

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №8
по курсу: «Языки и методы программирования»
«Разработка шаблона класса»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Целью данной работы является изучение шаблонов классов языка C++.

2. Персональный вариант

Queue<T, N> – очередь с элементами типа T и максимальным размером N, реализованная через двойной стек и имеющая обычные для очереди операции. Queue<int, N> дополнительно имеет операцию нахождения максимального числа в очереди.

3. Решение

3.1. Код

```
#include <cassert>
#include <iostream>
#include <stack>
#include <vector>

using namespace std;

constexpr int INF = 1e9;

template <typename T, int N> class Queue {
private:
    int max_size, current_size;
    vector<T> s1, s2;

public:
    Queue() : max_size(N), current_size(0) {}

    void enqueue(T value) {
        assert(current_size != max_size);
        s1.push_back(value);
        current_size++;
    }

    T dequeue(void) {
        if (s2.size() == 0) {
            while (!s1.empty()) {
                s2.push_back(s1.back());
                s1.pop_back();
            }
        }

        current_size--;
```

```

        auto res = s2.back();
        s2.pop_back();
        return res;
    }

    bool empty() { return s1.size() + s2.size() == 0; }
};

template <int N> class Queue<int, N> {
private:
    int max_size;
    int current_size;

    vector<pair<int, int>> s1, s2;

    void push(vector<pair<int, int>> &s, int value) {
        int prev = s.size() ? s.back().second : -INF;
        s.emplace_back(value, max(prev, value));
    }

    int pop(vector<pair<int, int>> &s) {
        if (s.empty())
            return -INF;
        int res = s.back().first;
        s.pop_back();
        return res;
    }

public:
    Queue() : max_size(N), current_size(0) {}

    int maximum() {
        return max(s1.size() ? s1.back().second : -INF,
            ↪ s2.size() ? s2.back().second : -INF);
    }

    void enqueue(int value) {
        assert(current_size != max_size);
        current_size++;
        push(s1, value);
    }

    int dequeue(void) {
        if (s2.empty())

```

```

        while (!s1.empty())
            push(s2, pop(s1));

        if (s2.size() > 0)
            return pop(s2);

        current_size--;

        return -INF;
    }

    bool empty() { return s1.size() + s2.size() == 0; }
};

int main() {
    Queue<int, 5> q;
    q.enqueue(12);
    q.enqueue(15);
    q.enqueue(5);
    q.enqueue(-5);
    q.enqueue(14);
    // q.enqueue(5);
    cout << q.maximum() << '\n';
    cout << q.dequeue() << '\n';
    cout << q.maximum() << '\n';
    cout << q.dequeue() << '\n';
    cout << q.maximum() << '\n';

    Queue<string, 4> q1;
    q1.enqueue("Hello, world");
    q1.enqueue("Test string");
    q1.enqueue("lkjdkfksd");
    cout << q1.dequeue() << '\n';
    cout << q1.dequeue() << '\n';
    cout << q1.dequeue() << '\n';

    return 0;
}

```

Код 1: main.cc

3.2. Скриншоты

```
15  
12  
15  
15  
14  
Hello, world  
Test string  
lkjdkfksd  
  
Process finished with exit code 0
```

Рис. 1: Пример работы программы