面向对象UML系列第一次作业指导书

注意

请不要提交官方包代码！其代码仅供参考，请使用已经编译打包好的jar文件。

因提交官方包源代码导致的任何问题，请自行承担后果。

摘要

本次作业，需要完成的任务为实现一个UML类图分析器`UmlInteraction`，学习目标为**UML入门级的理解、UML类图的构成要素及其解析方法**

问题

基本目标

本次作业最终需要实现一个UML类图分析器，可以通过输入各种指令来进行类图有关信息的查询。

基本任务

本次作业的程序主干逻辑（包括解析`mdj`格式的文件为关键数据）我们均已实现，只需要同学们完成剩下的部分。

需要实现`UmlInteraction`这个官方提供的接口，来实现一个`UmlInteraction`解析器

`UmlInteraction`的**接口定义源代码**都已在接口源代码文件中给出，各位同学需要实现相应的接口，并保证**代码实现功能正确**。

具体来说，各位同学需要新建一个类`MyUmlInteraction`（仅举例，具体类名可自行定义并配置），并实现相应的接口方法。

当然，还需要同学们在主类中调用官方包的`AppRunner`类，并载入自己实现的`MyUmlInteraction`类，来使得程序完整可运行，具体形式下文中有提示。

测试模式

本次作业**不设置互测环节**。针对本次作业提交的代码实现，课程将使用公测+bug修复的黑箱测试模式，具体测试规则参见下文。

代码结构说明

UmlInteraction类

`UmlInteraction`的具体接口规格见官方包的jar文件，此处不加赘述。

除此之外，`UmlInteraction`类必须实现一个构造方法

```
1 public class MyUmlInteraction implements UmlInteraction {
2     public MyUmlInteraction(UmlElement[] elements);
3 }
```

或者

```
public class MyUmlInteraction implements UmlInteraction {
    public MyUmlInteraction(UmlElement... elements);
}
```

构造函数的逻辑为将`elements`数组内的各个UML类图元素传入`UmlInteraction`类，以备后续解析。

请确保构造函数正确实现，且类和构造函数均定义为`public`。`AppRunner`内将自动获取此构造函数进行`UmlInteraction`实例的生成。

（注：这两个构造方法实际本质上等价，不过后者实际测试使用时的体验会更好，想要了解更多的话可以百度一下，关键词：`Java 变长参数`）

交互模式

交互的模式为：

- 测试调用类调用上述构造函数，生成一个实例，同时将UML模型元素传入。
- 之后测试调用类将根据输入输出调用各个接口方法，以实现基于之前传入的UML模型元素的各类查询操作。

输入输出

本次作业将会下发`mdj`文件解析工具、UmlInteraction类、输入输出接口（实际上为二合一的工具，接口文档会详细说明）和全局测试调用程序，其中输入输出接口、全局测试程序均在官方接口中。

- 解析工具用于将`mdj`格式文件解析为输入输出接口可以识别的格式，该格式包含了文件内模型中所有关键信息的元素字典表
- 输入输出接口用于对元素字典表的解析和处理、对查询指令的解析和处理以及对输出信息的处理
- 全局测试调用程序会实例化同学们实现的类，并根据输入接口解析内容调用对应方法，并把返回结果通过输出接口进行输出

输入输出接口的具体字符格式已在接口内部定义好，各位同学可以阅读相关代码，这里只给出程序黑箱的字符串输入输出。

规则

- 输入一律在标准输入中进行，输出一律向标准输出中输出
- 输入内容以指令的形式输入，一条指令占一行，输出以提示语句的形式输出，一句输出占一行
- 输入使用官方提供的输入接口，输出使用官方提供的输出接口
- 输入的整体格式如下：
    - 由`mdj`文件解析而来的关键元素表
    - `END_OF_MODEL`分隔开行
    - 指令序列，每条指令一行

指令格式一览

各个指令对应的方法名和参数的表示方法详见官方接口说明。

模型中一共有多少个类

输入指令格式：`CLASS_COUNT`

举例：`CLASS_COUNT`

输出：

- `Total class count is x.` x为模型中类的总数

类中的操作有多少个

输入指令格式：`CLASS_OPERATION_COUNT classname modes`

举例：`CLASS_OPERATION_COUNT Elevator NON_RETURN`

输出：

- `Ok, operation count of class "classname" is x.` `x`为模型中指定类型的操作个数
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个
- `Failed, conflict modes.` 模式之间存在冲突

