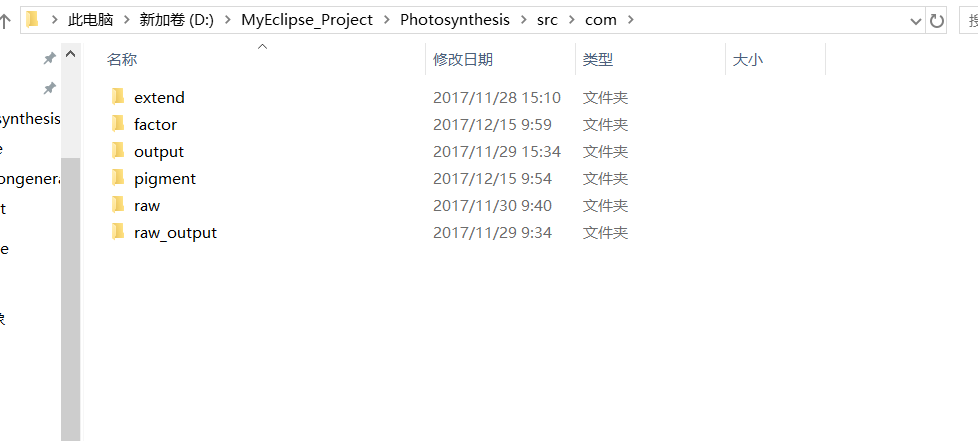
**Github网址：https://github.com/YiFraternity/QuestionGeneration**

**一、我将题目分成了五类：原料、产物、原料和产物、影响因素以及色素。**



（0）“extend”这个包，包含了一些被用来继承的类。

（1）“raw”这一个包，是进行对有关原料和生物过程进行提问的。

（2）“output”这一个包，是进行对有关产物和生物过程进行提问的。

（3）“raw output”这个包是对全部进行提问的。

（4）“factor”这个包，是进行对有关光合作用条件进行提问的

（5）“pigment”这个包，是进行对光合作用有影响的色素进行提问的。

**二、extend包**

extend这个包是用来继承的，里面写了一些父类。



1. SPARQLParse.java这个类是使用Jena包对owl文件进行处理的，用来解析owl文件。
2. GetMaterial.java这个类是用根据SPARQLParse.java解析的owl文件，将解析得到的内容写入数组里面，方便以后调用使用。
3. OperateFile.java这个类是用来对文件进行处理的，写文件，清空文件，读取文件，等等。
4. Struct.java这个类主要是用来子过程使用的原料和产物，返回到父过程时需要对子过程进行封装用的。

**三、raw\_output包**

* 1. **raw\_output这个包里又有一些包，每个包对应一道题目。**



1. function这个包是用来对用途进行提问的，问：生物物质在生物过程中的作用
2. relation这个包是用来对两个生物过程进行提问的，问：生物过程A如何和生物过程B建立联系。
3. whats这个包是用来对原料和产物进行提问的，问生物过程A中的原料有哪些，产物有哪些？
   1. **function包：生物物质的作用**



1. GetRawOutputFunctionForNeed.java这个类是用来产生对用途提问的关键词。
2. GetProblemForFunction.java是用来提出问题和给出答案的类
3. GetRawOutputFunctionForNeed.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找原料和产物。

③将子过程的原料和产物作为其父过程的原料和产物进行返回

④将原料和产物相同的生物物质去掉。

⑤将得到生物物质写入文件

1. GetProblemForFunction.java算法

①写一个生成问题的模板，解析模板。

②读取GetRawOutputFunctionForNeed.java得到的原料和产物。

③按照模板生成题目。

* 1. **relation包：生物物质之间的关系**



1. GetRawOutputForRelationNeed.java使用生成所需要的关键字的类
2. GetRawOutputForRelationNeed.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找原料和产物。

