

# Lock-free контейнер

Необходимо реализовать в lock-free стиле следующий интерфейс:

```
/**  
 * Lock-Free множество.  
 * @param <T> Тип ключей  
 */  
public interface LockFreeSet<T extends Comparable<T>> {  
    /**  
     * Добавить ключ к множеству  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @param value значение ключа  
     * @return false если value уже существует в множестве, true если элемент  
был добавлен  
     */  
    boolean add(T value);  
  
    /**  
     * Удалить ключ из множества  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @param value значение ключа  
     * @return false если ключ не был найден, true если ключ успешно удален  
     */  
    boolean remove(T value);  
  
    /**  
     * Проверка наличия ключа в множестве  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @param value значение ключа  
     * @return true если элемент содержится в множестве, иначе - false  
     */  
    boolean contains(T value);  
  
    /**  
     * Проверка множества на пустоту  
     *  
     * Алгоритм должен быть wait-free (достаточно lock-free, wait-free для сильно  
уверенных в себе)  
     *  
     * @return true если множество пусто, иначе - false  
     */  
    boolean isEmpty();
```

}

Дополнительные условности:

1. Имя класса реализации - *LockFreeSetImpl*
2. Класс должен иметь конструктор без параметров

From:  
<http://wiki.osll.ru/> - Open Source & Linux Lab

Permanent link:  
[http://wiki.osll.ru/doku.php/courses:high\\_performance\\_computing:lock\\_free?rev=1515167389](http://wiki.osll.ru/doku.php/courses:high_performance_computing:lock_free?rev=1515167389)

Last update: **2018/01/05 18:49**

