

1 项目设计思路

本项目实现了一个简单的计算器，能够从输入文件中读取表达式，将中缀表达式转换为后缀表达式，然后计算后缀表达式的值，并将结果写入输出文件。以下是项目的设计思路：

1. **Calculator 类**: 这个类包含了计算器的主要功能。它有以下方法和属性：

- `Calculator(const std::string& inputFileName, const std::string& outputFileName)`: 构造函数，用于初始化输入和输出文件流，同时检查文件是否成功打开。
- `int getOperatorPriority(char c)`: 根据运算符的优先级返回一个整数。
- `bool isOperator(char c)`: 检查一个字符是否为运算符。
- `std::string remove_illegalchar(std::string expression)`: 删除表达式中的非法字符。
- `std::queue<std::string> in_to_post(std::string expression)`: 将中缀表达式转换为后缀表达式。
- `double cal_post(std::queue<std::string> postfix)`: 计算后缀表达式的值。
- `void processInputFile()`: 读取输入文件，处理每行表达式，计算结果并写入输出文件。
- `std::ifstream inputFile`: 用于读取输入文件的文件流。
- `std::ofstream outputFile`: 用于写入输出文件的文件流。

2 测试说明

为了测试计算器的功能，我们提供了一个输入文件 `input.txt`，其中包含了多个表达式，每行一个。以下是输入文件的内容：

```
3.1 * (1.2 + 3.5)
1.2sdf3!!3 - 0.23
1t.y21 * (3RR4T + 1s.s00)
1+)2
1((2
4/0
```

2.1 运行程序

为了运行程序，只需执行 `main()` 函数即可。程序将从 `input.txt` 读取表达式，并将结果写入 `output.txt` 文件。

```
int main()
{
    Calculator calculator("input.txt", "output.txt");
    calculator.processInputFile();
    return 0;
}
```

2.2 结果分析和结论

根据输入文件中的表达式，程序将计算结果写入输出文件。结果如下：

输入表达式	测试结果
3.1 * (1.2 + 3.5)	14.57
1.2sdf3!!3 - 0.23	1.003
1t.y21 * (3RR4T + 1s.s00)	42.35
1+)2	Error: Unmatched closing parenthesis.
1((2	Error: Unmatched opening parenthesis.
4/0	Error: Division by zero is not allowed.

计算器成功计算了合法表达式，正确报告了括号不匹配的错误，以及不允许除以零的错误。因此，计算器在处理各种情况时表现出良好的稳定性和健壮性。