**Concurrent Collection– Part\_08**

* **Difference between ConcurrentHashMap, synchronizedMap and HashTable:**

|  |  |  |
| --- | --- | --- |
| **ConcurrentHashMap** | **synchronizedMap()** | **HashTable** |
| We will get thread-safety without locking total Map object, just with bucket or segment level locking. | We will get thread-safety by locking total Map object | We will get thread-safety by locking total Map object. |
| At a time, multiple threads are allowed to operate on Map object in safe manner. | At a time only one thread is allowed to perform any operation on Map object. | At a time only one thread is allowed to perform any operation on Hashtable object. |
| Read operation can be performed without lock. But write or update can be performed with bucket level lock. | Every read and write operation require total Map object lock. | Every read and write operation require total Hashtable object lock. |
| While one thread iterating Map object other threads are allowed to modify the Map object. We will not get concurrent modification exception. | While one thread iterating Map object, other threads are not allowed to modify the map, otherwise will get ConcurrentModification  Exception. | While one thread iterating Map object, other threads are not allowed to modify the map, otherwise will get ConcurrentModification  Exception. |
| Iterator is fail-safe | Iterator is fail-fast | Iterator is fail-fast |
| Null not allowed for both keys and values | Null is allowed for both keys and values. | Null is not allowed for both key and values. |
| Introduced in 1.5 Version | Introduced in 1.2 Version | Introduced in 1.0 Version (legacy). |