③将子过程的原料和产物作为其父过程的原料和产物进行返回

④记录原料和产物相同的生物物质。

⑤将得到的相同的生物物质写入文件。

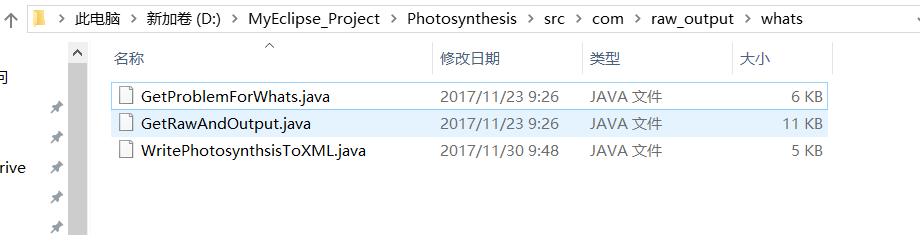
1. GetProblemForRelation.java使用GetRawOutputForRelation.java得到的关键字生成题目。
2. GetProblemForRelation.java算法：

①写一个生成问题的模板，解析模板。

②读取GetProblemForRelation.java得到的原料和产物。

③按照模板生成题目。

* 1. **whats包：生成生物过程使用的原料和产生的产物**



1. GetRawAndOutput.java，获得所需要的生物物质
2. GetRawAndOutput.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找原料和产物。

③将子过程的原料和产物作为其父过程的原料和产物进行返回。

④去掉相同的原料和产物。

⑤将得到的相同的生物物质写入文件。

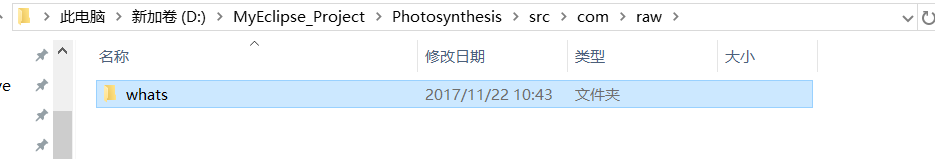
1. GetProblemForWhats.java，得到问题和答案。
2. GetProblemForWhats.java算法

①写一个生成问题的模板，解析模板。

②读取GetRawAndOutput.java得到的原料和产物。

③按照模板生成题目。

**四、raw包**



1. **raw包下有一个题目whats**

whats这个包是用来对原料进行提问的，问生物过程A中的原料有哪些

1. **whats包：生成生物过程使用的原料**



1. GetRawFromBioprocess.java，获得所需要的生物物质
2. GetRawFromBioprocess.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找原料和产物。

③将子过程的原料和产物作为其父过程的原料和产物进行返回

④去掉相同的原料和产物。

⑤将得到的原料和生物过程写入文件。

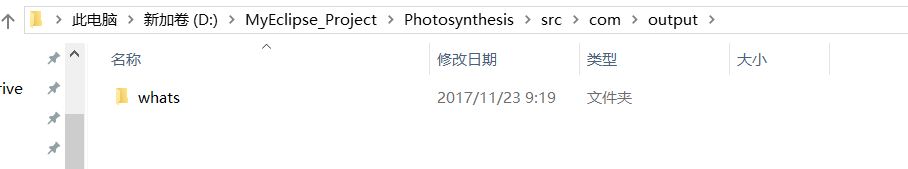
1. GetProblemForWhats.java，得到问题和答案。
2. GetProblemForWhats.java算法

①写一个生成问题的模板，解析模板。

②读取GetRawFromBioprocess.java得到的原料和生物过程。

③按照模板生成题目。

**五、output包**



1. **output包下有一个题目whats**

whats这个包是用来对原料进行提问的，问生物过程A中的生成的产物有哪些

1. **whats包：生物过程生成的产物**



1. GetOutputFromBioprocess.java，获得所需要的生物物质
2. GetOutputFromBioprocess.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找原料和产物。

③将子过程的原料和产物作为其父过程的原料和产物进行返回

④去掉相同的原料和产物。

⑤将得到的产物和生物过程写入文件。

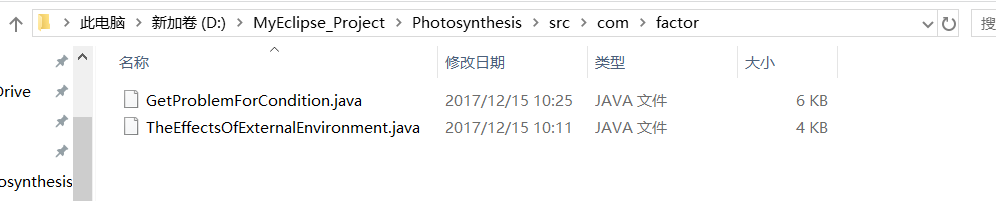
1. GetProblemForWhats.java，得到问题和答案。
2. GetProblemForWhats.java算法

