Отчёт HT09

Ошибки

1. 152 строка

} else {  
 dA = Double.parseDouble(sTmp);  
 stack.push(dA);

Происходит ошибка парса к даблу, которая никак не обрабатывается функцией.

Решение:

try {  
 dA = Double.*parseDouble*(sTmp);  
 stack.push(dA);  
}  
catch (NumberFormatException n)  
{  
 throw new CalcException("Bad input");  
}

Обернул парс try-catch с ловлей NumberFormatException, который возникает только при неправильном парсе, тогда если данные в правильном виде, но вызывают исключение, то оно пробросится дальше.

2) 126 строка

// Считываем первый символ  
switch (sTmp.charAt(1)) {  
 case '+':

sTmp.charAt(1) смотрит на второй символ sTmp, хотя условия для входа в данную секцию кода включает sTmp.length()==1, т.е. sTmp.charAt(1) всегда будет приводить к исключению, да и читать по логике нужно первой символ, ведь мы читаем строку на вход посимвольно.

Решение:

// Считываем первый символ  
switch (sTmp.charAt(0)) { //1 to 0  
 case '+':

Просто меняем индекс на 0 и всё работает.

3) 145 строка

case '!':  
 throw new UnsupportedOperationException("TODO: Не забыть реализовать оператор !");

Оператор ! не реализован и в задании не прописано что делать.

Предложенное решение: просто удалить данный case.

4) 160 строка

if (stack.size() > 1) {  
 throw new CalcException("Количество операторов не соответствует количеству операндов");  
}

Нет проверки что стэк может быть пустым.

Решение:

if (stack.size() != 1) { //changed > to !=  
 throw new CalcException("Количество операторов не соответствует количеству операндов");  
}

Статистика фаззинга:

Предел проверки достигается при Cov 42, ft 197.