说明：

- `modes`表示查询限制条件，可能会有多个或0个，数据类型为`OperationQueryType`，取值为：
  - `NON_RETURN` 无返回值操作数量
  - `RETURN` 有返回值操作数量
  - `NON_PARAM` 无传入参数操作数量
  - `PARAM` 有传入参数操作数量

  需要返回同时满足所有查询条件的操作个数

  其中，`NON_RETURN`和`RETURN`同时出现或者`NON_PARAM`和`PARAM`同时出现均为错误，为模式冲突错误。
- 关于返回值的问题，是这样定义的：**当且仅当一个`UmlOperation`下所属的`UmlParameter`，全部非`return`时，才算是`NON_RETURN`类型。**（实际上，`void`也算是一种返回值类型，C/C++/Java对于这件事也都是这样的定义）
- 本指令中统计的一律为此类自己定义的操作，不包含继承自其各级父类所定义的操作

类中的属性有多少个

输入指令格式：`CLASS_ATTR_COUNT classname mode`

举例：`CLASS_ATTR_COUNT Elevator SELF_ONLY`

输出：

- `Ok, attribute count of class "classname" is x.` `x`为类中属性的个数
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

说明：

- `mode`表示查询的模式，数据类型为`AttributeQueryType`，取值为：
  - `ALL` 全部属性数量（包括各级父类定义的属性）
  - `SELF_ONLY` 此类自身定义的属性数量

类有几个关联

输入指令格式：`CLASS_ASSO_COUNT classname`

举例：`CLASS_ASSO_COUNT Elevator`

输出：

- `Ok, association count of class "classname" is x.` `x`为类关联的个数
  - 如果出现自关联行为的话，也算在内
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

注意：

- Associate关系需要考虑父类的继承。即，如果X类继承了Y类，那么Y的Association也属于X。

类的关联的对端是哪些类

输入指令格式：`CLASS_ASSO_CLASS_LIST classname`

举例：`CLASS_ASSO_CLASS_LIST Elevator`

输出：

- `Ok, associated classes of class "classname" are (A, B, C).` A、B、C为类所有关联的对端的类名，其中
  - 传出列表时可以乱序，官方接口会自动进行排序（但是**需要编写者自行保证不重不漏**；**特别的，对于同名但id不同的类，如果结果同时包含多个的话，需要在列表中返回对应数量个类名**）
  - 如果出现自关联的话，那么自身类也需要加入输出
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

注意：

- 同上一条，Association关系需要考虑父类的继承。即，假如X类继承了Y类，那么Y的Association对端节点也属于X。

类的操作可见性

输入指令格式：`CLASS_OPERATION_VISIBILITY classname methodname`

举例：`CLASS_OPERATION_VISIBILITY Taxi setStatus`

输出：

- `Ok, operation visibility of method "methodname" in class "classname" is public: xxx, protected: xxx, private: xxx, package-private: xxx.` 该操作的实际可见性统计
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

说明：

- 本指令中统计的一律为此类自己定义的操作，不包含其各级父类所定义的操作
- 在上一条的前提下，需要统计出全部的名为methodname的方法的可见性信息

类的属性可见性

输入指令格式：`CLASS_ATTR_VISIBILITY classname attrname`

举例：`CLASS_ATTR_VISIBILITY Taxi id`

输出：

- `Ok, attribute "attrname" in class "classname"'s visibility is public/protected/private/package-private.` 该属性的实际可见性
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个
- `Failed, attribute not found.` 类中没有该属性
- `Failed, duplicated attribute.` 类中属性存在多个同名

说明：

- 本指令的查询均需要考虑属性的继承关系。
- 其中对于父类和子类均存在此名称的属性时，需要按照`duplicated attribute`处理。

类的顶级父类

输入指令格式：`CLASS_TOP_BASE classname`

举例：`CLASS_TOP_BASE AdvancedTaxi`

输出：

- `Ok, top base class of class "classname" is top_classname.` `top_classname`为顶级父类
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

说明：

- 具体来说，对于类X，如果Y为其顶级父类的话，则满足
  - X是Y的子类（此处特别定义，X也是X的子类）
  - 不存在类Z，使得Y是Z的子类
  - 简单来说，假如把继承关系比作上下级关系的话（被继承者为上级），那么顶级父类就相当于顶头上司

