说明文档

作业要求：利用任意代码生成技术实现一个代码生成器，能够读取任意 Aquila 模型，并将其中的类图转换成代码框架。

本程序实现了将任意一个 Aquila 类图模型（输入）转化成 C++代码的功能。

```
41824179-王丹琳
├── BaseClassTemplate.tpl 类模板
├── generate.py           主体代码
├── OOA.aquila            Aquila 模型
├── person.cpp             所生成的类
├── 督导.cpp               所生成的类
├── 教师.cpp               所生成的类
├── 教务.cpp               所生成的类
├── 课程.cpp               所生成的类
├── 课程活动.cpp           所生成的类
├── 课堂.cpp               所生成的类
├── 评语.cpp               所生成的类
├── 学生.cpp               所生成的类
└── 用户.cpp               所生成的类
```

1. 代码简要说明

主体代码 generate.py 主要分为两部分：Aquila 模型的解析与模板渲染。

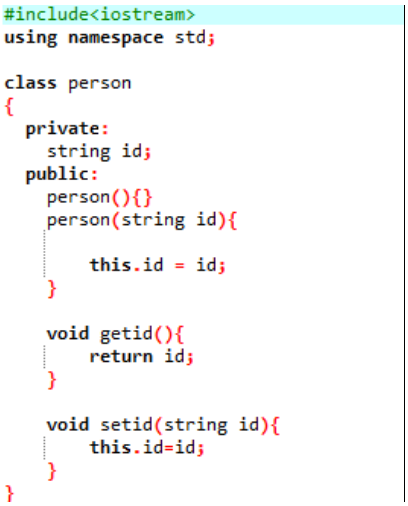
Aquila 模型的解析使用了 xml.etree.ElementTree 模块，它可以用于处理任何树结构的数据，而 Aquila 类图模型其实也是一个树结构模型。通过 findall()可以查找当前元素的直接子元素中带有指定标签的元素，get()用来访问元素的属性，这样就可以把 Aquila 模型中需要的部分抽提出来。

对于模板渲染，首先就是模板的书写，本来想使用 Template 及 safe_substitute()来实现，但是除了渲染变量，无法实现选择判断，循环；于是选用了专用的模板渲染引擎 Jinja2, 它除渲染变量外还支持更丰富的功能，如 if 判断和 for 循环遍历, 以及过滤器等。

2. 功能实现简要说明

对于继承，均默认公有继承；类所拥有的属性均默认为私有属性；类所拥有的行为均默认为公有。

设置了默认构造函数，属性 get，set 函数方法。



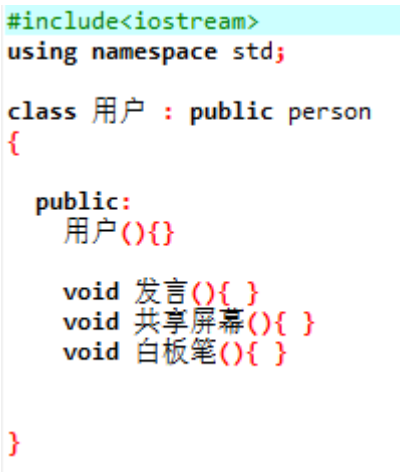
```cpp
#include<iostream>
using namespace std;

class person
{
  private:
    string id;
  public:
    person(){}
    person(string id){
        this.id = id;
    }

    void getid(){
        return id;
    }

    void setid(string id){
        this.id=id;
    }
}
```

person 类图所生成的 person.cpp



```cpp
#include<iostream>
using namespace std;

class 用户 : public person
{

  public:
    用户(){}

    void 发言(){ }
    void 共享屏幕(){ }
    void 白板笔(){ }


}
```

用户类图所生成的用户.cpp

对于代码的测试，我选用了自己第二次的作业与同学的第二次的作业进行测试，发现均正常转换为 C++代码。