

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 №6**

по дисциплине: «ООП»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Буханов Дмитрий Геннадьевич

Харитонов Сергей Дмитриевич

Белгород 2023 г.

**Тема:** «Потоки в C++»

**Вариант по списку:** 23

**Вариант работы:** 6

**Цель работы:** изучение основных возможностей потоков управления и потоков ввода-вывода. Получение навыков работы со стандартными средствами управления потоками в C++11. Знакомство с классом Thread и стандартными средствами синхронизации потоков.

**Выполнение работы:**

**Задание:** Один поток выполняет подсчет количества гласных букв в тексте, а другой вставляет или удаляет случайным образом гласную букву. Произвести синхронный вывод при каждой итерации. Показать выполнение работы программы в синхронном и асинхронном режимах.

**Код программы:**

```
#include <iostream>
#include <thread>

using namespace std;

void vowelCount(string &s, int &count){
    string str = "aeiouAEIOU";
    for (int i = 0; i < s.size(); i++) {
        if (str.find(s[i]) != string::npos)
            count++;
        cout << "func: vowelCount. Count = " << count << endl;
    }
}

void deleteVowel(string &s) {
    string str = "aeiouAEIOU";
    const std::string empty;
    for (int i = 0; i < s.size(); i++) {
        if (str.find(s[i]) != string::npos && rand() & 1)
            s.replace(i, 1, empty);
        cout << "func: deleteVowel. string = " << s << endl;
    }
}

int main() {
    string res = "abbbabbbaaaaa";
    int count = 0;
    thread t1(vowelCount, ref(res), ref(count));
    thread t2(deleteVowel, ref(res));

    t1.join();
    t2.join();
    cout << "End Work. count = " << count;
    return 0;
}
```

- Результат работы программы в **асинхронном** режиме

```
"D:\BGTU\00P\lab 6\Code\cmake-build-debug\Code.exe"
func: vowelCount. Count = func: deleteVowel. string = bbbabbbbaaaaaa
func: deleteVowel. string = bbbabbbbaaaaaa
func: deleteVowel. string = bbbabbbbaaaaaa
func: deleteVowel. string = bbbbbbbaaaaaa
func: deleteVowel. string = bbbbbbbaaaaaa
func: deleteVowel. string = bbbbbbbaaaaaa
func: deleteVowel. string = bbbbbbbaaaaaa
1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 2
func: vowelCount. Count = 3
func: vowelCount. Count = 4
func: vowelCount. Count = 5
func: deleteVowel. string = bbbbbbbaaaa
func: deleteVowel. string = bbbbbbbaaaa
End Work. count = 5
Process finished with exit code 0
```

- Режим работы программы в **синхронном** режиме:

```
"D:\BGTU\00P\lab 6\Code\cmake-build-debug\Code.exe"
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 1
func: vowelCount. Count = 2
func: vowelCount. Count = 2
func: vowelCount. Count = 2
func: vowelCount. Count = 2
func: vowelCount. Count = 3
func: vowelCount. Count = 4
func: vowelCount. Count = 5
func: vowelCount. Count = 6
func: vowelCount. Count = 7
func: deleteVowel. string = bbbabbbbbaaaaa
func: deleteVowel. string = bbbabbbbbaaaaa
func: deleteVowel. string = bbbabbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaaa
func: deleteVowel. string = bbbbbbbaaaa
func: deleteVowel. string = bbbbbbbaaaa
End Work. count = 7
Process finished with exit code 0
```

## Задание 2:

```
#include <iostream>
#include <Windows.h>
#include <mutex>
#include <string>

using namespace std;
mutex mtx;

DWORD WINAPI vowelCount(LPVOID lpParam){
    string str = "aeiouAEIOU";
    auto s = *(string*)lpParam;
    int count = 0;
    for (int i = 0; i < s.size(); i++) {
        //mtx.lock();
        Sleep(10);
```

```

    if (str.find(s[i]) != string::npos)
        count++;
    cout << "func: vowelCount. Count = " << count << endl;
    //mtx.unlock();
}
}

DWORD WINAPI deleteVowel(LPVOID lpParam) {
    auto s = *(string*)lpParam;
    string str = "aeiouAEIOU";
    const std::string empty;
    for (int i = 0; i < s.size(); i++) {
        //mtx.lock();
        if (str.find(s[i]) != string::npos && rand() & 1)
            s.replace(i, 1, empty);
        cout << "func: deleteVowel. string = " << s << endl;
        //mtx.unlock();
    }
}

int main()
{
    string str = "abbbabbbbaaaaa";
    // Создаем два потока
    HANDLE hThread1 = CreateThread(NULL, 0, deleteVowel, &str, 0, NULL);
    HANDLE hThread2 = CreateThread(NULL, 0, vowelCount, &str, 0, NULL);
    // Ждем, пока оба потока завершат свою работу
    WaitForSingleObject(hThread1, INFINITE);
    WaitForSingleObject(hThread2, INFINITE);
    // Закрываем дескрипторы потоков
    CloseHandle(hThread1);
    CloseHandle(hThread2);
    return 0;
}

```

**Вывод:** изучили основные возможности потоков управления и потоков ввода-вывода. Получили навыки работы со стандартными средствами управления потоками в C++11. Познакомились с классом Thread и стандартными средствами синхронизации потоков.