类实现的全部接口

输入指令格式：`CLASS_IMPLEMENT_INTERFACE_LIST classname`

举例：`CLASS_IMPLEMENT_INTERFACE_LIST Taxi`

输出：

- `Ok, implement interfaces of class "classname" are (A, B, C).` A、B、C为继承的各个接口
  - 传出列表时可以乱序，官方接口会自动进行排序（但是需要编写者自行保证不重不漏；**特别的，对于同名但id不同的接口，如果结果同时包含多个的话，需要在列表中返回对应数量个接口名**）
  - 特别值得注意的是，无论是直接实现还是通过父类或者接口继承等方式间接实现，都算做实现了接口
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

类是否违背信息隐藏原则

输入指令格式：`CLASS_INFO_HIDDEN classname`

举例：`CLASS_INFO_HIDDEN Taxi`

输出：

- `Yes, information of class "classname" is hidden.` 满足信息隐藏原则。
- `No, attribute xxx in xxx, xxx in xxx, are not hidden.` 不满足信息隐藏原则。
- `Failed, class "classname" not found.` 类不存在
- `Failed, duplicated class "classname".` 类存在多个

说明：

- 信息隐藏原则，指的是**在类属性的定义中，不允许使用private以外的任何可见性修饰**
- 本指令中需要列出全部的非隐藏属性，同时也需要考虑继承自父类的非隐藏属性
- 值得注意的是，父类和子类中，是可以定义同名属性的（甚至还可以不同类型，不同可见性，感兴趣的话可以自己尝试尝试），然而**父类中定义的和子类中定义的实际上并不是同一个属性，需要在输出时进行分别处理**
- 同样的，返回的列表可以乱序，官方接口会进行自动排序（但是依然需要编写者保证不重不漏）

样例

样例一

输入文本（模型部分导出自`mdj`文件：传送门，导出方法见官方包开源文档）

```
{"_parent":"AAAAAAFq3tvYM76UevI=","visibility":"public","name":"Key","_type":"UMLClass","_id":"AAAAAAFq7weIMSb5xqQ="}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","visibility":"public","name":"equals","_type":"UMLOperation","_id":"AAAAAAFq7weIMSb8qxc="}
{"_parent":"AAAAAAFq7weIMSb8qxc=","name":"o","_type":"UMLParameter","_id":"AAAAAAFq7weIMSb9G0k=","type":"Object","direction":"in"}
{"_parent":"AAAAAAFq7weIMSb8qxc=","name":null,"_type":"UMLParameter","_id":"AAAAAAFq7weIMSb+Au4=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","visibility":"public","name":"getMatchedLockId","_type":"UMLOperation","_id":"AAAAAAFq7weIMSb\/6gM="}
{"_parent":"AAAAAAFq7weIMSb\/6gM=","name":null,"_type":"UMLParameter","_id":"AAAAAAFq7weIMScAoOk=","type":"int","direction":"return"}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","visibility":"public","name":"Operation1","_type":"UMLOperation","_id":"AAAAAAFq7w1zLCePJrI="}
{"_parent":"AAAAAAFq7w1zLCePJrI=","name":"Parameter1","_type":"UMLParameter","_id":"AAAAAAFq7w2dZCeV4K8=","type":
{"$ref":"AAAAAAFq7weIMSb5xqQ="},"direction":"return"}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","name":null,"_type":"UMLGeneralization","_id":"AAAAAAFq7wfNvyd+GgY=","source":"AAAAAAFq7weIMSb5xqQ=","target":"AAAAAAFq7weqoCcQE7I="}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","visibility":"private","name":"keyID","_type":"UMLAttribute","_id":"AAAAAAFq7weIMSb6+v8=","type":"int"}
{"_parent":"AAAAAAFq7weIMSb5xqQ=","visibility":"private","name":"matchedLockID","_type":"UMLAttribute","_id":"AAAAAAFq7weIMSb7oPM=","type":"int"}
{"_parent":"AAAAAAFq3tvYM76UevI=","visibility":"public","name":"ElcKey","_type":"UMLClass","_id":"AAAAAAFq7weqoCcQE7I="}
{"_parent":"AAAAAAFq7weqoCcQE7I=","visibility":"public","name":"equals","_type":"UMLOperation","_id":"AAAAAAFq7weqoCcTngY="}
{"_parent":"AAAAAAFq7weqoCcTngY=","name":"o","_type":"UMLParameter","_id":"AAAAAAFq7weqoCcUI6g=","type":"Object","direction":"in"}
{"_parent":"AAAAAAFq7weqoCcTngY=","name":null,"_type":"UMLParameter","_id":"AAAAAAFq7weqoCcVxI0=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFq7weqoCcQE7I=","visibility":"private","name":"sigCode","_type":"UMLAttribute","_id":"AAAAAAFq7weqoCcSu1Q=","type":"long"}
END_OF_MODEL
```

```
CLASS_COUNT
CLASS_TOP_BASE Key
```

