)（仅在[查询模块](#)中可用）

导入模块(Importing modules)

将代码存储在模块中的主要好处是可以在其他模块中重用它。要访问外部模块的内容，可以使用[import语句](#)导入模块。

导入模块时，这会将其名称空间中的所有名称（除了[专用名称](#)之外）都带入当前模块的[名称空间](#)中。

import语句

导入语句用于导入模块。它们的形式为：

```
1 import <module_expression1> as <name>
```

```
2 import <module_expression2>
```

导入语句通常在模块的开头列出。每个import语句导入一个模块。您可以通过包含多个import语句（要导入的每个模块一个）来导入多个模块。import语句也可以用`private`注释。

您可以使用`as`关键字以其他名称导入模块，例如。`import javascript as js`

小结

本节我们学习了QL语言里模块的概念，模块是将类、谓词、其他模块组合在一起，为某一种特定的业务进行数据集合及查询方法。后续我们将重点学习QL语言为Java安全扫描提供的内置模块方法

58安全应急响应中心