

Задание N 44.

a) using namespace std;
const string months[12] = { "Январь", "Февраль", ..., "Декабрь" };

class Attend {

private:

string fio;

map<string, int> absent;

public:

Attend(string fio) {

this->fio = fio;

this->absent = { { "Январь", 0 }, ..., { "Декабрь", 0 } };

}

← конструктор

string getFio() {

~~return~~

return this->fio;

}

map<string, int> getAbsent() {

return this->absent;

}

Attend & operator = (Attend & second) {

this->fio = second.getFio();

this->absent = second.getAbsent();

}

1) Оператор присваивания

int& operator [](int number) {

return this->absent [months[number]];

}

2) Оператор [] для доступа

void addAbsent(int number) {

this->absent [months[number]] += 1;

}

3) метод добавления пропуска

6)

~~vector<int>~~

vector<int> v { ... };

auto iteration = [&v](int ~~x~~) {

double lower = x - (double)x / 10.0;

double higher = x + (double)x / 10.0;

~~#~~

for (~~vector<int>~~ :: iterator it = v.begin();

it != v.end(); it++) {

if (! (lower <= *it && *it <= ~~higher~~))

cout << *it << ' ';

}

};

1) umepatop

auto lambda = [&v](int x) {

double lower = x - (double)x / 10.0;

double higher = x + (double)x / 10.0;

for_each(v.begin(), v.end(), [lower, higher]

(int i) {

~~#~~

if (! (lower <= i && i <= higher))

cout << i << ' ';

});

};

2)

многообразие

```

c) mutex first;
    mutex second;

    void even() {
        for (int i = 0; i < 1000; i += 2) {
            cout << i << " (this - thread :: get-id()) << endl;
            first.lock();
            second.unlock();
        }
    }

    int main() {
        thread
        second.lock();
        thread even t(even);
        for (int i = 1; i < 1000; i += 2) {
            cout << i << " (this - thread :: get-id()) << endl;
            second.lock();
            first.unlock();
        }
        t.join();
        return 0;
    }

```