输出文本

```
Total class count is 2.
Ok, top base class of class "Key" is ElcKey.
```

样例二

输入文本

```
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"ClassBotto
m","_type":"UMLClass","_id":"AAAAAAFrAa3mz8zhDcw="}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"ClassBotto
m","_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzij74="}
{"_parent":"AAAAAAFrAa3mOMzij74=","name":"privateValue","_type":"UMLParame
ter","_id":"AAAAAAFrAa3mOMzj+8k=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3mOMzij74=","name":"protectedValue","_type":"UMLPara
meter","_id":"AAAAAAFrAa3mOMzkQFE=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3mOMzij74=","name":"privateValue1","_type":"UMLParam
eter","_id":"AAAAAAFrAa3mOMzlV8E=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3mOMzij74=","name":"publicValue","_type":"UMLParamet
er","_id":"AAAAAAFrAa3mOMzmubE=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3mOMzij74=","name":"packagePrivateValue","_type":"UM
LParameter","_id":"AAAAAAFrAa3mOMznmOg=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"getPrivate
Value","_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzoDII="}
{"_parent":"AAAAAAFrAa3mOMzoDII=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMzpSeg=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"getPublicV
alue","_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzq4kk="}
{"_parent":"AAAAAAFrAa3mOMzq4kk=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMzrPvM=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"getPackage
PrivateValue","_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzstPA="}
{"_parent":"AAAAAAFrAa3mOMzstPA=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMztY04=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"equals","_
type":"UMLOperation","_id":"AAAAAAFrAa3mOMzuZvA="}
{"_parent":"AAAAAAFrAa3mOMzuZvA=","name":"o","_type":"UMLParameter","_id":
"AAAAAAFrAa3mOMzvonO=","type":
{"$ref":"AAAAAAFrAa3m4MOmD5A="},"direction":"in"}
{"_parent":"AAAAAAFrAa3mOMzuZvA=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMzwL\/A=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"hashCode",
"_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzx+nE="}
{"_parent":"AAAAAAFrAa3mOMzx+nE=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMzy1R8=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"toString",
"_type":"UMLOperation","_id":"AAAAAAFrAa3mOMzzldE="}
{"_parent":"AAAAAAFrAa3mOMzzldE=","name":null,"_type":"UMLParameter","_id"
:"AAAAAAFrAa3mOMzOuNc=","type":"String","direction":"return"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","name":null,"_type":"UMLGeneralization",
"_id":"AAAAAAFrAa3m38OcADc=","source":"AAAAAAFrAa3mz8zhDcw=","target":"AAA
AAAFrAa3m2Mz1OFQ="}
```

```
{"_parent":"AAAAAAFrAa3mz8zhDcw=","name":"AssociationToInterfaceMiddle","_type":"UMLAssociation","end2":"AAAAAAFrAbAPXtAuVoM=","end1":"AAAAAAFrAbAPXtAtTlY=","_id":"AAAAAAFrAbAPXtAs+dQ="}
{"reference":"AAAAAAFrAa3m3c0SkUU=","multiplicity":"","_parent":"AAAAAAFrAbAPXtAs+dQ=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbAPXtAuVoM="}
{"reference":"AAAAAAFrAa3mz8zhDcw=","multiplicity":"","_parent":"AAAAAAFrAbAPXtAs+dQ=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbAPXtAtTlY="}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"private","name":"privateValue","_type":"UMLAttribute","_id":"AAAAAAFrAa3m4M0hJeg=","type":"int"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"public","name":"publicValue","_type":"UMLAttribute","_id":"AAAAAAFrAa3m4M0iFac=","type":"int"}
{"_parent":"AAAAAAFrAa3mz8zhDcw=","visibility":"package","name":"packagePrivateValue","_type":"UMLAttribute","_id":"AAAAAAFrAa3m4M0jzDs=","type":"int"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"ClassMiddle","_type":"UMLClass","_id":"AAAAAAFrAa3m2Mz10FQ="}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","visibility":"public","name":"ClassMiddle","_type":"UMLOperation","_id":"AAAAAAFrAa3m2Mz2Nhg="}
{"_parent":"AAAAAAFrAa3m2Mz2Nhg=","name":"privateValue","_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz3u5o=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3m2Mz2Nhg=","name":"protectedValue","_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz4YbU=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","visibility":"public","name":"getProtectedValue","_type":"UMLOperation","_id":"AAAAAAFrAa3m2Mz531E="}
