查询（Querie）

查询（Querie）

查询是QL程序的输出。他们评估结果集。有两种查询。对于给定的查询模块，该模块中的查询为：

- 在该模块中定义的select子句（如果有）。
- 该模块的谓词 命名空间中的任何查询谓词。也就是说，它们可以在模块本身中定义，也可以从其他模块导入。

我们通常还将整个QL程序称为查询。

Select子句（Select clauses）

编写查询模块时，可以包括以下形式的**select子句**（通常在文件末尾）：

```
from /* ... variable declarations ... */
where /* ... logical formula ... */
select /* ... expressions ... */
```

在`from`与`where`部分是可选的

select子句中，除了“表达式”中描述的表达式外，select子句还可以包括：

- as关键字，后面可以接搜索结果的别名（类似sql），并允许您在后续的选择表达式中使用它们
- order by 关键字（类似sql的 order by），随后是结果列的名称，以及任选的关键字或。这确定了显示结果的顺序。`order by` `asc` `desc`

例如：

```
from int x, int y
where x = 3 and y in [0 .. 2]
select x, y, x * y as product, "product: " + product
```

结果：

x	y	product	
3	0	0	product: 0
3	1	3	product: 3
3	2	6	product: 6

您也可以在select子句的末尾添加。现在，结果根据列中的值以降序排列：`order by y desc y`

x	y	product	
3	2	6	product: 6
3	1	3	product: 3
3	0	0	product: 0

查询谓词（Query predicates）

查询谓词是带有 注释的非成员谓词`query`。它返回谓词求值的所有元组。

例如：

```
query int getProduct(int x, int y) {
  x = 3 and
  y in [0 .. 2] and
  result = x * y
}
```

该谓词返回以下结果：

x	y	result
3	0	0
3	1	3
3	2	6

编写查询谓词而不是select子句的好处是，您也可以在代码的其他部分调用谓词。例如，您可以`getProduct`在类的主体内部进行调用：

```
class MultipleOfThree extends int {
  MultipleOfThree() { this = getProduct(_, _) }
}
```

相反，select子句就像一个匿名谓词，因此您以后不能调用它。
`query`在调试代码时将注释添加到谓词也可能会有帮助。这样，您可以显式查看谓词要求值的元组集。

小结

本节我们学习了，在QL语言中的Select语句，我们可以看到，它与SQL语言的语法存在高度的相似，我们理解QL的查询语句时，也可以通过集合的概念去帮助理解QL语句的含义