Checklist review today:

Exception

* checked exception
  + IO or Compile time exception
* unchecked exception
  + Runtime or Null Pointers Exception
* Handle Exception
  + try catch (finally)
  + throws
  + throw vs throws
* customize exception
  + user class extends Exception
* handle multiple exceptions
  + catch(E1){}

catch(E2){}…

* + catch(E1|E2|E3…){}

15. Generics - <K, V, E> ..

* easier and less error-prone
* enforce type correctness at compile time
* without causing any extra overhead to your application

New things learned today:

* Authentication
  + username + password
* Authorization
  + role
* security
  + at rest
  + in transit
  + password, ssn, …
* Encryption
  + Symmetric (Data encryption / same key)
  + Asymmetric (signature, authentication / public key, private key)
* Hashing
  + md5, SHA -> 128bit (hex) for verify
* Encoding
  + URLencoding, FileEncoding (binary-character) - Base64
* 1. validate input in controller (SQL Injection, XSS attack..)
* 2. HTTP + security (TSL > SSL) (HTTPS)
  + Both request and response are encrypted by a unique key before they are sent.
* 3. Token -> JWT (Json web token) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
  + JWT - base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
  + used for carrying data between client and server.
  + data can be read by either side
  + ONLY server can verify and generate the JWT
  + signature is used for verification
  + in the payload we have "claims" like expiration date, issuer, and other customized data.
  + the client sends the token back to the server by using the "Authorization" header.
* OAuth2
  + using third party account ( e.g. Google, Facebook, …) public or private (company)
* SSO
  + single sign on - login once, login everything
* LDAP
  + Lightweight Directory Access Protocol

Learned Plan tomorrow:

* + Review Lock
    - synchronized
    - Lock Interface
    - ReadWriteLock Interface
  + Garbage Collector
    - serial GC
    - parallel GC
    - G1 GC
    - minor GC
    - major GC