{"_parent":"AAAAAAFrAa3m2Mz531E=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz6QcU=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","visibility":"public","name":"equals","_type":"UMLOperation","_id":"AAAAAAFrAa3m2Mz7AaI="}
{"_parent":"AAAAAAFrAa3m2Mz7AaI=","name":"o","_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz8ivw=","type":
{"$ref":"AAAAAAFrAa3m4M0mD5A="},"direction":"in"}
{"_parent":"AAAAAAFrAa3m2Mz7AaI=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz91b8=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","visibility":"public","name":"hashCode","_type":"UMLOperation","_id":"AAAAAAFrAa3m2Mz+Rr8="}
{"_parent":"AAAAAAFrAa3m2Mz+Rr8=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m2Mz\/dQA=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","visibility":"public","name":"toString","_type":"UMLOperation","_id":"AAAAAAFrAa3m2M0ADws="}
{"_parent":"AAAAAAFrAa3m2M0ADws=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m2M0BWDY=","type":"String","direction":"return"}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","name":null,"_type":"UMLGeneralization","_id":"AAAAAAFrAa3m380dOcI=","source":"AAAAAAFrAa3m2Mz10FQ=","target":"AAAAAAFrAa3m280C+z8="}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","name":null,"_type":"UMLInterfaceRealization","_id":"AAAAAAFrAa3m380ffw8=","source":"AAAAAAFrAa3m2Mz10FQ=","target":"AAAAAAFrAa3m3c0SkUU="}
{"_parent":"AAAAAAFrAa3m2Mz10FQ=","name":"AssocationToMySelf","_type":"UMLAssociation","end2":"AAAAAAFrAa+izs+dvj4=","end1":"AAAAAAFrAa+izs+cfVk=","_id":"AAAAAAFrAa+izc+b8uY="}
{"reference":"AAAAAAFrAa3m2Mz10FQ=","multiplicity":"","_parent":"AAAAAAFrAa+izc+b8uY=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAa+izs+dvj4="}
{"reference":"AAAAAAFrAa3m2Mz10FQ=","multiplicity":"","_parent":"AAAAAAFrAa+izc+b8uY=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAa+izs+cfVk="}
```

```
{"_parent":"AAAAAAFrAa3m2Mz1OFQ=","name":"AssociationToInterfaceMiddle","_type":"UMLAssociation","end2":"AAAAAAFrAbCIDdD75KU=","end1":"AAAAAAFrAbCIDdD68kA=","_id":"AAAAAAFrAbCIDdD5gLA="}
{"reference":"AAAAAAFrAa3m3cOSkUU=","multiplicity":"","_parent":"AAAAAAFrAbCIDdD5gLA=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbCIDdD75KU="}
{"reference":"AAAAAAFrAa3m2Mz1OFQ=","multiplicity":"","_parent":"AAAAAAFrAbCIDdD5gLA=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbCIDdD68kA="}
{"_parent":"AAAAAAFrAa3m2Mz1OFQ=","name":"AssociationBetweenParantAndSon","_type":"UMLAssociation","end2":"AAAAAAFrAbG3dNVDDOM=","end1":"AAAAAAFrAbG3dNVC6fs=","_id":"AAAAAAFrAbG3dNVBVBE="}
{"reference":"AAAAAAFrAa3mz8zhDcw=","multiplicity":"","_parent":"AAAAAAFrAbG3dNVBVBE=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbG3dNVDDOM="}
{"reference":"AAAAAAFrAa3m2Mz1OFQ=","multiplicity":"","_parent":"AAAAAAFrAbG3dNVBVBE=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbG3dNVC6fs="}
{"_parent":"AAAAAAFrAa3m2Mz1OFQ=","visibility":"protected","name":"protectedValue","_type":"UMLAttribute","_id":"AAAAAAFrAa3m4MOk1oU=","type":"int"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"ClassTop","_type":"UMLClass","_id":"AAAAAAFrAa3m28OC+z8="}
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"public","name":"ClassTop","_type":"UMLOperation","_id":"AAAAAAFrAa3m28ODrH8="}
{"_parent":"AAAAAAFrAa3m28ODrH8=","name":"privateValue","_type":"UMLParameter","_id":"AAAAAAFrAa3m28OEImU=","type":"int","direction":"in"}
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"public","name":"getPrivateValue","_type":"UMLOperation","_id":"AAAAAAFrAa3m28OFikk="}
{"_parent":"AAAAAAFrAa3m28OFikk=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m28OGVpc=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"public","name":"equals","_type":"UMLOperation","_id":"AAAAAAFrAa3m28OHW28="}
{"_parent":"AAAAAAFrAa3m28OHW28=","name":"o","_type":"UMLParameter","_id":"AAAAAAFrAa3m28OIXyg=","type":
{"$ref":"AAAAAAFrAa3m4MOmD5A="},"direction":"in"}
{"_parent":"AAAAAAFrAa3m28OHW28=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m28OJVuY=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"public","name":"hashCode","_type":"UMLOperation","_id":"AAAAAAFrAa3m28OKd1k="}
