**Python实验二**

**E51814014 吴振龙**

**【自动生成四则运算题库】**

首先将四个运算符组织在列表中

|  |
| --- |
| operators = ['+', '-', '\*', '/'] |

生成一道题目的函数为generateExpression(operatorNum)，generateExpression函数接收一个参数operatorNum，表示想要生成题目中四则运算符的个数，根据运算符的个数可以确定运算式的长度，可知题目中运算数的个数是operatorNum+1。函数先初始化一个储存表达式的字符串expressionString、一个存储运算数的列表numbersList, 为了支持生成括号，需要一个记录型变量leftBracket，记录此时表达式中没有被右括号匹配的左括号数量，初始化为0。

初始化完之后，函数会进行operatorNum+1次循环，前operatorNum次循环中，每次生成一个运算数和一个运算符，最后一次循环只生成一个运算数。

每次循环所做的操作依次为：

1、在生成运算符之前，先判断是否要生成左括号，

|  |
| --- |
| leftBracketThisTurn =  random.randint(0, 1) |

如果leftBracketThisTurn随机到1，则生成一个左括号，并将leftBracket的值加一，否则就不生成。但注意还需要考虑一种情况，如果这是第operatorNum+1次循环，只生成运算数，所以此时也不生成左括号：

|  |
| --- |
| leftBracketThisTurn =  random.randint(0, 1)  if leftBracketThisTurn and i != operatorNum:      # 随机到1且本轮不是生成最后一个运算符时，生成一个左括号      expressionString += '('       leftBracket += 1 |

2、随机生成一个1-99的数字。注意，如果运算式的最后一个运算符是除，则不生成1。

|  |
| --- |
| # 生成数字  if expressionString and expressionString[-1] == '/':       # 如果最后一个运算符是除，不生成1        numbersList.append(random.randint(2, 99))  else:         numbersList.append(random.randint(1, 99))  expressionString += str(numbersList[-1]) |

3、判断是否已经生成了足够数量的运算符，即判断这是否是最后一次循环，如果是，将不足的右括号在末尾补全，结束循环。

|  |
| --- |
| if i == operatorNum:        # 如果已经生成了足够多的运算符，在此结束，不再生成运算符        # 在运算式末尾补全右括号        for i in range(leftBracket):              expressionString += ')'        break |

4、判断是否要生成右括号，如果 本轮没有生成过左括号（防止一个数字两边有两个括号）且random.randint(0, 1)随机到1且目前运算式中有未匹配到的左括号，生成一个右括号。

|  |
| --- |
| if not leftBracketThisTurn and leftBracket>0 and random.randint(0, 1):        # 如果本轮没有生成过左括号且此时有未匹配过的左括号随机到1，生成一个右括号        expressionString += ')'         leftBracket -= 1 |

5、随机生成一个运算符，方法是随机生成一个0-3的数字作为operators的索引，索引对应的运算符就是要生成的运算符。注意，如果上一个数字是1，不生成乘号。

|  |
| --- |
| # 随机生成一个运算符    if numbersList[-1] == 1:        # 如果最后一个数字是1，运算符不生成乘号        expressionString += ['+', '-', '/'][random.randint(0, 2)]   else:        expressionString += operators[random.randint(0, 3)] |

循环结束之后再进行判断，是否会出现0除的情况：使用eval直接计算字符串形式的运算式，再使用try-except捕获ZeroDivisionError异常，如果出现0除，返回空，否则返回字符串运算式和保留两位小数的运算结果。

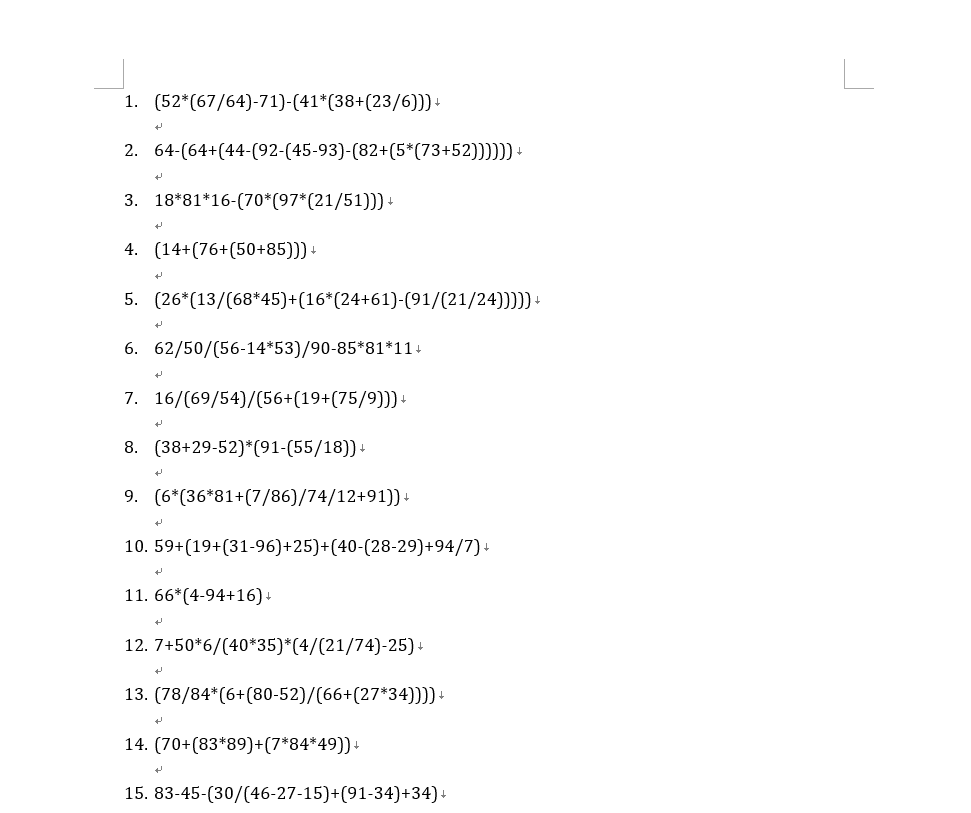
|  |
| --- |
| try:      eval(expressionString)  except ZeroDivisionError:      # 表达式中出现0除，返回空      return None, None  answer = eval(expressionString)  # 答案保留两位小数  answer = "{:.2f}".format(answer)  return expressionString, answer |

主程序中循环50次，每次调用generateExpression函数随机生成长度为3-9的运算式，将运算式字符串保存到expressions列表，将参考答案保存到answers列表，再利用python-docx库将题目和参考答案保存到docx格式文件中。

|  |
| --- |
| *# 创建文档对象*  questionsDocument = docx.Document()  answersDocument = docx.Document()  for i in range(len(expressions)):  *# 以有序列表形式写入文档*      questionsDocument.add\_paragraph(expressions[i], style = 'List Number')      answersDocument.add\_paragraph(answers[i], style = 'List Number')  questionsDocument.save(questionsPath)  answersDocument.save(answersPath) |

**【运行结果】**

**题目**



**参考答案**

