

Report

- 一、思路
- 二、Highlight
- 三、代码
 - 1. 总览
 - 2. 结构体 `num`
 - 3. 字符串预处理 `num formal(string);`
 - 4. 高精度乘法 `string mul(num str1, num str2)`
 - 5. `main()`
- 四、效果展示

Report

一、思路

1. 定义了一个结构体`num`，储存了表示一个数的具体信息。
2. 对输入的字符串进行处理，用结构体`num`对数字进行表示。
3. 对`num`结构的数据进行高精度乘法操作，并将结果整合为可供输出的字符串。
4. 在过程中，对异常输入进行捕获，并输出提示。

二、Highlight

1. **功能强大而代码精简**，总计**100**余行
2. 良好的代码风格&方法分割，**高内聚低耦合**
3. 支持**任意位数**的乘法
4. 支持**科学计数法**计算
5. 支持**多种计算组合**，详见**第四部分效果展示**

三、代码

1. 总览

由一个 `struct`，两个方法 `mul()` 和 `formal()` 以及 `main()` 构成。

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  > struct num ...
12
13  string mul(num str1, num str2);
14
15  num formal(string);
16
17  > int main(int argc, char *argv[]) ...
33
34  > string mul(num str1, num str2) ...
92
93  > num formal(string ss) ...
132
```

2. 结构体 `num`

```
struct num
{
    int sign = 1; // 符号位 1 / -1
    string value; // e之前 除去小数点的数字
    long long point; // 小数点位置
    double e; // 指数部分
};
```

3. 字符串预处理 `num formal(string);`

```
num formal(string ss)
{
    num result;
    int idx = ss.find("e");
    if (idx == string::npos)
    {
        result.e = 0;
        idx = ss.length();
    }
    else
    {
        result.e = stold(ss.substr(idx+1));
    }

    string s = ss.substr(0,idx);
```

```

    if (s.substr(0,1) == "-")
    {
        result.sign = -1;
        s = s.substr(1);
    }
    else
    {
        result.sign = 1;
    }

    int idx2 = s.find(".");
    if (idx2 == string::npos)
    {
        result.point = 0;
        result.value = s;
    }
    else
    {
        result.point = s.length() - 1 - idx2;
        result.value = s.erase(idx2, 1);
    }

    return result;
}

```

4.高精度乘法 string mul(num str1, num str2)

```

string mul(num str1, num str2)
{
    int len1 = str1.value.size();
    int len2 = str2.value.size();
    int len3 = len1 + len2;
    long long p = str1.point + str2.point;
    int n1[len1] = {};
    int n2[len2] = {};
    int n3[len3] = {};
    string result = "";

    int sign = str1.sign * str2.sign;
    if (sign == -1)
    {
        result += "-";
    }

    for (int i = 0; i < len1; i++)
    {
        n1[i] = stoi(str1.value.substr(len1 - 1 - i, 1));
    }
    for (int i = 0; i < len2; i++)
    {
        n2[i] = stoi(str2.value.substr(len2 - 1 - i, 1));
    }

    for (int i = 0; i < len1; i++)
    {
        for (int j = 0; j < len2; j++)

```

```

    {
        n3[len3 - 1 - i - j] += n1[i] * n2[j];
        n3[len3 - 2 - i - j] += n3[len3 - 1 - i - j] / 10;
        n3[len3 - 1 - i - j] %= 10;
    }
}

// 整合成用于输出的字符串
int cnt = 0;
for (; cnt < len3 - p - 1; cnt++)
{
    if (n3[cnt] != 0)
        break;
}

for (; cnt < len3; cnt++)
{
    if (p != 0 && cnt == len3 - p)
        result += ".";
    result += to_string(n3[cnt]);
}

if (str1.e + str2.e != 0)
{
    result += "e";
    result += to_string(str1.e + str2.e);
}

return result;
}

```

5. main()

```

int main(int argc, char *argv[])
{
    try
    {
        num num1 = formal(argv[1]);
        num num2 = formal(argv[2]);

        string result = mul(num1, num2);
        printf("%s * %s = %s", argv[1], argv[2], result.c_str());
    }
    catch(const std::exception& e)
    {
        printf("The input cannot be interpret as numbers!");
    }
    return 0;
}

```

四、效果展示

1. 基础乘法

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 2 3  
2 * 3 = 6
```

2. 小数乘法

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 3.1416 2  
3.1416 * 2 = 6.2832
```

3. 高精度小数

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 0.0000005 0.0001  
0.0000005 * 0.0001 = 0.0000000005
```

4. 小数与科学计数法

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 3.1415 2.0e-2  
3.1415 * 2.0e-2 = 6.28300e-2.000000
```

5. 大整数相乘

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 1234567890 -1234567890  
1234567890 * -1234567890 = -1524157875019052100
```

6. 科学计数法相乘

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe 1.0e200 1.0e200  
1.0e200 * 1.0e200 = 1.00e400.000000
```

7. 非法输入提示

```
(base) PS D:\gitRepo\C-project\project1> g++ .\mul.cpp ; .\a.exe a 2  
The input cannot be interpret as numbers!
```