{"_parent":"AAAAAAFrAa3m28OKd1k=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m28OLzRM=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"public","name":"toString","_type":"UMLOperation","_id":"AAAAAAFrAa3m28OMa4c="}
{"_parent":"AAAAAAFrAa3m28OMa4c=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m28ONYMo=","type":"String","direction":"return"}
{"_parent":"AAAAAAFrAa3m28OC+z8=","name":null,"_type":"UMLInterfaceRealization","_id":"AAAAAAFrAa3m38OgF9E=","source":"AAAAAAFrAa3m28OC+z8=","target":"AAAAAAFrAa3m3s0Vct4="}
{"_parent":"AAAAAAFrAa3m28OC+z8=","name":"AssociationToInterfaceMiddle","_type":"UMLAssociation","end2":"AAAAAAFrAbDsHtJAF54=","end1":"AAAAAAFrAbDsHtI\/xkO=","_id":"AAAAAAFrAbDsHtI+x+U="}
{"reference":"AAAAAAFrAa3m3cOSkUU=","multiplicity":"","_parent":"AAAAAAFrAbDsHtI+x+U=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbDsHtJAF54="}
{"reference":"AAAAAAFrAa3m28OC+z8=","multiplicity":"","_parent":"AAAAAAFrAbDsHtI+x+U=","visibility":"public","name":null,"_type":"UMLAssociationEnd","aggregation":"none","_id":"AAAAAAFrAbDsHtI\/xkO="}
```

```
{"_parent":"AAAAAAFrAa3m28OC+z8=","visibility":"private","name":"privateValue","_type":"UMLAttribute","_id":"AAAAAAFrAa3m4MOlIdk=","type":"int"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"InterfaceAnother","_type":"UMLInterface","_id":"AAAAAAFrAa3m3MOO4wM="}
{"_parent":"AAAAAAFrAa3m3MOO4wM=","visibility":"package","name":"eqauls","_type":"UMLOperation","_id":"AAAAAAFrAa3m3MOPGpA="}
{"_parent":"AAAAAAFrAa3m3MOPGpA=","name":"o","_type":"UMLParameter","_id":"AAAAAAFrAa3m3MOQMPY=","type":{"$ref":"AAAAAAFrAa3m4MOmD5A="},"direction":"in"}
{"_parent":"AAAAAAFrAa3m3MOPGpA=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m3MOR6P8=","type":"boolean","direction":"return"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"InterfaceMiddle","_type":"UMLInterface","_id":"AAAAAAFrAa3m3cOSkUU="}
{"_parent":"AAAAAAFrAa3m3cOSkUU=","visibility":"package","name":"hashCode","_type":"UMLOperation","_id":"AAAAAAFrAa3m3cOTOSI="}
{"_parent":"AAAAAAFrAa3m3cOTOSI=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m3cOUZM8=","type":"int","direction":"return"}
{"_parent":"AAAAAAFrAa3m3cOSkUU=","name":null,"_type":"UMLGeneralization","_id":"AAAAAAFrAa3m38OekAk=","source":"AAAAAAFrAa3m3cOSkUU=","target":"AAAAAAFrAa3m3MOO4wM="}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"InterfaceTop","_type":"UMLInterface","_id":"AAAAAAFrAa3m3sOVct4="}
{"_parent":"AAAAAAFrAa3m3sOVct4=","visibility":"package","name":"toString","_type":"UMLOperation","_id":"AAAAAAFrAa3m3sOWQHc="}
{"_parent":"AAAAAAFrAa3m3sOWQHc=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m3sOXx+w=","type":"String","direction":"return"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"Main","_type":"UMLClass","_id":"AAAAAAFrAa3m3sOYiFQ="}
{"_parent":"AAAAAAFrAa3m3sOYiFQ=","visibility":"public","name":"main","_type":"UMLOperation","_id":"AAAAAAFrAa3m3sOZXms="}
{"_parent":"AAAAAAFrAa3m3sOZXms=","name":"args","_type":"UMLParameter","_id":"AAAAAAFrAa3m3sOaG9s=","type":"String","direction":"in"}
{"_parent":"AAAAAAFrAa3m3sOZXms=","name":null,"_type":"UMLParameter","_id":"AAAAAAFrAa3m3sObZS8=","type":"void","direction":"return"}
{"_parent":"AAAAAAFrAa3mxMzgjSo=","visibility":"public","name":"Object","_type":"UMLClass","_id":"AAAAAAFrAa3m4MOmD5A="}
END_OF_MODEL
CLASS_COUNT
CLASS_OPERATION_COUNT ClassBottom
CLASS_OPERATION_COUNT ClassBottem NON_PARAM
CLASS_OPERATION_COUNT ClassBottom NON_PARAM NON_RETURN
CLASS_OPERATION_COUNT ClassBottom PARAM
CLASS_OPERATION_COUNT ClassBottom NON_RETURN
CLASS_OPERATION_COUNT ClassBottom RETURN
CLASS_ATTR_COUNT ClassBottom ALL
CLASS_ATTR_COUNT ClassBottom SELF_ONLY
CLASS_ASSO_COUNT ClassTop
CLASS_ASSO_COUNT ClassMiddle
CLASS_ASSO_COUNT ClassBottom
CLASS_ASSO_CLASS_LIST ClassTop
CLASS_ASSO_CLASS_LIST ClassMiddle
CLASS_ASSO_CLASS_LIST ClassBottom
CLASS_OPERATION_VISIBILITY ClassBottom hashCode
CLASS_ATTR_VISIBILITY ClassBottom privateValue
CLASS_ATTR_VISIBILITY ClassBottom protectedValue
CLASS_ATTR_VISIBILITY ClassBottom publicValue
CLASS_ATTR_VISIBILITY ClassBottom packagePrivateValue
CLASS_TOP_BASE ClassBottom
```

```
CLASS_TOP_BASE ClassMiddle
CLASS_IMPLEMENT_INTERFACE_LIST ClassBottom
CLASS_IMPLEMENT_INTERFACE_LIST ClassMiddle
CLASS_IMPLEMENT_INTERFACE_LIST ClassTop
CLASS_INFO_HIDDEN ClassBottom
CLASS_INFO_HIDDEN ClassMiddle
CLASS_INFO_HIDDEN ClassTop
```

