Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(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»

Вариант 7

Студент группы ИУ9-21Б Шиятов Н.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Необходимо составить шаблон класса, перегрузив указанные операции. Проверку работоспособности класса требуется организовать в функции main, размещённой в файле «main.cpp».

PtrQueue<T> – очередь указателей на структуры типа T, реализованная через кольцевой буфер. Операции, которые должны быть перегружены для PtrQueue<T>:

1. «≪» – добавление указателя на в очередь (enqueue);
2. «!» – вытаскивание указателя из очереди (dequeue);
3. empty – проверка на пустоту очереди;
4. унарный «∗ » – возвращает значение, адрес которого лежит в начале очереди (туда указывает head);
5. «− >» – осуществляет доступ к полям структуры, адрес которой лежит в начале очереди.

2 Результаты

Исходный код программы представлен в листингах 1– 2.

Результат запуска представлен на рисунке 1.

Листинг 1 — Класс PtrQueue

```
template<typename T>
class PtrQueue {
private:
    T **data;
    int head, tail, max_size;
public:
    PtrQueue(int size) {
        data = new T*[size];
        max_size = size;
        head = 0;
        tail = 0;
    }

    bool empty() {
        return head == tail;
    }

    friend PtrQueue<T>& operator<<(PtrQueue<T> &queue, T *ptr) {
        queue.data[queue.tail] = ptr;
        queue.tail = (queue.tail + 1) % queue.max_size;
        return queue;
    }

    friend T* operator!(PtrQueue<T> &queue) {
        T *ptr = queue.data[queue.head];
        queue.head = (queue.head + 1) % queue.max_size;
        return ptr;
    }

    T* operator*() {
        if (empty()) {
            return 0;
        }
        return data[head];
    }

    T* operator->() {
        if (empty()) {
            return 0;
        }
        return data[head];
    }
};
```

Листинг 2 — Проверка работоспособности

```
#include "PtrQueue.cpp"

#include <iostream>
using namespace std;

struct Point {
    int x;
    int y;
};

int main() {
    PtrQueue<Point> queue1(3);
    PtrQueue<Point> queue2(3);

    Point point1 = {1, 2};
    Point point2 = {7, 8};

    Point point3 = {5, 6};
    Point point4 = {9, 0};

    queue1 << &point1;
    queue1 << &point2;

    queue2 << &point3;
    queue2 << &point4;


    Point *p1 = *queue1;
    Point *p2 = !queue1;
    Point *p3 = *queue2;
    Point *p4 = !queue1;


    cout << "p1->x = " << p1->x << endl;
    cout << "queue2->x = " << queue2->x << endl;
    cout << "p2->x = " << p2->x << endl;
    cout << "p3->x = " << p3->x << endl;
    cout << "p4->x = " << p4->x << endl;

    return 0;
}
```

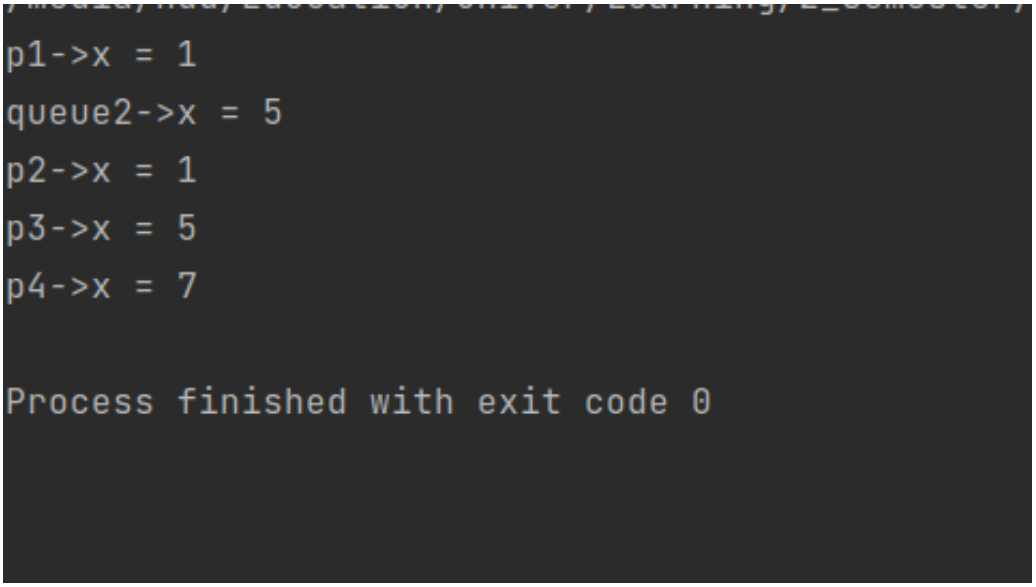

Рис. 1 — Результат