Checklist review today:

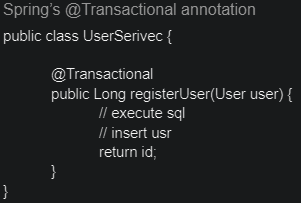
Collection

* List
  + ArrayList
    - insert/remove takes O(n)
    - retrieve data O(1) (random access)
  + LinkedList
    - insert/remove takes O(1)
    - retrieve data O(n)
  + ~~Vector~~ (not under Collection)
    - Deprecated
    - thread safe
* Set (unique)
  + HashSet HashMap
    - doesn’t keep insertion order
    - internal work by HashMap (store key (unique), value = null)
  + LinkedHashSet
    - insertion order
  + TreeSet
    - sorted
* Deque (double-ended queue)
  + ArrayDeque
    - deque.offerFirst(), deque.pollFirst()
  + ~~Stack~~ (not under Collection)
    - push(), pop()
    - Deprecated
    - thread safe
* Map (not implements Collection)
  + HashMap
    - internal work
      * hashcode()
      * equals()
      * separate chaining (Linked List) O(n)
        + red-black tree (JAVA 8) O(logn)
  + ~~HashTable~~
    - Deprecated
    - thread safe (1 lock)
  + ConcurrentHashMap
    - thread safe (16 lock)
  + LinkedHashMap
    - Insertion Oder
  + TreeMap
    - Sorted (by key)

New things learned today:

AOP (implement by dynamic proxy design pattern)

* Aspect oriented programming
* aspects enable the modularization of concerns such as transaction management, logging or security that cut across multiple types and objects (cross cutting corners)
* Aspect
  + a class that implements enterprise application concerns that cut across multiple classes, such as transaction management, logging, security
* Join Point
  + a point during the execution of a program, such as the execution of a method or the handling of an exception
* PointCut
  + a predicate that matches join points,
  + Advice is associated with a pointcut expression and runs at any join point matched by the pointcut
  + Spring framework uses the AspectJ Pointcut expression language
* Advice
  + before: advices that executes before a join point
  + after: advice to be executed regardless of the means by which a join point exits (normal or exceptional return)
  + after return: advice to be executed after a join point completes
  + after throwing: advice to be executed if a method exits by throwing an exception
  + around: advice that surrounds a join point such as a method invocation
* Transaction
  + JDBC: con -> try(con){statement} catch(SQLExcetopm e)
  + Spring
    - 一張含有 文字 的圖片

      自動產生的描述
    - XML file
    - 
* Propagation Level and Isolation Level
  + required
  + requires\_new
  + support
  + mandatory
  + not\_supported
  + never
  + nested
  + 
* Isolation Level
  + default: will use the default isolation level for the RDBMS
  + read\_uncommitted
  + read\_committed
  + repeatable\_read
  + serializable

Learned Plan tomorrow:

* Comparator vs Comparable
  + Comparator
    - compareTo
  + Comparable
    - compara
* Review Multithreads
  + states
  + 3 creation ways
    - extend Thread
    - implements Runnable
    - implements Callable