

文本处理工具类实现

题目描述

[刷新](#)

实现一个文本处理工具类 `TextProcessor`，提供常用的字符串处理功能：

1. 基础字符串操作：移除所有空格、转换为小写、转换为大写、反转字符串
2. 字符串查找和替换：统计子串出现次数、查找所有位置、全部替换
3. 字符串分割：按字符分割
4. 统计：字符数统计（包含空格）、字数统计（不包含空格）、行数统计
5. 字符串验证：是否回文、是否为数字，是否只包含指定字符
6. 格式化功能：居中对齐、左对齐、右对齐

`TextProcessor.h` 的如下，可通过注释来理解每个函数的作用：

```
class TextProcessor {
public:
    // 构造函数
    TextProcessor();
    TextProcessor(const std::string& text);

    // 设置和获取文本
    void setText(const std::string& text);
    std::string getText() const;

    // 基础字符串操作
    std::string removeSpaces();           // 移除文本里所有的空格，并返回移除空格后的字符串
    std::string toLowerCase();            // 将字符串里所有字符均转为小写，并返回转换后的字符串
    std::string toUpperCase();            // 将字符串里所有字符均转为大写，并返回转换后的字符串
    std::string reverse();                // 将字符串反转，并返回反转后的字符串

    // 字符串查找和替换
    int countOccurrences(const std::string& substr); // 返回子串substr在字符串中出现的次数，如果从未出现则返回0
    std::vector<int> findPositions(const std::string& substr); // 返回子串substr的首字母在字符串中的所有位置，如果不含该子串
    std::string replaceAll(const std::string& from, const std::string& to); // 将字符串中的from字符串全部替换为to字符串

    // 字符串分割
    std::vector<std::string> split(char delimiter); // 按字符delimiter分割

    // 字符和单词统计
    int getCharCount();                    // 统计字符数（包含空格）
    int getCharCountNoSpaces();            // 统计字符数（不含空格）
    int getLineCount();                   // 统计行数（最少有一行）

    // 字符串验证
    bool isPalindrome();                  // 是否为回文字符串
    bool isNumeric();                    // 是否为数字（允许开头有正负号，允许是小数）
    bool containsOnly(const std::string& charset); // 是否只包含指定字符charset

    // 格式化功能
    std::string centerAlign(int width, char fillChar = ' '); // 使用fillChar进行填充（默认为空格），将字符串以居中对齐的方式扩充
    std::string leftAlign(int width, char fillChar = ' ');   // 使用fillChar进行填充（默认为空格），将字符串以左对齐的方式扩充
    std::string rightAlign(int width, char fillChar = ' ');  // 使用fillChar进行填充（默认为空格），将字符串以右对齐的方式扩充

private:
    std::string text_;
    bool isDigit(char c) const {
        // 辅助函数，判断c是否是数字
        return c >= '0' && c <= '9';
    }
    bool isAlpha(char c) const {
        // 辅助函数，判断c是否是字母
        return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
    }
};
```

完整的 `TextProcessor.h` 文件可从此处下载：下载链接 (</staticdata/problem/2177.IO9Rt2ChhZKcglw9/TextProcessor.h>)

你需要提交 `TextProcessor.cpp` 文件，实现头文件中定义的相关功能

附：TextProcessor.h 中函数的使用样例参考如下：

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include <sstream>
#include <cassert>
#include "TextProcessor.h"
using namespace std;

int main()
{
    TextProcessor a("abcdefgh");
    cout<<a.getText()<<endl;
    a.setText("hgfedcba");
    cout<<a.getText()<<endl;
    TextProcessor b("Ab Cd Ef Gh");
    cout<<b.removeSpaces()<<endl;
    cout<<b.toLowerCase()<<endl;
    cout<<b.toUpperCase()<<endl;
    cout<<b.reverse()<<endl;
    TextProcessor c("Ab Cd Ef Gh hgfedcba Ab Cd Ef Gh");
    cout<<c.countOccurrences("Ab")<<endl;
    cout<<c.findPositions("Ab").size()<<endl<<c.findPositions("Ab")[1]<<endl;
    cout<<c.replaceAll("Ab", "Ba")<<endl;
    cout<<b.getCharCount()<<endl;
    cout<<b.getCharCountNoSpaces()<<endl;
    cout<<b.getLineCount()<<endl;
    TextProcessor d("a bcdef gH hgfeD cba");
    cout<<d.isPalindrome()<<endl;
    cout<<a.isPalindrome()<<endl;
    TextProcessor e("-114.514");
    TextProcessor f("3.14159.26532");
    TextProcessor g("-.");
    cout<<e.isNumeric()<<endl;
    cout<<f.isNumeric()<<endl;
    cout<<d.containsOnly("abcdefgh")<<endl;
    cout<<b.containsOnly("abcdefgh")<<endl;
    cout<<a.centerAlign(15, '1')<<endl;
    cout<<a.leftAlign(12, '1')<<endl;
    cout<<a.rightAlign(12)<<endl;
    return 0;
}
/*
以下为输出参考
abcdefgh
hgfedcba
AbCdEfGh
ab cd ef gh
AB CD EF GH
hG fE dC bA
2
2
21
Ba Cd Ef Gh hgfedcba Ba Cd Ef Gh
11
8
1
1
0
1
0
0
0
111hgfedcba1111
hgfedcba1111
    hgfedcba
*/
```

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	oop_custom	make		1048576 B

递交历史

#	状态	时间
352970	Accepted	2025-06-02 22:50:02
352967	Accepted	2025-06-02 22:48:15
352966	No Source	2025-06-02 22:42:46
352964	Accepted	2025-06-02 22:39:41
352961	No Source	2025-06-02 22:38:46
352917	Accepted	2025-06-02 21:30:33
352916	No Source	2025-06-02 21:29:49
352913	No Source	2025-06-02 21:27:48
352905	Wrong Answer	2025-06-02 21:18:39
352899	Wrong Answer	2025-06-02 21:11:15

12

当前没有提交权限！