输出文本

```
Total class count is 5.
Ok, operation count of class "ClassBottom" is 7.
Failed, class "ClassBottem" not found
Ok, operation count of class "ClassBottom" is 0.
Ok, operation count of class "ClassBottom" is 2.
Ok, operation count of class "ClassBottom" is 1.
Ok, operation count of class "ClassBottom" is 6.
Ok, attribute count of class "ClassBottom" is 5.
Ok, attribute count of class "ClassBottom" is 3.
Ok, association count of class "ClassTop" is 1.
Ok, association count of class "ClassMiddle" is 5.
Ok, association count of class "ClassBottom" is 7.
Ok, associated classes of class "ClassTop" are ().
Ok, associated classes of class "ClassMiddle" are (ClassBottom,
ClassMiddle).
Ok, associated classes of class "ClassBottom" are (ClassBottom,
ClassMiddle).
Ok, operation visibility of method "hashCode" in class "ClassBottom" is
public: 1, protected: 0, private: 0, package-private: 0.
Failed, duplicated attribute "privateValue" in class "ClassBottom".
Ok, attribute "protectedValue" in class "ClassBottom"'s visibility is
protected.
Ok, attribute "publicValue" in class "ClassBottom"'s visibility is public.
Ok, attribute "packagePrivateValue" in class "ClassBottom"'s visibility is
package-private.
Ok, top base class of class "ClassBottom" is ClassTop.
Ok, top base class of class "ClassMiddle" is ClassTop.
Ok, implement interfaces of class "ClassBottom" are (InterfaceAnother,
InterfaceMiddle, InterfaceTop).
Ok, implement interfaces of class "ClassMiddle" are (InterfaceAnother,
InterfaceMiddle, InterfaceTop).
Ok, implement interfaces of class "ClassTop" are (InterfaceTop).
No, attribute packagePrivateValue in ClassBottom, publicValue in
ClassBottom, protectedValue in ClassMiddle, are not hidden.
No, attribute protectedValue in ClassMiddle, are not hidden.
Yes, information of class "ClassTop" is hidden.
```

