**Output模块使用方法**

//创建一个PaperGenerator实例

PaperGenerator pg = **new** PaperGenerator();

//设置试卷头部信息

pg.setHeader(getHeader());

//为试卷准备所需的所有题型的列表

List<BaseModel> questions = **new** ArrayList<BaseModel>();

questions.add(getSingleChoice());

questions.add(getBlank());

questions.add(getJudgement());

questions.add(getShortAnswer());

questions.add(getComprehensive());

//将所有题型的列表传给PaperGenerator 实例

pg.setQuestions(questions);

//调用PaperGenerator实例的init方法和output方法

pg.init();

pg.output();

**下面演示创建头部信息(上面的getHeader()方法)**

**private** Header getHeader(){

HeaderData hd = **new** HeaderData();

hd.setStartTime(2017);

hd.setEndTime(2018);

hd.setExamTerm(1);

hd.setExamSubject("数据库");

hd.setExamType("开卷");

hd.setExamTime(120);

Header header = **new** Header();

header.setHeaderData(hd);

**return** header;

}

**下面演示如何创建各种题型的题目（即上面的getXxx()方法）**

1)循环创建15道选择题

//返回一个SingleChoice实例，其中singleChoice对象中就包含了singleChoices这样一个List

**private** SingleChoice getSingleChoice(){

List<SingleChoiceData> singleChoices = **new** ArrayList<SingleChoiceData>();

**for**(**int** i=0;i<15;i++){

SingleChoiceData scd = **new** SingleChoiceData();

scd.setTitle("我是选择题");

scd.setaOption("我是选项a");

scd.setbOption("我是选项b");

scd.setcOption("我是选项c");

scd.setdOption("我是选项d");

singleChoices.add(scd);

}

SingleChoice singleChoice = **new** SingleChoice();

singleChoice.setSingleChoices(singleChoices);

**return** singleChoice;

}

2）循环创建5道填空题

//返回一个blank实例，其中blank对象中就包含了blanks这样一个List

**private** Blank getBlank(){

List<BlankData> blanks = **new** ArrayList<BlankData>();

**for**(**int** i=0;i<5;i++){

BlankData bd = **new** BlankData();

bd.setTitle("我是填空题，请填空\_\_\_\_\_\_");

blanks.add(bd);

}

Blank blank = **new** Blank();

blank.setBlanks(blanks);

**return** blank;

}

3）循环创建5道判断题

**private** Judgement getJudgement(){

List<JudgementData> judgements = **new** ArrayList<JudgementData>();

**for**(**int** i=0;i<5;i++){

JudgementData jd = **new** JudgementData();

jd.setTitle("我是判断题,请打√或者×( )");

judgements.add(jd);

}

Judgement judgement = **new** Judgement();

judgement.setJudgements(judgements);

**return** judgement;

}

4）循环创建5道简答题

**private** ShortAnswer getShortAnswer(){

List<ShortAnswerData> shortAnswers = **new** ArrayList<ShortAnswerData>();

**for**(**int** i=0;i<5;i++){

ShortAnswerData sad = **new** ShortAnswerData();

sad.setTitle("我是简答题,请回答");

shortAnswers.add(sad);

}

ShortAnswer shortAnswer = **new** ShortAnswer();

shortAnswer.setShortAnswers(shortAnswers);

**return** shortAnswer;

}

5）循环创建3道综合题

**private** Comprehensive getComprehensive(){

List<ComprehensiveData> comprehensives = **new** ArrayList<ComprehensiveData>();

**for**(**int** i=0;i<3;i++){

ComprehensiveData cd = **new** ComprehensiveData();

cd.setTitle("我是综述题，sdadadsada!!!@#~~~#@!#@!，请回答。");

comprehensives.add(cd);

}

Comprehensive comprehensive = **new** Comprehensive();

comprehensive.setComprehensives(comprehensives);

**return** comprehensive;

}

**最后附上各包下的java文件功能的解释：**

**cn.hengxin.paper.data**包下的**XxxData.java**:

用于为对应的XxxData\_template模板渲染数据

**cn.hengxin.paper.model**包下的**BaseModel**类作为其他类的父类，存放各种题型公有的属性，如出卷时的题号num 和配置信息config，而其他类则为普通的JavaBean，每种题型自带有一个output方法用于输出该题型的题目

**cn.hengxin.paper.utils**包下的**XxxUtils**类为一些通用模块，如**OutputFormatUtils**类将数据按想要的格式（中文）输出，**POIMergeDocUtils**类用于合成试卷各模块，整合成一张卷，**TemplateOutputUtils**类用于在各题型模板填充好后输出到docx，避免重复代码。

**cn.hengxin.paper.generator**包下的**PaperGenerator**类用于生成试卷