# 异常与日志实验报告

马斓轩 1813076

## 一、对初级表达式计算实现异常的抛出与处理

### 1. 自定义异常：

```
class expressionException extends Exception{
    private Solution ex;
    expressionException(Solution ex,String msg){
        super(msg);
        this.ex=ex;
    }
    public Solution getStr() {
        return ex;
    }
}
```

### 2. 异常类型：

①表达式中有非数字

```
if(!tmp.equals("+")&&!tmp.equals("-")&&!tmp.equals("*")&&!tmp.equals("/")&&!tmp.equals("%")&&!tmp.equals("(")
    if(!tmp.matches("^[-+]?\\d+(\\.\\d+)?$ ")) {
        throw(new expressionException(this,"formException"));
    }
    num.push(tmp);
    continue;
}
```

控制台输出

```
a+2=
a+2=:formException
```

```
3$65=
3$65=:formException
```

②空字符串抛出

```
if(str==" ") { throw(new expressionException(this,"null expression")); }
```

3. 异常处理

```
Solution exp=new Solution(str);
try {
    exp.compute();
}catch(expressionException s) {
    System.out.println(s.getStr().str+":"+s.getMessage());
}
```

## 二、异常日志记录

控制台输出日志信息：

```
public static void main(String[] args)throws Exception {
    Scanner in=new Scanner(System.in);
    String str=in.nextLine();
    Solution exp=new Solution(str);
    Log.setLevel(Level.CONFIG);
    try {
        exp.compute();
    }catch(expressionException s) {
        Log.info(""+s.getMessage());
        System.out.println(s.getStr().str+":"+s.getMessage());
    }
```

```
a+2
a+2:formException
五月05, 2020 10:38:58 上午 Expression_evaluation.Solution main
信息: formException
```

附源代码：

```
import java.util.*;
import java.util.logging.*;

class expressionException extends Exception{
    private Solution ex;
    expressionException(Solution ex,String msg){
        super(msg);
        this.ex=ex;
```

```
    }
    public Solution getStr() {
        return ex;
    }
}

public class Solution {
    public final static Logger Log=Logger.getLogger("Exception");
    private String str;
    public Solution(String str){
        super();
        this.str=str;
    }
    String[] stringArray(String s) {
        ArrayList<String> list=new ArrayList<>();
        String num="";
        for(int i=0;i<s.length();i++) {
            char tmp=s.charAt(i);
            if(tmp==' ') {
                continue;
            }
            else if(num==""&&tmp=='-'&&(list.isEmpty()||list.get(list.size()-
1).equals("("))){
                num+=tmp;
            }
            else if(tmp>='0'&&tmp<='9'||tmp=='.') {
                num+=tmp;
                if(i==s.length()-1) {
                    list.add(num);
                }
            }
            else {
                if(i!=0&&num!="") {
                    list.add(num);
                }
                num=tmp+"";
                list.add(num);
                num="";
            }
        }
        return list.toArray(new String[list.size()]);
    }

    String optCompute(String v1,String opt,String v2){
```

```
if(v1.matches("^-?[0-9]+$")&&v2.matches("^-?[0-9]+$")) {
	int a=Integer.parseInt(v1);
	int b=Integer.parseInt(v2);
	switch (opt) {
		case "+": return a+b+"";
		case "-": return a-b+"";
		case "*": return a*b+"";
		case "/": return a/b+"";
		case "%": return a%b+"";
	}
}
else if(!v1.matches("^-?[0-9]+$")&&v2.matches("^-?[0-9]+$")) {
	float a=Float.parseFloat(v1);
	int b=Integer.parseInt(v2);
	switch (opt) {
		case "+": return a+b+"";
		case "-": return a-b+"";
		case "*": return a*b+"";
		case "/": return a/b+"";
		case "%": return a%b+"";
	}
}
else if(v1.matches("^-?[0-9]+$")&&!v2.matches("^-?[0-9]+$")) {
	int a=Integer.parseInt(v1);
	float b=Float.parseFloat(v2);
	switch (opt) {
		case "+": return a+b+"";
		case "-": return a-b+"";
		case "*": return a*b+"";
		case "/": return a/b+"";
		case "%": return a%b+"";
	}
}
else if(!v1.matches("^-?[0-9]+$")&&!v2.matches("^-?[0-9]+$")) {
	float a=Float.parseFloat(v1);
	float b=Float.parseFloat(v2);
	switch (opt) {
		case "+": return a+b+"";
		case "-": return a-b+"";
		case "*": return a*b+"";
		case "/": return a/b+"";
		case "%": return a%b+"";
	}
}
```

```
        return 0+"";
    }

    String compute()throws expressionException {

        if(str==" ") { throw(new expressionException(this,"null expression")); }

        String[] res=stringArray(str);
        LinkedList<String> opt=new LinkedList<>();
        LinkedList<String> num=new LinkedList<>();
        for(String tmp:res) {
            if(!tmp.equals("+")&&!tmp.equals("-
")&&!tmp.equals("*")&&!tmp.equals("/")&&!tmp.equals("%")&&!tmp.equals("(")&
&!tmp.equals(")")&&!tmp.equals("=")) {
                if(!tmp.matches("^[-+]?\\d+(\\.\\d+)?$ ")) {
                    throw(new expressionException(this,"formException"));
                }
                num.push(tmp);
                continue;
            }
            else if(tmp.equals("+")||tmp.equals("-")) {
                while(!opt.isEmpty()) {
                    if(!opt.peek().equals("(")) {
                        String v2=num.pop();
                        String v1=num.pop();
                        num.push(optCompute(v1,opt.pop(),v2));
                    }
                    else {
                        break;
                    }

                }
                opt.push(tmp);
            }
            else if(tmp.equals("*")||tmp.equals("/")||tmp.equals("%")) {
                while(!opt.isEmpty()) {

    if(opt.peek().equals("*")||opt.peek().equals("/")||opt.peek().equals("%")) {
                        String v2=num.pop();
                        String v1=num.pop();
                        num.push(optCompute(v1,opt.pop(),v2));
                    }
                    else {
                        break;
```

```
                    }
                }
                opt.push(tmp);
            }
            else if(tmp.equals("(")) {
                opt.push(tmp);
            }
            else if(tmp.equals(")")) {
                while(!opt.isEmpty()&&!opt.peek().equals("(")) {
                    String v2=num.pop();
                    String v1=num.pop();
                    num.push(optCompute(v1,opt.pop(),v2));
                }
                if(!opt.isEmpty()&&opt.peek().equals("(")) {
                    opt.pop();
                }
            }
            else if(tmp.equals("=")) {
                continue;
            }

        }
        while(!opt.isEmpty()) {
            String v2=num.pop();
            String v1=num.pop();
            num.push(optCompute(v1,opt.pop(),v2));
        }
        return num.peek();
    }


    public static void main(String[] args)throws Exception {
        Scanner in=new Scanner(System.in);
        String str=in.nextLine();
        Solution exp=new Solution(str);
        Log.setLevel(Level.CONFIG);
        try {
            exp.compute();
        }catch(expressionException s) {
            Log.info(""+s.getMessage());
            System.out.println(s.getStr().str+":"+s.getMessage());
        }
        in.close();
    }
```

}