Assignment 4

Языки для решения задачи: Java, Kotlin

Lock-Free контейнер

Реализуйте LockFreeSet. Использовать контейнеры из java.util.concurrent запрещено

Lock Free Set

```java
/**
 * Lock-Free множество.
 * @param <T> Тип ключей
 */
public interface Set<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 wait-free
     *
     * @param value значение ключа
```

```
35          * @return true если элемент содержится в множестве, иначе - false
36          */
37         boolean contains(T value);
38
39
40         /**
41          * Проверка множества на пустоту
42          *
43          * Алгоритм должен быть как минимум lock-free
44          *
45          * @return true если множество пусто, иначе - false
46          */
47         boolean isEmpty();
48     }
```

Тестирование

1. Для разработанных алгоритмов необходимо написать тесты. Покрытие тестами должно превышать 70%.