# **JAVA**

## **Usage**

1. Web development
2. Desktop development
3. Mobile development
4. Game development
5. Distributed processing

## **Features of Java**

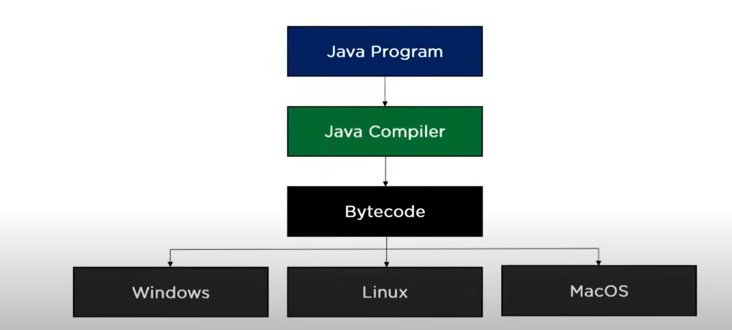
1. Object oriented
2. Platform independent
3. Strong Type-checking
4. Compile & interpret
5. Robust (Exception handling related).
6. Secure
7. Multi-Threading
8. Garbage Collector

* Not available in c and c++.
* It will find the unused objects and variables and removed while executing to free up the memory.

## **How a java program is**

Using the byte code we can execute it anywhere because of that it is platform independent.

Once a java program is executed, the compiler generates a .class file.



## **Data types**

### Primitive data types

* Byte – 8 bit shortest data type to store integer (-128 to 127)
* Short – 16 bit (-32768 to 32767)
* Int – 32bit (-231 to 231-1)
* Long – 64 bit (-264 to 264-1)
* Float – 32 bit
* Double – 64 bit
* Boolean
* Char

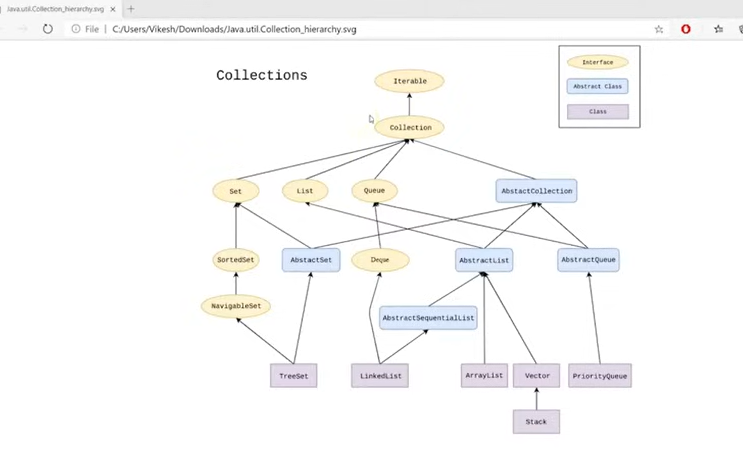
### Non primitive data types

* Array
* String

## **Java Tokens**

* Keywords
* Identifiers – variable name
* Constants – final variables (final int a = 1)
* Symbols – [] {} () , ; \* =
* Operators – arithmetic, comparison, logical, bitwise etc.

## **Collections**



### List

#### Array list

Used to sequence of data in same data type

#### Linked list

Same to array list but here it was represented in node. In each node it has the value and the address to the next node. If you want to access the last or particular code every time need to iterate to the node based on the address.

### Set

Set always store unique values, duplicate values are not allowed to store in set.

#### Hashset

If we store 10 items in hashset, and while retrieving for every time it return the value in different order, it only the stores the value not the sequence, that’s why every time retrieving it returns is an random order.

#### Treeset

In there will be order, and it is hierarchy structure not in stored structure based on value.

