МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №8**

по дисциплине: Объектно-ориентированное программирование Тема: Создание шаблонов классов в С++.

Выполнил: студент группы ПВ-223

Дмитриев А.А.

Проверил:

Черников С.В.

Белгород 2024 г.

**Цель работы:** Получение теоретических знаний о шаблонах классов в С++. Получение практических навыков по созданию классов-шаблонов С++.

**Задание:**

1. Изучить теоретические сведения о шаблонах классов в С++.

2. Разработать программу в соответствии с заданным вариантом задания.

3. Оформить отчет.

2 вариант: Стек. Дан файл содержащий текст, который представляет собой различные предложения. Считать предложения из текста в разработанную структуру данных таким образом, чтобы первыми словами были существительные, далее глаголы, затем все остальное. Для получения информации к кокой части речи принадлежит слово использовать справочники, хранящиеся в других файлах(использовать те же разработанные шаблоны).

Stack.h

#pragma once

#include <vector>

template<typename T>

class Stack

{

private:

std::vector<T> \*stack;

public:

Stack() {

stack = new std::vector<T>();

}

~Stack() {

delete stack;

}

void add(T t) {

stack->push\_back(t);

}

T pop() {

T temp = see();

stack->pop\_back();

return temp;

}

T see() {

return \*(stack->end()-1);

}

int size() {

return stack->size();

}

bool isEmpty() {

return stack->empty();

}

void clear() {

stack->clear();

}

};

WordCollector.h

#pragma once

#include <fstream>

#include <iostream>

#include <exception>

#include <set>

#include "Stack.h"

class WordCollector

{

private:

std::set<std::string>\* hashSetNoun;

std::set<std::string>\* hashSetVerb;

std::set<std::string>\* hashSetOther;

Stack<std::string>\* stackNoun;

Stack<std::string>\* stackVerb;

Stack<std::string>\* stackOther;

std::set<std::string> \*getWordSet(std::string fileName) {

std::ifstream file(fileName);

if (!file.is\_open()) {

std::string msg = "Не удалось открыть файл: " + fileName + "\n";

throw std::ifstream::failure(msg.data());

}

std::set<std::string>\* res = new std::set<std::string>();

std::string word;

while (file >> word)

res->insert(word);

file.close();

return res;

}

bool isWordInSet(std::set<std::string> \*set, std::string word) {

return set->find(word) != set->end();

}

public:

WordCollector() {

try {

hashSetNoun = getWordSet("dict\_noun.txt");

hashSetVerb = getWordSet("dict\_verb.txt");

hashSetOther = getWordSet("dict\_other.txt");

}

catch (const std::ifstream::failure& e) {

std::cerr << e.what();

throw std::ofstream::failure("");

}

stackNoun = new Stack<std::string>();

stackVerb = new Stack<std::string>();

stackOther = new Stack<std::string>();

}

~WordCollector() {

if (hashSetNoun != nullptr) delete hashSetNoun;

if (hashSetVerb != nullptr) delete hashSetVerb;

if (hashSetOther != nullptr) delete hashSetOther;

if (stackNoun != nullptr) delete stackNoun;

if (stackVerb != nullptr) delete stackVerb;

if (stackOther != nullptr) delete stackOther;

}

void recordToFile(std::string sourceName, std::string destinationName) {

std::ifstream source(sourceName);

if (!source.is\_open()) {

std::string msg = "Не удалось открыть файл: " + sourceName + "\n";

throw std::exception(msg.data());

}

std::ofstream destination(destinationName);

if (!destination.is\_open()) {

std::string msg = "Не удалось создать файл: " + destinationName + "\n";

throw std::exception(msg.data());

}

std::string word;

while (source >> word)

{

if (std::ispunct(word[word.size() - 1]))

word.pop\_back();

for (size\_t i = 0; i < word.size(); i++)

word[i] = std::tolower(word[i]);

if (isWordInSet(hashSetNoun, word))

stackNoun->add(word);

else if (isWordInSet(hashSetVerb, word))

stackVerb->add(word);

else if (isWordInSet(hashSetOther, word))

stackOther->add(word);

else

throw std::runtime\_error("Слова нет в справочнике!");

}

source.close();

while (!stackNoun->isEmpty())

{

destination << stackNoun->pop() << " ";

}

while (!stackVerb->isEmpty())

{

destination << stackVerb->pop() << " ";

}

while (!stackOther->isEmpty())

{

destination << stackOther->pop() << " ";

}

destination.close();

}

};

main.cpp

#include <iostream>

#include "WordCollector.h"

int main()

{

WordCollector \*wc = new WordCollector();

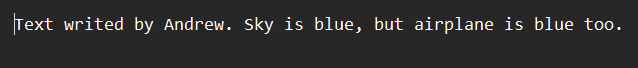
wc->recordToFile("source.txt", "text.txt");

wc->~WordCollector();

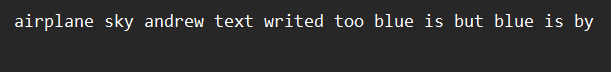
return 0;

}

Пример работы



↓



**Вывод:** В ходе лабораторной работы получили теоретические знания о шаблонах классов в С++. Получили практические навыки по созданию классов-шаблонов С++.