

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: Объектно-ориентированное программирование  
тема: «Потоки в C++»

Выполнил: ст. группы ПВ-223  
Игнатьев Артур Олегович

Проверил:  
асс. Черников Сергей Викторович

Белгород 2024г.

## Лабораторная работа №6

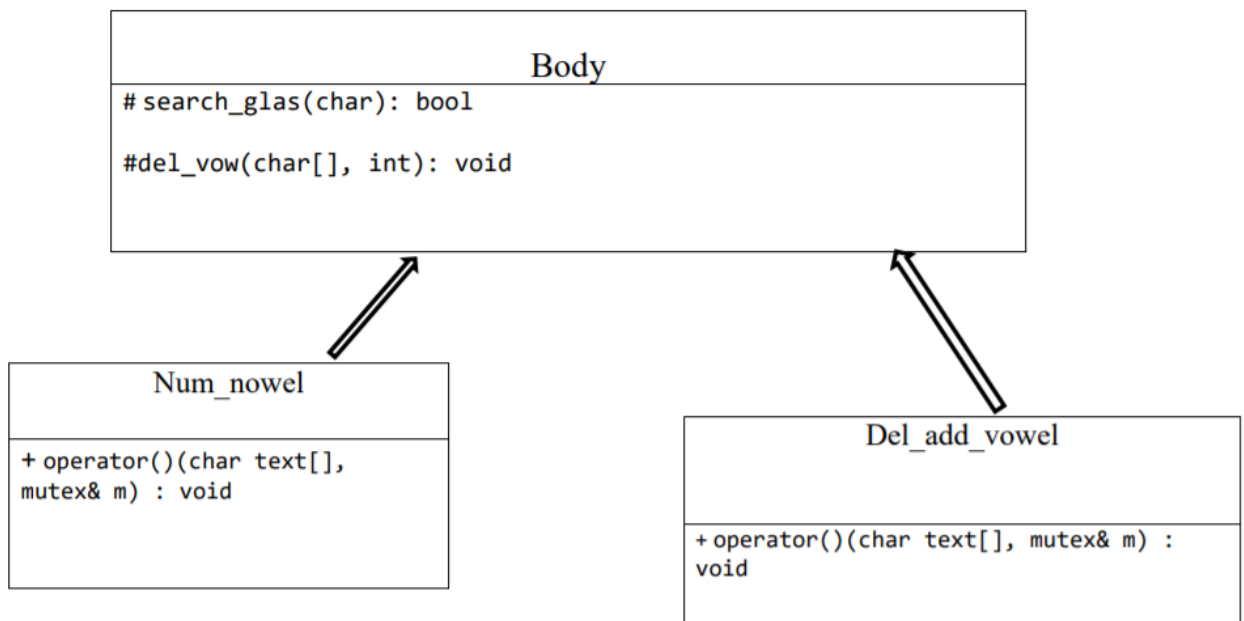
### «Потоки в C++»

**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.

### Вариант 6

#### Задание 2

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 (синхронное выполнение):

```
#include <mutex>
#include <thread>
#include <iostream>
#include <chrono>
#include <cstring>
#include <cstdlib>
#include <ctime>
#include <condition_variable>
#include <windows.h>

#define MAXSIZE 255
using namespace std;
```

```

class Body {
protected:
    bool search_glas(char);
    void del_now(char[], int);
};

bool Body::search_glas(char c) {
    switch (c) {
        case 'a':
        case 'e':
        case 'o':
        case 'u':
        case 'i':
        case 'y':
            return true;
        default:
            return false;
    }
}

class Num_nowel : public Body {
public:
    void operator()(char text[], mutex& m, condition_variable& cv) {
        for (int i = 0; i < 100; i++) {
            int count = 0;
            {
                lock_guard<mutex> lock(m);
                for (int j = 0; text[j] != '\0'; j++) {
                    if (search_glas(text[j]))
                        count++;
                }
                cout << "Число гласных равно: " << count << endl;
            }
            cv.notify_one(); // Сигнализируем о завершении итерации
            this_thread::sleep_for(chrono::seconds(1));
        }
    };

    void Body::del_now(char text[], int j) {
        for (int i = j; text[i] != '\0'; i++)
            text[i] = text[i + 1];
        text[strlen(text) - 1] = '\0';
    }

    class Del_add_vowel : public Body {
    public:
        void operator()(char text[], mutex& m, condition_variable& cv) {
            for (int i = 0; i < 100; i++) {
                {
                    unique_lock<mutex> lock(m);
                    cv.wait(lock); // Ждем сигнала о завершении итерации
                    srand(unsigned(time(NULL)));
                    int j = 0;
                    if (strlen(text) != 0) {
                        j = int(rand() % strlen(text));
                        if (search_glas(text[j])) {
                            cout << j << endl;
                            del_now(text, j);
                        } else {
                            text[j] = 'a';
                        }
                    } else {
                }
            }
        }
    };
}