判定

公测（包括弱测、中测与强测）数据基本限制

- `mdj`文件内容限制
  - 包含且仅包含类图，并在`UMLModel`内进行建模，且每个`UMLModel`内的元素不会引用当前`UMLModel`以外的元素（即关系是一个闭包）
  - 原始mdj文件符合`starUML`规范，可在`starUML`中正常打开和显示

    - `mdj`文件中最多只包含`300`个元素
    - **此外为了方便本次的情况处理，保证所建模的类图模型，均可以在Oracle Java 8中正常实现出来**
- 输入指令限制
    - 最多不超过200条指令
    - 输入指令满足标准格式
- 测试数据限制
    - 所有公测数据不会对接口中定义的属性，类属性(static attribute)，类方法(static method)做任何测试要求，本次作业不需要对这些情况进行考虑。

测试模式

公测均通过标准输出输出进行。

指令将会通过查询UML类图各种信息的正确性，从而测试UML解析器各个接口的实现正确性。

对于任何满足基本数据限制的输入，程序都应该保证不会异常退出，如果出现问题则视为未通过该测试点。

程序的最大运行cpu时间为`2s`，保证强测数据有一定梯度。

提示&说明

- 本次作业中可以自行组织工程结构。任意新增`java`代码文件。只需要保证`UmlInteraction`类的继承与实现即可。
- **关于本次作业解析器类的设计具体细节，本指导书中均不会进行过多描述，请自行去官方包开源仓库中查看接口的规格，并依据规格进行功能的具体实现**，必要时也可以查看AppRunner的代码实现。关于官方包的使用方法，可以去查看开源库的`README.md`。
- 开源库地址：https://gitlab.buaaoo.top/oo_2020_public/oo_2020-unit_4-homework_1-opensource
- 推荐各位同学在课下测试时使用Junit单元测试来对自己的程序进行测试
    - Junit是一个单元测试包，**可以通过编写单元测试类和方法，来实现对类和方法实现正确性的快速检查和测试。**还可以查看测试覆盖率以及具体覆盖范围（精确到语句级别），以帮助编程者全面无死角的进行程序功能测试。
    - Junit已在评测机中部署（版本为Junit4.12，一般情况下确保为Junit4即可），所以项目中可以直接包含单元测试类，在评测机上不会有编译问题。
    - 此外，Junit对主流Java IDE（Idea、eclipse等）均有较为完善的支持，可以自行安装相关插件。推荐两篇博客：
        - Idea下配置Junit
        - Idea下Junit的简单使用
    - 感兴趣的同学可以自行进行更深入的探索，百度关键字：`Java Junit`。
- 强烈推荐同学们
    - 去阅读本次的源代码
    - **好好复习下本次的ppt，并理清楚各个`UmlElement`数据模型的结构与关系。**
- **不要试图通过反射机制来对官方接口进行操作**，我们有办法进行筛查。此外，如果发现有人试图通过反射等手段hack输出接口的话，请邮件login@buaa.edu.cn或使用微信向助教进行举报，**经核实后，将直接作为无效作业处理。**