一、需求分析

(1) 能够自动生成简单运算的式子，可以定义其的数量和大小。

(2)运算符要包括"+，-，x，+”，其中还要支持括号和分数的计算。

(3)程序要能够自己生成答案，并保存在文件夹中。

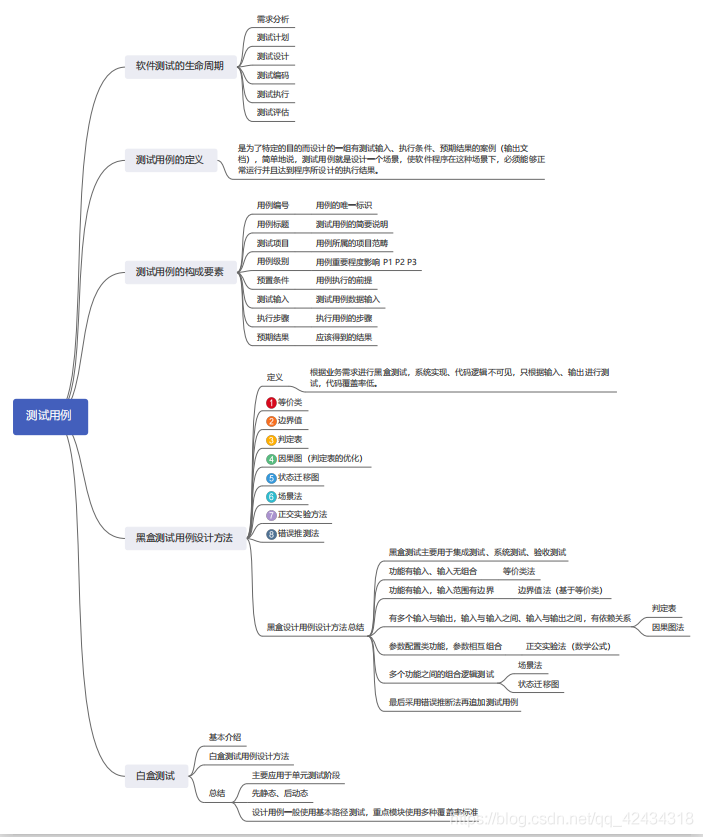
(4)可以计算做对以及做错的题数和题号

二、功能设计

通过控制台生成四则运算，进行查重，将式子写入Exercises.txt中，

计算每道题的结果写入Answers.txt

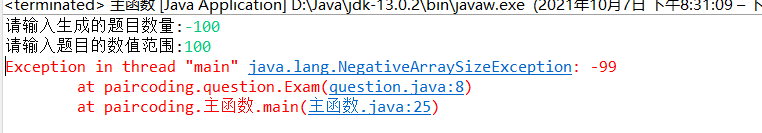
在shurudaan.txt中作答，通过比较，在Grade.txt中生成成绩



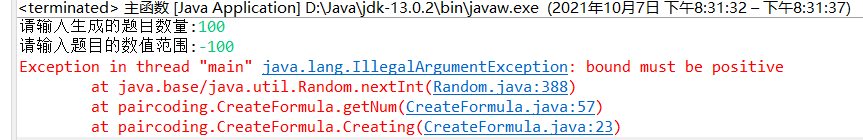
测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例序号 | 用例描述 | 输入数据 | 预期输出数据 | 实际输出数据 | 通过/不通过 | 评价 |
| 1 | 题目生成数量为负 | {-100，100} | 错误，无法生成文件 | 错误 | 通过 | 题目数无法为负数，正确 |
| 2 | 取值范围为负 | {100，-100} | 错误，无法生成文件 | 错误 | 通过 | 说明题目内数字需要大于零 |
| 3 | 生成题目数量无限大 | {100000，100} | 能生成100000题 | 能成功生成 | 通过 | 证明算法正确性 |
| 4 | 取值范围无限大 | {100，100000} | 能成功生成文件 | 能成功生成 | 通过 | 证明算法正确性 |

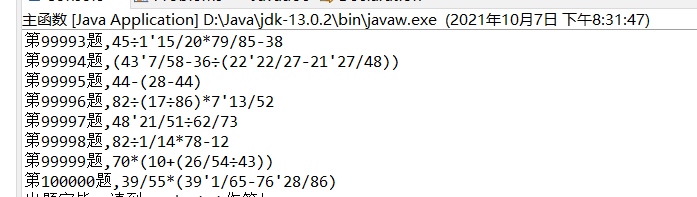
{-100，100}：



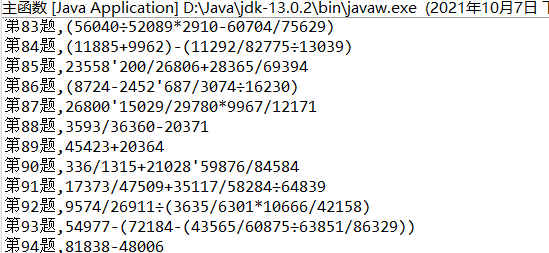
{100，-100}：



{100000，100}:



{100，100000}:



1、伪码描述：

Test main(){//测试类

//解析用户命令

CommandAnalyze.getRange();}

CommandAnalyze.getRange(){

/\*

\*获取用户命令并解析

\*不完整则调用自己重新输入命令

\*完整则返回int数组

\*/

//转入题数和数值范围以生成表达式

DataGenerate.generateExp(num.range):

}

DataGenerate.generateExpint numjint range){

/\*

\*生成表达式和答案

\*/

//将生成的表达式和答案以HashMap返回

DataStorage. printExp(expMap):

}

DataStorage printExp(HashMap expMap){

/\*

\*io流将表达式和答案

\*以字符串形式写入文件

\*/

}

/\*System.out.println("Repeat:"+(re));

System.out.println("RepeatDetail:");

for(int j=0;j<re;j++)

{

System.out.print(j+1);

System.out.print(" ");

System.out.println(repeat[j]);

}\*/

2、注释

@命令解析类: CommandAnalyze

@方法名: public void getRange0

@功能:获取并解析用户指令，不完整则需重新输入

@梦数:无”

@返回值:无

@数据生成类: DataGenerate

方法:

@方法名: public HashMap generateExpint num int range)

@功能:根据题日数和数值范国生成表达式和答案:

同时要兼顾需求

@参数: int, int题目数，数值范围

@返回值: HashMap (表达式和答案)

@数据存储类: DataStorage

方法:

@方法名: public void printExp( HashMap expMap)

@功能:将表达式和客案分别写入文件

@参数: HashMap (表达式和答案

@返回值:无