ConcurrentMap

ConcurrentMap

- The java.util.concurrent.ConcurrentMap interface represents a Map which is capable of handling concurrent access (puts and gets) to it.
- The ConcurrentMap has a few extra atomic methods in addition to the methods it inherits from its superinterface, java.util.Map.
- Since ConcurrentMap is an interface, you need to use one of its implementations in order to use it. The java.util.concurrent package contains the following implementations of the ConcurrentMap interface:
  - ConcurrentHashMap
- java.util.concurrent.ConcurrentMap interface parallel dasturda ishlatish (put va get va hk… ammalarni bajratish)mumkin bo'lgan Map ni taqdim etadi.
- ConcurrentMap java.util.Map dan nasil olgan va qo'shimcha tarzda atomic methodlarni taqdim etadi.
- ConcurrentMap bu interface va siz undan implemt olgan class lardan bittasini ishlatishingiz kerak. java.util.concurrent package da undan implement olgan class lar bor.
  - ConcurrentHashMap

ConcurrentHashMap

- The **ConcurrentHashMap** class is introduced in JDK 1.5 belongs to **java.util.concurrent** package, which implements ConcurrentMap as well as to Serializable interface also.
- ConcurrentHashMap is an enhancement of HashMap as we know that while dealing with Threads in our application HashMap is not a good choice because performance-wise HashMap is not up to the mark.
- **ConcurrentHashMap classi java.util.concurrent package da joylashgan bo'lib u JDK ni 1.5 chi versiyasida tagdim etilgan. U** ConcurrentMap va Serializable dan implements olgan.
- ConcurrentHashMap bu HashMap ni kengaytmasi hisoblanadi. Biz bilamizki Thread lar bilan ishlayotganda HashMap yaxshi tanlov emas, HashMap Thread-Save emsligi uchun.

Key points of ConcurrentHashMap 1:

- The underlined data structure for ConcurrentHashMap is Hashtable.
- ConcurrentHashMap class is thread-safe i.e. multiple threads can operate on a single object without any complications.
- At a time any number of threads are applicable for a read operation without locking the ConcurrentHashMap object which is not there in HashMap.
- In ConcurrentHashMap, the Object is divided into a number of segments according to the concurrency level.

- ConcurrentHashMap HashTable ga asoslangan.
- ConcurrentHashMap classi thread-safe. Shu sabandan birnechta thread lar hechqanday muommosiz bitta ConcurrentHashMap ob’ektini ishlatishi mumkin.
- Bir vaqtning o’zida ConcurrentHashMap ob’ektini lock/quluf lamasdan turib birnechta Thread lar ishlatishi mumkin. Bu imkoniyat HashMap mavjut emas.
- ConcurrentHashMap-da Ob'ekt parallellik darajasiga ko'ra bir qancha segmentlarga bo'linadi.

Key points of ConcurrentHashMap 2:

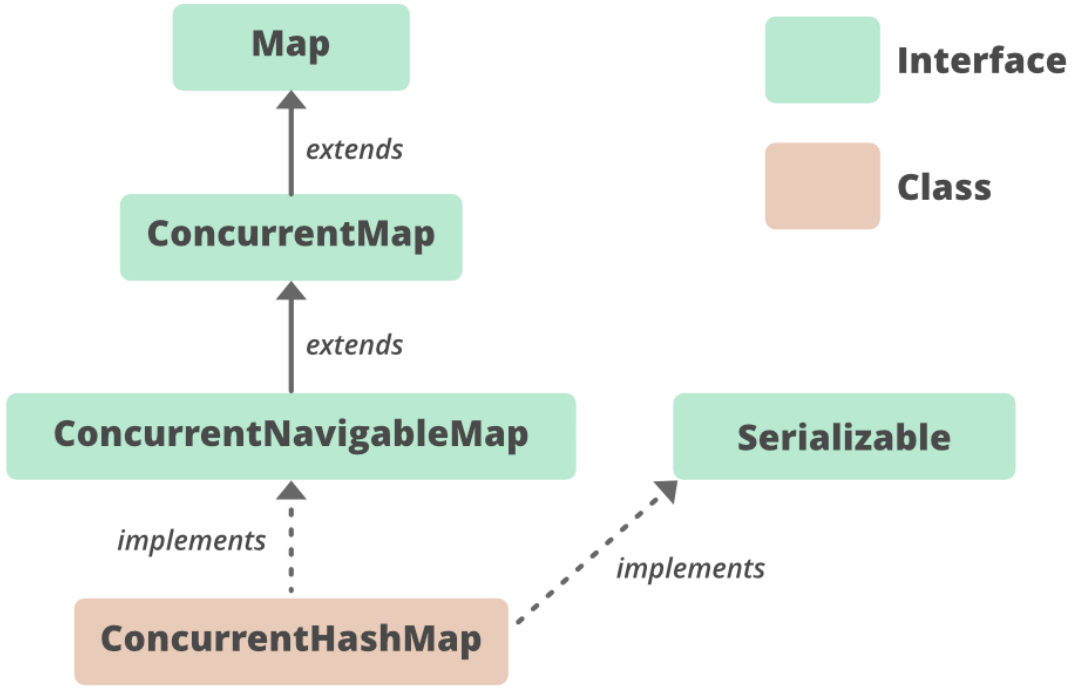
- The default concurrency-level of ConcurrentHashMap is 16.
- In ConcurrentHashMap, at a time any number of threads can perform retrieval operation.
- But for updated in the object, the thread must lock the particular segment in which the thread wants to operate.
- This type of locking mechanism is known as **Segment locking or bucket locking**. Hence at a time, 16 update operations can be performed by threads.
- Inserting null objects is not possible in ConcurrentHashMap as a key or value.