```

```

        cout << "Все символы закончились" << endl;
        exit(0);
    }
    this_thread::sleep_for(chrono::seconds(1));
}
};

void show(char text[], mutex& m, condition_variable& cv) {
    for (int i = 0; i < 100; i++) {
        {
            unique_lock<mutex> lock(m);
            cout << text << endl;
        }
        cv.notify_one(); // Сигнализируем о завершении итерации
        this_thread::sleep_for(chrono::seconds(1));
    }
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(CP_UTF8);
    setlocale(LC_ALL, "Russian");

    mutex m;
    condition_variable cv;
    Del_add_vowel del_add;
    Num_nowel num;
    char text[255] = "Hello";

    cout << text << endl;

    thread threadDel_add(del_add, text, ref(m), ref(cv));
    thread threadNum_now(num, text, ref(m), ref(cv));
    thread threadShow(show, text, ref(m), ref(cv));

    threadDel_add.join();
    threadNum_now.join();
    threadShow.join();

    cout << "\n\n";
    return 0;
}

```

## Результат выполнения (синхронное выполнение)

```
Hello
Число гласных равно: 2
Helao
Число гласных равно: 3
Helao
1
Число гласных равно: 1
Hla
Число гласных равно: 2
ala
ala
Число гласных равно: 2
Число гласных равно: 3
aaa
1
a
Число гласных равно: 1
0
Число гласных равно: 0

Число гласных равно: 0

Все символы закончились
```

## Код программы (Асинхронное выполнение):

```
#include <mutex>
#include <thread>
#include <iostream>
#include <chrono>
#include <cstring>
#include <cstdlib>
#include <ctime>
#include <windows.h>

#define MAXSIZE 255
using namespace std;

class Body {
protected:
    bool search_glas(char);

    void del_now(char[], int);
};

bool Body::search_glas(char c) {
    int i = 0;
    switch (c) {
        case 'a':
        case 'e':
        case 'o':
        case 'u':
        case 'i':
        case 'y':
            return true;

        default:
            return false;
    };
}

class Num_nowel : public Body { //подсчет числа гласных в тексте
public:
    void operator()(char text[], mutex &m) {
        int j = 0;

        for (int i = 0; i < 100; i++) {
            int count = 0;
            m.lock();

            for (int j = 0; text[j] != '\0'; j++) {
                if (search_glas(text[j]))
                    count++;
            }
            m.unlock();
            this_thread::sleep_for(chrono::seconds(1));
            cout << "Число гласных равно: " << count << endl;
        }
    };
};

void Body::del_now(char text[], int j) {
    int i;
    for (i = j; text[i] != '\0'; i++)
```

```

        text[i] = text[i + 1];
        text[i] = '\0';
    }

class Del_add_vowel : public Body {
public:
    void operator()(char text[], mutex &m) {
        for (int i = 0; i < 100; i++) {
            m.lock();
            srand(unsigned(time(NULL)));
            int j = 0;
            if (strlen(text) != 0) {
                j = int(rand() % strlen(text));
                if (search_glas(text[j])) {

                    cout << j << endl;
                    del_now(text, j);
                } else {

                    text[j] = 'a';
                }
            }
            m.unlock();
            this_thread::sleep_for(chrono::seconds(1));
        } else {
            cout << "Все символы закончились" << endl;
            break;
        }
    }
};

void show(char text[], mutex &m) {

    for (int i = 0; i < 100; i++) {
        m.lock();
        cout << text << endl;
        m.unlock();
        this_thread::sleep_for(chrono::seconds(1));
    }
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(CP_UTF8);
    setlocale(LC_ALL, "Russian");

    mutex m;
    Del_add_vowel del_add;
    Num_nowel num;
    char text[255] = "Hello";

    cout << text << endl;

    thread threadDel_add(del_add, text, ref(m));
    thread threadNum_now(num, text, ref(m));
    thread threadShow(show, text, ref(m));

    threadNum_now.detach();
    threadDel_add.detach();
    threadShow.join();
}

```

```
cout << "\n\n";
return 0;
}
```

Результат выполнения (Асинхронное выполнение):

```
Hello
aello
Число гласных равно: ae1ao
3
ae1ao
Число гласных равно: 4
1
Число гласных равно: 3
0
1ao
2
Число гласных равно: 2
1a
aa
Число гласных равно: 1
Число гласных равно: 2
aa
1
a
Число гласных равно: 1
0
Число гласных равно: 0
```



### Задание 3

Реализовать классы и выполнить перегрузку оператора функтора для реализации поставленной основной задачи.