①写一个生成问题的模板，解析模板。

②读取GetOutputFromBioprocess.java得到的生物过程和产物。

③按照模板生成题目

**六、factor包**



**factor包是查找影响光合作用的因素**

1. TheEffectsOfExternalEnvironment.java，获得所需要的生物物质
2. TheEffectsOfExternalEnvironment.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找条件。

③把生物过程及条件返回。

④将得到的条件和生物过程写入文件。

1. GetProblemForCondition.java，得到问题和答案。
2. GetProblemForCondition.java算法

①写一个生成问题的模板，解析模板。

②读取TheEffectsOfExternalEnvironment.java得到的生物过程和条件。

③按照模板生成题目

**七、pigment包：色素对光合作用的影响**

光合色素里面包含的一些色素对光合作用的影响



1. GetPigment.java，获得所需要的生物物质
2. GetPigment.java算法：

①从光合作用开始进行递归进行查找其所有子过程。

②对所有子过程，查找条件。

③对查找到的条件进行查找包含的资料。

④将找到的生物过程、条件以及包含的资料返回写入文本

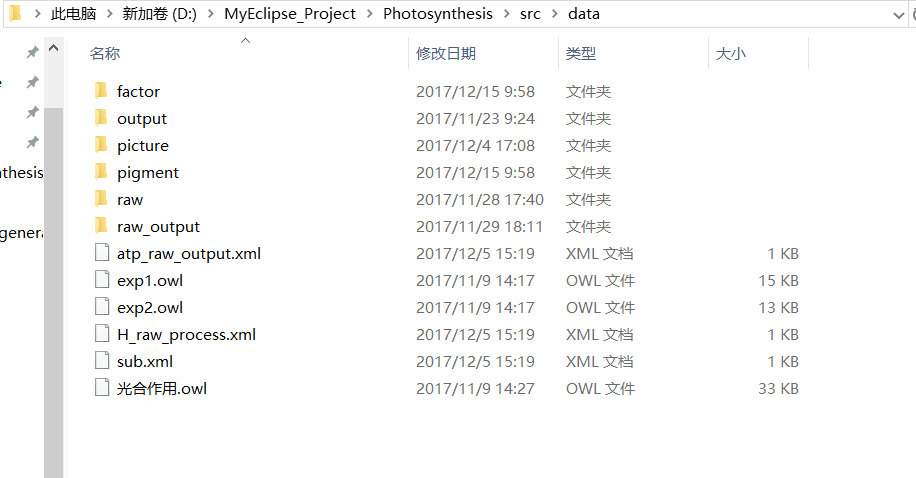
1. GetProblemForPigment.java，得到问题和答案。
2. GetProblemForPigment.java算法

①写一个生成问题的模板，解析模板。

②读取GetPigment.java得到的生物过程、条件和资料。

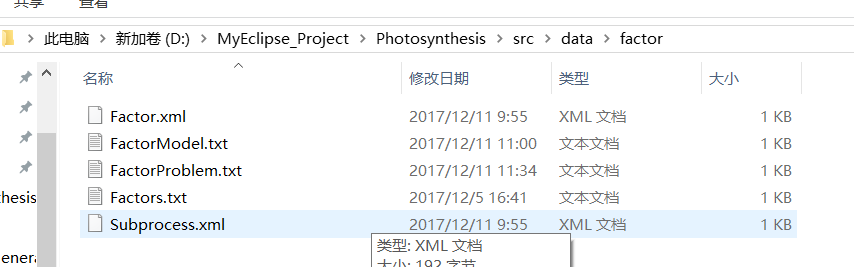
③按照模板生成题目

**八、data这个文件夹是存放数据的文件夹**



1.每个文件夹都存放的是以自己名字为包所调用的文件。

例如：factor存放的是com.factor包里Java文件使用和生成存放的文件夹。



2.data文件夹也存放一些根文件，例如:光合作用.owl文件。