- ConcurrentHashMap ning parallellik darajasi 16 ga teng.
- ConcurrentHashMap da bir vaqtning o'zida istalgan miqdordagi Thread lar qidirish/get amallarini bajarishi mumkin.
- Ammo update amalida Thread update qilmoqchi bo'lgan bo'lagini lock/quluf ga olishi kerak.
- Lock/Quluf ga olish maxanizmi Segment locking yoki bucket locking deb nomlangna. Shu sabadan Bir vaqtni o'zida Thread lar tomonidan 16 ta update amali bajarilishi mumkin.
- ConcurrentHashMap da null qilmay ni key yoki value sifatida insert qilish mumkin emas.

ConcurrentHashMap vs HashTable

- The ConcurrentHashMap is very similar to the java.util.HashTable class, except that ConcurrentHashMap offers better concurrency than HashTable does. ConcurrentHashMap does not lock the Map while you are reading from it. Additionally, ConcurrentHashMap does not lock the entire Map when writing to it. It only locks the part of the Map that is being written to, internally.
- Another difference is that ConcurrentHashMap does not throw ConcurrentModificationException if the ConcurrentHashMap is changed while being iterated. The Iterator is not designed to be used by more than one thread though.

ConcurrentHashMap Hierarchy



ConcurrentHashMap Example 1

- Thread Class

```
public class MyThread implements Runnable {

    private ConcurrentHashMap<Integer, Integer> map;

    public MyThread(ConcurrentHashMap<Integer, Integer> map) {
        this.map = map;
    }

    @Override
    public void run() {
        for (int i = 0; i < 10; i++) {
            int n = ThreadLocalRandom.current().nextInt(100);
            map.put(n, n);
        }
    }
}
```

ConcurrentHashMap Example 2

- Main Class

```
public class ConcurrentHashMapDemoMain {
    public static void main(String[] args) throws InterruptedException {
        ConcurrentHashMap<Integer, Integer> map = new ConcurrentHashMap<>();
        new Thread(new MyThread(map)).start();
        new Thread(new MyThread(map)).start();
        new Thread(new MyThread(map)).start();

        Thread.sleep(5000);
        for (Integer n : map.values()) {
            System.out.println(n);
        }
    }
}
```

ConcurrentHashMap Links

- https://www.geeksforgeeks.org/concurrenthashmap-in-java/
- https://www.javatpoint.com/java-concurrenthashmap
- https://www.baeldung.com/java-concurrent-map

- https://jenkov.com/tutorials/java-util-concurrent/concurrentmap.html

ConcurrentNavigableMap

- The java.util.concurrent.ConcurrentNavigableMap class is a java.util.NavigableMap with support for concurrent access, and which has concurrent access enabled for its submaps.
- The "submaps" are the maps returned by various methods like headMap(), subMap() and tailMap().
- java.util.concurrent.ConcurrentNavigableMap classi parallel dasturlashda ishlatish mumkin bo’lgan java.util.NavigableMap dir. U subMap lariga parallel dasturlashda ishlatish imkonini beradi.
- subMap larni headhMap(),subMap(), tailMap() methodlari orqali olish mumkin.

ConcurrentNavigableMap

- https://jenkov.com/tutorials/java-util-concurrent/concurrentnavigablemap.html