Write caching layer along with Junit test cases for standalone java application with following assumptions:

- Cache should be globally accessible throughout the JVM
- Cache client will access cached objects by providing table name & primary key value
- All tables have integer primary key
- Initially three tables are required to be cached, REF_COUNTRY, REF_CURRENCY & REF_HOLIDAY
- Cache should have provision to support more tables if required
- Cache should have provision to manual and auto removal of cache
- Code should be adhered to java development standards
- Concurrent access should be enabled using Read/Write Locks

Note:

1. **Code should be written using JDK only (don’t use any readymade APIs or Framework)**
2. **You are not required to write complete DAO layer for the tables (instead you can return dummy data from DAO methods to pass the JUnit test cases)**
3. **Provided solution should not have any specific dependencies and should be runnable on any Java enabled machine.**
4. **Test Cases and Client code is expected to test the cache implementation.**

REF_COUNTRY		
Column Name	**Pk**	**Data Type**
COUNTRY_ID	1	NUMBER (9)
COUNTRY_CD		VARCHAR2 (20 Byte)
COUNTRY_NAME		VARCHAR2 (50 Byte)
STATUS_IND		NUMBER (3)
LAST_MODIFY_DT		DATE
MODIFY_PERSON_NUM		NUMBER (9)

REF_CURRENCY		
Column Name	**Pk**	**Data Type**
CURR_ID	1	NUMBER (9)
CURR_CD		VARCHAR2 (3 Byte)
CURR_NAME		VARCHAR2 (50 Byte)
CURRENCY_SYMBOL		VARCHAR2 (5 Byte)
STATUS_IND		NUMBER (3)
LAST_MODIFY_DT		DATE
MODIFY_PERSON_NUM		NUMBER (9)

REF_HOLIDAY		
Column Name	**Pk**	**Data Type**
HOLIDAY_ID	1	NUMBER (9)
HOLIDAY_CD		VARCHAR2 (30 Byte)
HOLIDAY_NAME		VARCHAR2 (50 Byte)
HOLIDAY_DT		DATE
STATUS_IND		NUMBER (3)
LAST_MODIFY_DT		DATE
MODIFY_PERSON_NUM		NUMBER (9)