Deque

Интерфейс **Deque** определяет «двунаправленную» очередь и, соответственно, методы доступа к первому и последнему элементам двусторонней очереди. Методы обеспечивают удаление, вставку и обработку элементов. Каждый из этих методов существует в двух формах. Одни методы создают исключительную ситуацию в случае неудачного завершения, другие возвращают какое-либо из значений (**null** или **false** в зависимости от типа операции). Вторая форма добавления элементов в очередь сделана специально для реализаций **Deque**, имеющих ограничение по размеру. В большинстве реализаций операции добавления заканчиваются успешно.

В следующим примере реализована работа с интерфейсом **Deque**. Методы **addFirst()**, **addLast()** вставляют элементы в начало и в конец очереди соответственно. Метод **add()** унаследован от интерфейса **Queue** и абсолютно аналогичен методу **addLast()** интерфейса **Deque**.

```
/* пример # 8 : демонстрация Deque : DequeRunner.java */
package chapt10;
import java.util.*;

public class DequeRunner {
      public static void printDeque(Deque<?> d){
                  for (Object de : d)
                        System.out.println(de + "; ");
            }
      public static void main(String[] args) {
            Deque<String> deque = new ArrayDeque<String>();
                  deque.add(new String("5"));
                  deque.addFirst("A");
                  //deque.addLast(new Integer(5));//ошибка компиляции
                  System.out.println(deque.peek());
                  System.out.println("Before:");
                  printDeque(deque);
                  deque.pollFirst();
                  System.out.println(deque.remove(5));
                  System.out.println("After:");
                  printDeque(deque);
      }
}
```

В результате на консоль будет выведено:

```
A
Before:
A;
5;
false
After:
5;
```