

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК  
  
Кафедра прикладной математики, механики, управления и программного обеспечения**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6 ПО ДИСЦИПЛИНЕ  
«СТРУКТУРЫ И АЛГОРИТМЫ КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ»**

Направление 02.03.03 «Математическое обеспечение и администрирование  
информационных систем»

* Выполнил студент гр. Б8204  
  Проскурин Денис Александрович  
  Проверил:  
  Доцент, к.т.н С.Н.Остроухова

Владивосток 2019

**Неформальная постановка задачи**

Реализовать класс циклическая очередь с приоритетным выходом. Хвост указывает на последний добавленный элемент. Класс должен содержать следующие методы:

1. Конструктор
2. Деструктор
3. Добавить элемент
4. Удалить элемент
5. Неразрушающее чтение
6. Длина(текущее кол-во)
7. Печать (от left до right)
8. Перегрузка присваивания
9. Конструктор копирования

**Спецификация методов класса**

**class Circle\_queue\_exit** {

private:

struct element \*vector;//вектор записей вида статус, данные

unsigned int max\_l;//максимальная длина

unsigned int right, left;//правая, левая границы

void move\_left(const unsigned int pos);//сдвинуть вектор влево

void move\_right(const unsigned int pos); //сдвинуть вектор вправо

public:

Circle\_queue\_exit(unsigned int size);//конструктор

Circle\_queue\_exit(const Circle\_queue\_exit &obj);//конструктор копирования

~Circle\_queue\_exit();//дестуктор

unsigned int add(element);//добавить элемент в очередь

unsigned int del();//удалить приоритетный элемент

element read(unsigned int deep);//неразрушающее чтение

unsigned int length();//текущая длина

unsigned int print();//печать всей очереди

void operator =(Circle\_queue\_exit&);//перегрузка оператора =

}

**Circle\_queue\_exit(unsigned int size)** — конструктор,создающий очередь максимальным размером size. Если начальный размер задан некорректно создаётся циклическая очередь размером 10.

**~Circle\_queue\_exit()** — деструктор.

**unsigned int add(element)** — добавить элемент в конец очереди. Возвращает 0 если операция прошла успешно. Возвращает 1 если очередь переполнена.

**unsigned int del()** - удаление элемента с наивысшим приоритетом. Возвращает 0 , если операция прошла успешно. Возвращает 2, если очередь пуста.

**element read() —** возвращает элемент c самым большим приоритетом. Возвращает элемент empty, если очередь пуста

**unsigned int length()**-возвращает текущий размер очереди.

**unsigned int print()**-выводит на экран всю очередь от левой до правой границы.

**void operator =(Circle\_queue\_exit&)**-левой очереди присваивает данные правой очереди. Если левая очередь меньше правой, присваивается возможное количество элементов. Возможно множественное присваивание.

**Тесты**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **Тестирование конструктора класса** | |
| Circle\_queue\_exit example(5); | Head Tail   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | |
| Circle\_queue\_exit example(0); | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| **Тестирование конструктора-копирования класса** | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 1H | 2I | 3J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit example2(example); | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 1H | 2I | 3J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1A | 2B | 3C | 1H | 2I | 3J |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //Circle\_queue\_exit example2(example); | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1A | 2B | 3C | 1H | 2I | 3J |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   //Circle\_queue\_exit example2(example); | Head Tail   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| **Тестирование метода add** | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //добавление 10 элементов | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Коды возврата: 0 0 0 0 0 0 0 0 0 0 |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //добавление элементов больше максимальной размерности нашей очереди | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Коды возврата: 0 0 0 0 0 0 0 0 0 0 1 1 1 |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | 1A | 1H | 1V | 2W | 3X | 4Y | 5Z |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //Добавление 3 элементов в нашу очередь | Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1[ | 2\ | 3] | ]A | 1H | 1V | 2W | 3X | 4Y | 5Z |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(10);  Head  Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | 1J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 2K |  |  |  |  |  |  |  |  | 1J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| **Тестирование метода del** | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 9B | 8C | 7D | 6E | 5F | 4G | 3H | 2I | 1J | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление всех элементов | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Коды возврата: 0 0 0 0 0 0 0 0 0 0 |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление 3 элементов из пустой очереди | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Коды возврата:2 2 2 |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 1H | 2I | 3J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление 3 элементов со сдвигом хвоста влево | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 1H | 2I | 3J |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|  |  |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 1H | 2I | 5J |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление элемента с хвоста | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 1H | 2I |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 1H | 2I |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление элемента со сдвигом головы вправо | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1A | 2B | 3C | 1H | 2I |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 1H | 2I |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление 2 элементов со сдвигом головы вправо | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | 1A | 1H | 2I |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0T | 0U | 0V | 0W |  |  |  |  |  | 1J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление 2 элементов со сдвигом головы вправо | Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 0U | 0V | 0W |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 2B |  |  |  |  |  |  |  |  | 1A |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление элемента со сдвигом хвоста влево | Head  Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | 1A |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1B |  |  |  |  |  |  |  |  | 2A |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //удаление элемента со сдвигом головы вправо | Head  Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1B |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| **Тестирование метода read** | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 0empty |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Коды возврата: 0 0 0 0 0 0 0 0 0 0 | 10J |
| **Тестирование метода length** | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 0 |
| Tail Head   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0T | 0U | 0V | 0W |  |  |  |  |  | 1J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 5 |
| **Тестирование метода print** | |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | Код возврата 2 |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | head=0 tail=9  |1A|2B|3C|4D|5E|6F|7G|8H|9I|10J  |#0|#1|#2|#3|#4|#5|#6|#7|#8|#9 |
| Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | 4G | 3H | 2I | 1J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | head=6 tail=9  |4G|3H|2I|1J  |#6|#7|#8|#9 |
| **Тестирование метода =** | |
| Circle\_queue\_exit example(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 10A | 9B | 8C | 7D | 6E | 5F | 4G | 3H |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   Circle\_queue\_exit example2(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //example2=example | //example2 имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 10A | 9B | 8C | 7D | 6E | 5F | 4G | 3H |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 10A | 9B | 8C | 7D | 6E | 5F | 4G | 3H |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   Circle\_queue\_exit example2(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //example=example2 | //example имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit example2(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   //присваивание возможного колличества элементов наименьшей очереди | //example2 имет вид  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit exampl2(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit example3(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   //множественное присваивание  //example3=example2=example | // exampl2 имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   // exampl3 имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit exampl2(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit example3(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   //множественное присваивание  //example=example2=example3 | //example2 имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   //example имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Circle\_queue\_exit example(10);  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H | 9I | 10J |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   Circle\_queue\_exit exampl2(8);  Head Tail   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E | 6F | 7G | 8H |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |   Circle\_queue\_exit example3(5);  Head Tail   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E |  | | 0 | 1 | 2 | 3 | 4 | 5 |   //множественное присваивание  //example=example2=example3 | //example имеет вид  Head Tail   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1A | 2B | 3C | 4D | 5E |  |  |  |  |  |  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |