

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

Лабораторная работа №5
по дисциплине
«Объектно-ориентированное программирование»

Выполнил:
Пантелеев Никита Андреевич
Студент 2 курса группы ПИН-б-о-22-1
Направления подготовки
09.03.03 Прикладная информатика
очной формы обучения

Ставрополь, 2023 г.

Тема: Файловые и строковые потоки. Строки класса string.

Цель работы: изучить файловые и строковые потоки и научиться реализовать их.

Выполнение работы:

Вариант -16

Написать программу, которая считывает текст из файла и выводит на экран сначала вопросительные, а затем восклицательные предложения.

Листинг:

```
// main.cpp
```

```
#include <iostream>
```

```
#include "TextProcessor.h"
```

```
#include "TextProcessorTest.h"
```

```
int main() {
```

```
    setlocale(LC_ALL, "Russian");
```

```
    TextProcessor processor;
```

```
    processor.readFromFile("input.txt");
```

```
    std::cout << "Вопросительные предложения:\n";
```

```
    processor.printQuestions();
```

```
    std::cout << "\nВосклицательные предложения:\n";
```

```
    processor.printExclamations();
```

```
testTextProcessor();
```

```
return 0;
```

```
}
```

```
// TextProcessor.h
```

```
#ifndef TEXTPROCESSOR_H
```

```
#define TEXTPROCESSOR_H
```

```
#include <string>
```

```
#include <vector>
```

```
class TextProcessor {
```

```
public:
```

```
void readFromFile(const std::string& filename);
```

```
void printQuestions() const;
```

```
void printExclamations() const;
```

```
bool isQuestion(const std::string& sentence) const;
```

```
bool isExclamation(const std::string& sentence) const;
```

```
// Add a public method to access sentences vector
```

```
const std::vector<std::string>& getSentences() const {
```

```
    return sentences;
```

```
}
```

```
private:
```

```
    std::vector<std::string> sentences;
```

```
};
```

```
#endif // TEXTPROCESSOR_H
```

```
#include "TextProcessor.h"
```

```
#include <fstream>
```

```
#include <algorithm>
```

```
#include <cctype>
```

```
#include <iostream>
```

```
void TextProcessor::readFromFile(const std::string& filename) {
```

```
    std::ifstream file(filename);
```

```
    if (file.is_open()) {
```

```
        std::string sentence;
```

```
        while (std::getline(file, sentence)) {
```

```
            sentences.push_back(sentence);
```

```
        }
```

```
        file.close();
```

```
    }
```

```
}
```

```
void TextProcessor::printQuestions() const {
```

```
    for (const auto& sentence : sentences) {
```

```
        if (isQuestion(sentence)) {
```

```
            std::cout << sentence << '\n';
```

```
        }
```

```
    }
```

```
}
```

```
void TextProcessor::printExclamations() const {
```

```
    for (const auto& sentence : sentences) {
```

```
        if (isExclamation(sentence)) {
```

```
            std::cout << sentence << '\n';
```

```
        }
```

```
    }
```

```
}
```

```
bool TextProcessor::isQuestion(const std::string& sentence) const {
```

```
    return !sentence.empty() && sentence.find('?') != std::string::npos;
```

```
}
```

```
bool TextProcessor::isExclamation(const std::string& sentence) const {
```

```
    return !sentence.empty() && sentence.find('!') != std::string::npos;
}
```

```
// TextProcessorTest.h
```

```
#ifndef TEST_TEXT_PROCESSOR_H
```

```
#define TEST_TEXT_PROCESSOR_H
```

```
void testTextProcessor();
```

```
#endif TEST_TEXT_PROCESSOR_H
```

```
// TextProcessorTest.cpp
```

```
#include <cassert>
```

```
#include <iostream>
```

```
#include "TextProcessor.h"
```

```
#include "TextProcessorTest.h"
```

```
void testTextProcessor() {
```

```
    // Test case 1: Check if reading from a file works
```

```
    TextProcessor processor;
```

```
    processor.readFromFile("input.txt");
```

```
    assert(!processor.getSentences().empty());
```

```

// Test case 2: Check if questions are correctly identified

assert(processor.isQuestion("Is this a question?"));

assert(!processor.isQuestion("This is not a question.));

// Test case 3: Check if exclamations are correctly identified

assert(processor.isExclamation("Wow!"));

assert(!processor.isExclamation("This is not an exclamation.));

std::cout << "All tests passed successfully!\n";

}

/*

int testMain() {

    testTextProcessor();

    return 0;

}

*/

```

TextProcessor
sentences: std::vector
readFromFile(filename:const std::string&)
printQuestions() const
printExclamations() const
isQuestion(sentence:const std::string&)
isExclamation(sentence:const std::string&)
getSentences() const
sentences: std::vector

Ссылка на полностью сделанные задания на github:
<https://github.com/Filin546/OOP>

Вывод: изучил файловые и строковые потоки и научился реализовать их.