Код программы (Синхронный вывод):

```
#include <Windows.h>
#include <iostream>
#include <cstdlib>
#include <string>
#include <ctime>
#include <tchar.h>

#define MAXSIZE 255
using namespace std;

char text[MAXSIZE] = "Wizards of the Kirin Tor! I am Arthas, first of the
Lich King's death knights! I demand that you open the gates and surrender to
the might of the Scourge! ";

DWORD WINAPI thread_1(LPVOID lpParam) {
    int i = 0;
    for (int i = 0; i < 100; i++) {

        HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, _T("mutex"));
        int pos = strlen(text);
        for (int j = pos - 1; j != -1; j--) {
            text[j + 1] = text[j];
        }
        text[0] = ' ';
        text[pos + 2] = '\\0';
        i++;
        cout << text << endl;
        ReleaseMutex(mutex);
    }
    return 0;
}

DWORD WINAPI thread_2(LPVOID lpParam) {
    int i = 0;
    for (int i = 0; i < 100; i++) {
        HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, _T("mutex"));
        srand(unsigned(time(NULL)));
        int j = 0;
        if (strlen(text) != 0) {
            j = int(rand() % strlen(text));
            if (text[j] == ' ') {

                int i;
                for (i = j; text[i] != '\\0'; i++)
                    text[i] = text[i + 1];
                text[i] = '\\0';

                j = int(rand() % strlen(text));
                int pos = strlen(text);
                for (int k = pos - 1; k != j - 1; k--) {
                    text[k + 1] = text[k];
                }
                text[j] = ' ';
            }
        }
    }
}
```

```

        text[pos + 2] = '\\0';
    }

    ReleaseMutex(mutex);
}

return 0;
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(CP_UTF8);
    setlocale(LC_ALL, "Russian");

    HANDLE arr[2];
    int data_1 = 0, data_2 = 0;

    HANDLE handle_1 = CreateThread(NULL, 0, thread_1, &data_1, 0, NULL);
    HANDLE handle_2 = CreateThread(NULL, 0, thread_2, &data_2, 0, NULL);
    arr[0] = handle_1;
    arr[1] = handle_2;
    WaitForMultipleObjects(2, arr, TRUE, INFINITE);
    CloseHandle(handle_1);
    CloseHandle(handle_2);
    return 0;
}

```

Работа программы (Синхронный вывод):

[illegible]

## Код программы (Асинхронный вывод):

```
#include <iostream>
#include <thread>
#include <mutex>
#include <cstdlib>
#include <cstring>
#include <ctime>

#define MAXSIZE 255

using namespace std;

char text[MAXSIZE] = "Wizards of the Kirin Tor! I am Arthas, first of the
Lich King's death knights! I demand that you open the gates and surrender to
the might of the Scourge! ";

void thread_1() {
    for (int i = 0; i < 100; i++) {
        int pos = strlen(text);
        for (int j = pos - 1; j != -1; j--) {
            text[j + 1] = text[j];
        }
        text[0] = ' ';
        text[pos + 2] = '\\0';
        cout << text << endl;
        this_thread::yield();
    }
}

void thread_2() {
    for (int i = 0; i < 100; i++) {
        srand(unsigned(time(NULL)));
        int j = 0;
        if (strlen(text) != 0) {
            j = int(rand() % strlen(text));
            if (text[j] == ' ') {
                int k;
                for (k = j; text[k] != '\\0'; k++)
                    text[k] = text[k + 1];
                text[k] = '\\0';
                j = int(rand() % strlen(text));
                int pos = strlen(text);
                for (int l = pos - 1; l != j - 1; l--) {
                    text[l + 1] = text[l];
                }
                text[j] = ' ';
                text[pos + 2] = '\\0';
            }
        }
        this_thread::yield();
    }
}

int main() {
    thread t1(thread_1);
    thread t2(thread_2);

    t1.join();
    t2.join();

    return 0;
}
```

## Работа программы (Асинхронный вывод):

```
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
Wizards of the Kirin Tor! I am Arthas, first of the Lich King's death knights! I demand that you open the gates and surrender to the might of the Scourge!
```

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