**Лабораторная работа №1**

**«Стеки и очереди»**

**Задание №1.**

Напишите метод класса Queue программы queue.java (см. листинг 4.4) для

вывода содержимого очереди. Учтите, что задача не сводится к простому выводу содержимого базового массива. Содержимое очереди должно выводиться от первого вставленного элемента до последнего, а пользователь не должен видеть, что последовательность прерывается на границе массива. Будьте внимательны и проследите за тем, чтобы один элемент и содержимое пустой очереди выводились корректно независимо от положения front и rear.

Листинг 4.4. Программа queue.java

*// queue.java*

*// Работа с очередью*

*// Запуск программы: C>java QueueApp*

*////////////////////////////////////////////////////////////////*

**class** Queue

{

**private** **int** maxSize;

**private** **long**[] queArray;

**private** **int** front;

**private** **int** rear;

**private** **int** nItems;

*//--------------------------------------------------------------*

**public** Queue(**int** s) *// Конструктор*

{

maxSize = s;

queArray = **new** **long**[maxSize];

front = 0;

rear = -1;

nItems = 0;

}

*//--------------------------------------------------------------*

**public** **void** insert(**long** j) *// Вставка элемента в конец очереди*

{

**if**(rear == maxSize-1) *// Циклический перенос*

rear = -1;

queArray[++rear] = j; *// Увеличение rear и вставка*

nItems++; *// Увеличение количества элементов*

}

*//--------------------------------------------------------------*

**public** **long** remove() *// Извлечение элемента в начале очереди*

{

**long** temp = queArray[front++]; *// Выборка и увеличение front*

**if**(front == maxSize) *// Циклический перенос*

front = 0;

nItems--; *// Уменьшение количества элементов*

**return** temp;

}

*//--------------------------------------------------------------*

**public** **long** peekFront() *// Чтение элемента в начале очереди*

{

**return** queArray[front];

}

*//--------------------------------------------------------------*

**public** **boolean** isEmpty() *// true, если очередь пуста*

{

**return** (nItems==0);

}

*//--------------------------------------------------------------*

**public** **boolean** isFull() *// true, если очередь заполнена*

{

**return** (nItems==maxSize);

}

*//--------------------------------------------------------------*

**public** **int** size() *// Количество элементов в очереди*

{

**return** nItems;

}

*//--------------------------------------------------------------*

} *// Конец класса Queue*

*////////////////////////////////////////////////////////////////*

**class** QueueApp

{

**public** **static** **void** main([String](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string)[] args)

{

Queue theQueue = **new** Queue(5); *// Очередь из 5 ячеек*

theQueue.insert(10); *// Вставка 4 элементов*

theQueue.insert(20);

theQueue.insert(30);

theQueue.insert(40);

theQueue.remove(); *// Извлечение 3 элементов*

theQueue.remove(); *// (10, 20, 30)*

theQueue.remove();

theQueue.insert(50); *// Вставка еще 4 элементов*

theQueue.insert(60); *// (с циклическим переносом)*

theQueue.insert(70);

theQueue.insert(80);

**while**( !theQueue.isEmpty() ) *// Извлечение и вывод*

{ *// всех элементов*

**long** n = theQueue.remove(); *// (40, 50, 60, 70, 80)*

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.print(n);

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.print(" ");

}

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.println("");

}

} *// Конец класса QueueApp*

**Задание №2.**

Создайте класс Deque по описанию деков (двусторонних очередей). Класс должен содержать методы insertLeft(), insertRight(), removeLeft(), removeRight(), isEmpty() и isFull(). Также в нем должна быть реализована поддержка циклического переноса индексов, по аналогии с очередями.

**Задание №3.**

Напишите реализацию стека на базе класса Deque из п. 4.2. Класс стека должен поддерживать те же методы и возможности, что и класс StackX в программе stack.java (см. листинг 4.1).

Листинг 4.1. Программа stack.java

*// stack.java*

*// Работа со стеком*

*// Запуск программы: C>java StackApp*

*////////////////////////////////////////////////////////////////*

**class** StackX

{

**private** **int** maxSize; *// Размер массива*

**private** **long**[] stackArray;

**private** **int** top; *// Вершина стека*

*//--------------------------------------------------------------*

**public** StackX(**int** s) *// Конструктор*

{

maxSize = s; *// Определение размера стека*

stackArray = **new** **long**[maxSize]; *// Создание массива*

top = -1; *// Пока нет ни одного элемента*

}

*//--------------------------------------------------------------*

**public** **void** push(**long** j) *// Размещение элемента на вершине стека*

{

stackArray[++top] = j; *// Увеличение top, вставка элемента*

}

*//--------------------------------------------------------------*

**public** **long** pop() *// Извлечение элемента с вершины стека*

{

**return** stackArray[top--]; *// Извлечение элемента, уменьшение top*

}

*//--------------------------------------------------------------*

**public** **long** peek() *// Чтение элемента с вершины стека*

{

**return** stackArray[top];

}

*//--------------------------------------------------------------*

**public** **boolean** isEmpty() *// True, если стек пуст*

{

**return** (top == -1);

}

*//--------------------------------------------------------------*

**public** **boolean** isFull() *// True, если стек полон*

{

**return** (top == maxSize-1);

}

*//--------------------------------------------------------------*

} *// Конец класса StackX*

*////////////////////////////////////////////////////////////////*

**class** StackApp

{

**public** **static** **void** main([String](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string)[] args)

{

StackX theStack = **new** StackX(10); *// Создание нового стека*

theStack.push(20); *// Занесение элементов в стек*

theStack.push(40);

theStack.push(60);

theStack.push(80);

**while**( !theStack.isEmpty() ) *// Пока стек не станет пустым*

{ *// Удалить элемент из стека*

**long** value = theStack.pop();

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.print(value); *// Вывод содержимого*

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.print(" ");

}

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).out.println("");

}

} *// Конец класса StackApp*

*///////////////////////////////////////////////////////////////*