----------day 2.20

156.What are the differences between the two data structures: a Vector and an ArrayList?

ArrayList ----- not synchronized. ArrayList increments half of current array

Vector ---- **Vector is synchronized**. Vector increments 100% means doubles the array size

157.What are the differences between Collection and Collections in Java?

The Collection is an interface whereas Collections is a class. The Collection interface provides the standard functionality of data structure to List, Set, and Queue. However, Collections class is to sort and synchronize the collection elements.

158.In which scenario, LinkedList is better than ArrayList in Java?

A LinkedList consumes a bit more memory than an ArrayList since every node stores two references to the previous and next element. The **insertion, addition, and removal** operations are faster in a LinkedList because there is no resizing of an array done in the background.

159.What are the differences between a List and Set collection in Java?

Set ----unordered and unique elements,

list --- ordered and contain the same elements.

160.What are the differences between a HashSet and TreeSet collection in Java?

TreeSet --- elements are sorted.

HashSet---elements are not sorted or ordered.

161. In Java, how will you decide when to use a List, Set or a Map collection?

Set--- group of unique elements, not order.

List---duplicate elements with ordered.

Map: object have the key and value pair. Key unique.

162.What are the differences between a HashMap and a Hashtable in Java?

HashMap---key 1 null+ multiple null val. not synchronized

Hashtable ---key and val total not null. **Synchronized.**

163.What are the differences between a HashMap and a TreeMap?

HashMap---key 1 null + multiple null values. **not synchronized**

TreeMap ---key not null + multiple null values.

164.What are the differences between Comparable and Comparator?

both Comparable and Comparator are used for sorting objects. All interface.

Comparable---natural sorting order, lang package, 1 method (CompareTo()),single way sorting.

Comparator---customized sorting order, util package, 2 methods(compare(), equals()) multiple ways sorting .

165.In Java, what is the purpose of Properties file?

A Java properties file **defines the values of named resources that can specify program options such as database access information, environment settings, and special features and functions**. The property-key-name is an identifier for the resource

166.What is the reason for overriding equals() method?

check whether two objects have same data or not.

167.How does hashCode() method work in Java?

It returns an integer whose value represents the hash value of the input object. It overrides hashCode in class Object. By default, this method returns a random integer that is unique for each instance

168.Is it a good idea to use Generics in collections?

Generics means parameterized types. The idea is **to allow type (Integer, String, … etc., and user-defined types) to be a parameter to methods, classes, and interfaces**. Using Generics, it is possible to create classes that work with different data types.

1.What is maven?

Used for Java-based projects, **helping to download dependencies**, which refers to the libraries or JAR files. The tool helps get the right JAR files for each project as there may be different versions of separate packages.

2.Life cycle in Maven?

3lifecycles(clean, default(or build), site)

8 steps: Validate, Compile, Test, Package, Integration test, Verify, Install, and Deploy

- **Validate:** This step validates if the project structure is correct. For example – It checks if all the dependencies have been downloaded and are available in the local repository.
- **Compile:** It compiles the source code, converts the .java files to .class, and stores the classes in the target/classes folder.
- **Test:** It runs unit tests for the project.
- **Package:** This step packages the compiled code in a distributable format like JAR or WAR.
- **Integration test:** It runs the integration tests for the project.
- **Verify:** This step runs checks to verify that the project is valid and meets the quality standards.
- **Install:** This step installs the packaged code to the local Maven repository.
- **Deploy:** It copies the packaged code to the remote repository for sharing it with other developers.

3. Important commands in maven?

- **mvn clean:** Cleans the project and removes all files generated by the previous build.
- **mvn compile:** Compiles source code of the project.
- **mvn test-compile:** Compiles the test source code.
- **mvn test:** Runs tests for the project.
- **mvn package:** Creates JAR or WAR file for the project to convert it into a distributable format.
- **mvn install:** Deploys the packaged JAR/ WAR file to the local repository.
- **mvn site:** generate the project documentation.
- **mvn validate:** validate the project's POM and configuration.
- **mvn idea:idea:** generate project files for IntelliJ IDEA or Eclipse.
- **mvn release:perform:** Performs a release build.
- **mvn deploy:** Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project.
- **mvn archetype:generate:** This command is used to generate a new project from an archetype, which is a template for a project. This command is typically used to create new projects based on a specific pattern or structure.
- **mvn dependency:tree:** This command is used to display the dependencies of the project in a tree format. This command is typically used to understand the dependencies of the project and troubleshoot any issues.

4.What is pom.xml ?

Project Object Model, and it is the core of a project's configuration in Maven. It is a single configuration XML file called pom. xml that contains the majority of the information required to build a project

5.What is packaging available?

Maven offers many default packaging types that include a **jar, war, ear, pom, rar, ejb, and maven-plugin**. Each packaging type follows a build lifecycle that consists of phases. Usually, every phase is a sequence of goals and performs a specific task.