|  |
| --- |
| 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** |
|  |
|  |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**  **ПО ДИСЦИПЛИНЕ СТРУКУРЫ И АЛГОРИТМЫ КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ**  По направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем» |
| |  |  | | --- | --- | |  | Выполнил студент гр. Б8204  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Г.М. Сысоев  Проверил доцент, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Н. Остроухова |   г. Владивосток |
| 2018 |

**СПЕЦИФИКАЦИЯ**

**1. Спецификация шаблонного класса**

**template** <**class** T>  
**class** RingDeque {  
**private**:  
 std:: string name; // имя кольцевого дека  
 **int** dimension = 3; // размер дека поумолчанию равный 3  
 **int** leftEnter, rightEnter; // левая правая граница дека  
 T\* arr; // массив в котором хранятся элементы дека  
**public**:  
 RingDeque(**int** dim, std::string n); // конструктор класса  
 RingDeque(**const** RingDeque<T> &a); *//конструктор копирования* ~RingDeque();  
 **int** PushBack(T a); *//запись слева***int** PushFront(T a);*//запись справа* **int** PopBack(); *//получение элемента слева***int** PopFront(); *//получение элемента справа* T\* PeekBack(); *//чтение слева* T\* PeekFront(); *//чтение справа* **void** print(); *// печать* **int** getLength();*// получение длины дека (количества элементов добавленных в дек)* RingDeque& **operator**=(RingDeque&); // перегрузка оператора присваивания для кольцевых деков  
};

**2. Спецификация функций**

RingDeque<T>::RingDeque(**int** dim, std::string n); - конструктор класса, необходимый для создания объекта RingDeque<T>. Аргументы при вызове – dim – размерность массива в котором будут храниться элементы дека. По умолчанию равный 4. Если dim больше 4, то создаваемый объект примет значение dim + 1 для размерности. n – строка, которая будет являться именем у создаваемого дека. (Размерность задаётся на +1 больший элемент, т.к. дек хранит также опорный элемент)

RingDeque(**const** RingDeque<T> &a); - конструктор копирования класса. При вызове конструктора класса, как аргумент передаётся константная ссылка на RingDeque &a (объект, с которого будет происходить копирование). Конструктор копирования создаёт объект (дек) со значениями из полей RingDeque a.

~RingDeque(); - деструктор класса. Удаляет объект RingDeque<T>, который вызвал этот метод.

**int** PushBack(T a); - метод добавления по правой границе в объект RingDeque<T> элемента T a. В процессе добавления, метод вызывает метод RingDeque<T>.getLength(). В случае добавления функция возвращает 0. В случае если дек (размер дека == размерность – 1) заполнен возвращает -301.

**int** PushFront(T a); - метод добавления по левой границе в объект RingDeque<T> элемента T a. В процессе добавления, метод вызывает метод RingDeque<T>.getLength(). В случае добавления функция возвращает 0. В случае если дек заполнен (размер дека == размерность – 1) возвращает -301.

**int** PopBack(); - метод удаления по правой границе элемента T из RingDeque<T>. В случае, если дек пуст (Если (левая граница – правая граница == 1) или (правая граница – левая граница == размерность – 1)) функция вернёт -315. В случае, если функция удалит элемент, вернётся 0.

**int** PopFront(); - метод удаления по левой границе элемента T из RingDeque<T>. В случае, если дек пуст(Если (левая граница – правая граница == 1) или (правая граница – левая граница == размерность – 1)) функция вернёт -315. В случае, если функция удалит элемент, вернётся 0.

T\* PeekBack(); - метод чтения элемента из дека по правой границе. Возвращает элемент по правой границе.

T\* PeekFront(); - метод чтения элемента из дека по правой границе. Возвращает элемент по левой границе.

**void** print(); - метод производит вывод всех элементов дека на экран. Если дек пуст, выведет сообщение об этом.

**int** getLength(); - метод возвращающий длину дека.

**3. Тесты**

|  |  |  |
| --- | --- | --- |
| № Теста и название теста | Входные данные | Выходные данные |
| 1. **int** getLength(); | RingDeque<**int**> first(6, **"WHEEL"**); |  |
| 1.1) пустой дек | 0 0 0 0 0 0 0  R L | 0 |
| 1.2) заполненный дек | 1 1 1 1 1 0 1  R L | 6 |
| 1.3) левая граница больше правой | 1 1 1 0 0 0 1  R L | 4 |
| 1.4) правая граница больше левой | 0 0 0 1 1 0 0  L R | 2 |
| 2. **int** PushBack(T a); | -//-  PushBack(4); |  |
| 2.1) добавление в заполненный дек | 1 1 1 1 1 0 1  R L | -301 |
| 2.2) добавление в пустой дек | 0 0 0 0 0 0 0  R L | 0 0 0 1 0 0 0  RL (указывают на один элемент) |
| 2.3) переход при добавлении на начало массива | 0 0 0 1 1 1 1  L R | 1 0 0 1 1 1 1  R L |
| 3. **int** PushFront(T a); |  |  |
| 3.1) добавление в заполненный дек | 1 1 1 1 1 0 1  R L | -301 |
| 3.2) добавление в пустой дек | 0 0 0 0 0 0 0  R L | 0 0 1 0 0 0 0  RL (указывают на один элемент) |
| 3.3) переход при добавлении на конец массива | 1 1 1 0 0 0 0  L R | 1 1 1 0 0 0 1  R L |
| 4. **int** PopBack(); |  |  |
| 4.1) переход правой границы при удалении на конец массива | 1 0 0 1 1 1 1  R L | 0 0 0 1 1 1 1  L R |
| 4.2) удаление единственного элемента | 0 0 0 1 0 0 0  RL (указывают на один элемент) | 0 0 0 0 0 0 0  R L |
| 4.3) удаление элемента из пустого дека | 0 0 0 0 0 0 0  R L | -315 |
| 5. **int** PopFront(); |  |  |
| 5.1) переход левой границы при удалении на начало массива | 1 1 1 0 0 0 1  R L | 1 1 1 0 0 0 0  L R |
| 5.2)удаление единственного элемента | 0 0 1 0 0 0 0  RL (указывают на один элемент) | 0 0 0 0 0 0 0  R L |
| 5.3) удаление элемента из пустого дека | 0 0 0 0 0 0 0  R L | -315 |
| 6. Оператор присваивания |  |  |
| 6.1) Присваивание большего меньшему | 45 40 35 30 25 20 15 10 5 0  0 0 0 0 0 0 | 45 40 35 30 25 20 |
| 6.1) Присваивание меньшему большего | 25 20 15 10 5 0  0 0 0 0 0 0 0 0 0 0 | 25 20 15 10 5 0 |
| 7. Конструктор копирования |  |  |
| 7.1 Копирование пустого дека | 0 0 0 0 0 0 0 0 0 | Ring deque WHEEL.copy is empty. |
| 7.2 Копирование полного дека | 45 40 35 30 25 20 15 10 5 0  0 0 0 0 0 0 0 0 0 0 | 45 40 35 30 25 20 15 10 5 0 |