

Coroutines Map

Необходимо реализовать следующий интерфейс lock-free map в стиле coroutines:

```
public interface CoroutinesMap<K: Comparable<K>, V> {  
    /**  
     * Добавить (key, value) к ассоциативному контейнеру  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @param key ключ  
     * @param value значение  
     * @return вернуть существующее value если key уже существует в множестве,  
     null если элемент был добавлен  
     */  
    suspend fun put(key: K, value: V) : V?  
  
    /**  
     * Проверка наличия ключа в множестве  
     *  
     * Алгоритм должен быть как минимум wait-free  
     *  
     * @param key значение ключа  
     * @return значение если элемент содержится в множестве, иначе - null  
     */  
    suspend fun get(key: K) : V?  
  
    /**  
     * Удалить ключ из ассоциативного контейнера  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @param key значение ключа  
     * @return вернуть существующее value если key существовал в множестве, null  
     если элемента не было в контейнере  
     */  
    suspend fun remove(key: K) : V?  
  
    /**  
     * Проверка ассоциативного контейнера на пустоту  
     *  
     * Алгоритм должен быть как минимум lock-free  
     *  
     * @return true если множество пусто, иначе - false  
     */  
    suspend fun isEmpty(): Boolean  
  
    /**  
     * Возвращает lock-free Set элементов для ассоциативного контейнера
```

```
*  
* @return новый экземпляр Set для ассоциативного контейнера  
*/  
suspend fun entrySet(): Set<Map.Entry<K, V>>  
}
```

Имя класса реализации CoroutinesMapImpl.

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

Permanent link:
http://wiki.osll.ru/doku.php/courses:high_performance_computing:coroutines_map?rev=1591096860

Last update: 2020/06/02 14:21

