Git – Distributed Version Control Software.

It’s one of the Important Source Code Management Tool.

Git is Open Source Tool.

<https://git-scm.com> -- Linus Torvalds

Github – Online cloud based code storage service.

Git clients – CLI client and GUI clients.

While installing git, make sure the configure the default editor.

Git uses unix, linux style commands.

Add Git bash & Git GUI options in the file explorer.

Git Architecture

Workspace Folder

Staging Area

Local Repository

Remote Repository

Various Git Commands

Git init

Git clone

Git add

Git push

Git commit

Git status

Git log

Git diff

Git reset

Git remote

Forking

Cloning

Starring

Adding Collaborators

Changing visibility

Adding files and folders to repositories.

Java 8 Features

* Functional Interfaces
* Lambda
* Streams
* Parallel Programming
* Functional Programming
* Predicate
* New Date & Time
* JDBC improvements
* Concurrency Improvements

Functional Interface.

Relationships in JAVA

* Is A
* Has A

Car is a Vehicle

Bike is a Vehicle

Student is a Person

Employee is a Person

Trainer is a Person

Parrot is a Bird

Train is a vehicle

Flight is a vehicle

Interfaces is a kind of generalization.

Interface is the one used to provide the specification.

Important characteristics of Java Interfaces

* All the properties (variables defined in it) are by convention public.
* All the methods (behaviors) defined in interfaces are by convention public and abstract (non-concrete/in-complete).
* Interface can extend any number of interfaces
* Private/protected properties are not legal in interface
* Using un-initialized variable(property) is illegal in interface
* By convention all the properties of interfaces are static & final
* Interface allows public, static & final member variables only

Abstract is a keyword used in java.

Abstract keyword can be used in methods and classes only.

If a method is abstract, it means, it contains only the declaration part (Only the signature part- No body) no definition.

If a class contains one or abstract methods, then the class is Abstract Class.

Abstract class – is a type of class, where it has one or more in-complete/non-concrete/abstract methods.

Directly creating Objects of abstract class is not allowed in Java.

Access Modifiers in JAVA

1. Private
2. Default/package (No access modifier is specified)
3. Protected
4. Public

Types of Interfaces

1. Marker Interface – An interface with no methods (Serializable)
2. Functional Interface – An Interface with only one abstract method ( Java 8)

Serialization = Converting object into byte-stream. It is the process of storing the state of an object in a flat file.

Characteristics of Functional Interface

1. It should have only one abstract method
2. It can have many static methods, default methods
3. To avoid adding more than one abstract method accidentally, we can use @FucntionalInterface annotation to mark it as a functional Interface

Anonymous – Nameless (Without Name)

Lambda = Anonymous Function = Nameless function

//traditional function declaration in JAVA

Public void function\_name(args\_type\_1 arg\_name\_1,… args\_type\_n, arg\_name\_n) {

}

Public void (argslist) {

}

Anonymous function needs to be stored to some ref variables.

(args\_list) {

}

()-> {} ---- This is lambda syntax

Functional Programming – It’s elegant style of programming.

Doing multiple operations in a single line.

Functional Interface – Single method interface

Lambda helps to create nameless function/methods.

Also it is used to store a function into reference variables.

In Lambda, we are providing implementation to a abstract method of Functional Interface.

In angular & typescript it is known as arrow functions. [()=>{}] – Fat arrow function.

All the interfaces defined in java.util.function package are functional interfaces.

Synchronous VS Asynchronous

Synchronous Operation – Serial Operation

Asynchronous Operation – Parallel Operation

1. Printing a variable in the console (2ms)
2. Getting a username from user in console using Scanner (900ms)
3. Opening a table in the mysql database and reading content of a table (2500ms)
4. Creating a thread, and making it to sleep for 10000ms (10000ms)
5. Opening a file and reading its content and printing it in console. (1500ms)

Total time = 10000+2500+1500+900+2 = 14902ms (Sync)

Total time – 10000 ms (Async)

<https://codeshare.io/r9V1qK>

Lambda - -Anonymous method

Functional Interface reference is also called as Functional reference. (It stores a lambda expression)

Lambda expression is also called as functional expression.

Streams in Java – Introduced in Java8

Stream API is used to process collections of objects.

Stream – Sequence of Data/Objects that supports various methods which can be pipelines to generate desired output.

* Stream is not a Data Structure
* Data Structure – It’s a way of storing the data in an organized format [Array]
* Streams take input from Collections, Arrays or I/O channels
* Streams won’t change the original data, they just provide the result as per the pipelined/intermediate methods
* Each intermediate/pipelined operation will return a stream as a result(Lazily executed)
* Various intermediate operations can be pipelined
* Terminal Operations mark the end of the stream and pass/return the result

List of Intermediate Operations

1. Map – Used to return a stream consisting of the result applying given condition/function

List numbers = Arrays.asList(1,2,3,4,5,6,7,8,9,10);

List square = numbers.stream().map(x->x\*x).collect(Collectors.toList());

1. Filter – used to select few elements matching the search condition as per the predicate passed as an argument

List names = Arrays.asList(“ABC”, “XYZ”, “LMN”, “XXL”, “XL”);

List result = names.stream().filter(s->s.startsWith(“X”)).collect(Collectors.toList());

1. Sorted

Terminal Operations

1. Collect
2. forEach
3. reduce

Method Reference

* Instance Method Reference
* Static Method Reference
* Constructor Reference

Types of Methods in JAVA

1. Normal methods/ concrete methods
2. Static methods (Once per Class)
3. Constructor (default/parameterized)