### Priority queue

FIFO based on will process and it any thing will not wait for its turns comes we can assign priority to that and based on that also it will process

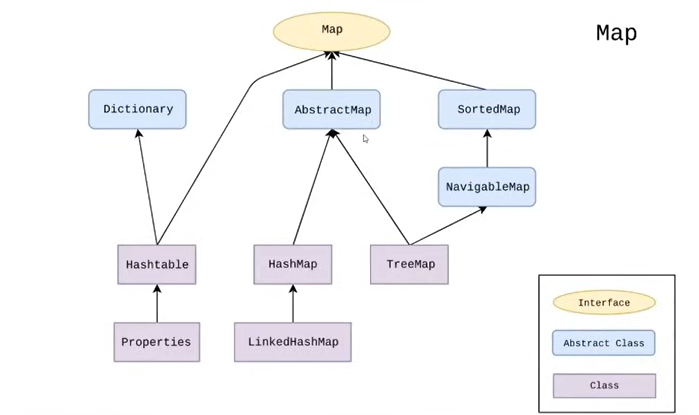
### Stack

LIFO manner

### Vector

Similar to list, but additionally it is thread safe.

## **Map**



Map is based on key value pair, it is used set to store keys, which the keys will be unique and list to store value, which allows to store duplicate values. HashMap and tree is not thread safe and hash table is thread safe. Hash table is slower because only one thread can access at a single time and hash map and tree map will be faster because multiple thread can access it at a single time.

## **Stream**

If we are watching video in youtube instead of downloading the whole video and placing it in the memory, it will download bit by bit. Introduced from java 8.

If we want to change the representation, we need to use map.

Lamda method introduced in java 8. Syntax: Introduction -> Operation or value. Example: a -> a\* 8

Collect method is a terminal method. Once we called an terminal method, the stream is ended and no more any streams functions added to it.

Filter method used to add an conditional logic, then we need to use filter method.

Whenever if we want to change the complete representation collection into a single result, then we need to use the result method.

* + - // Identity -- will the default value and default result if the stream is empty
    - // Accumulator -- have two params, partial result of reduction and the next
    - // element of stream
    - // Combiner -- optional function, if the accumulator is different data type, then we need to use combiner

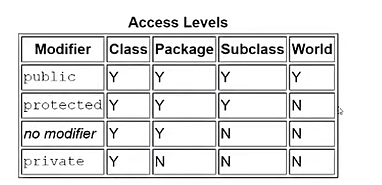
## **Access modifiers**

It is used for security. By default it is no modifer. We can use this in front of classes, variable and methods.

Package is more likely the folder that contains multiple related classes.

Subclass in comes under inheritance, it is the child class.

Word means outside the package.



## **Inheritance**

A base class properties can be extended to child class.

While using constructor to initialize child class variables, use super function to call the parent class constructor with required arguments and whenever using super function it should be the first in child constructor.

A child class will not have multiple parent class in java, it can be done with interface.

## **Encapsulation**

En Capsul ation – split of words

Capsul used to pack lot of things into small tube.

Encapsulation is used to restrict accessing properties outside word.

While implementing need to set attributes set as private.

Nested classes also comes under this encapsulation

## **Abstraction**

Abstraction is generally used to hide something.

Abstract class can not be have construct, but they can be subclassed.

Abstracted method is declared without an implementation.

Abstract class should have minimum one abstract method and it can also have normal methods.

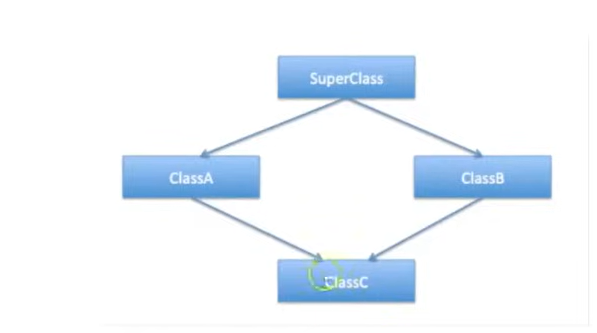
## **Interface**

For example: if we different car manufactures using different gear system, it is difficult to user to use different vehicles from manufactures.

Interfaces is going to define the specification of the class.

In interface we cannot define normal or concrete method. And we cannot declare only variable, we need to declare as only public static final. Abstract is less restricted and interface is more restricted. Interface cannot be initialized using super class.

## **Multiple inheritance in Java**



It is not possible to extend from two classes.

It is possible to extend from two interfaces.

It is possible to extend one class and multiple interfaces.

## **Polymorphism**

Particular object can take multiple form.

Polymorphism is nothing but it is method overloading. In method overloading the name of the method is same, the return method can change and no. and type of arguments should differ.

## **Method overriding**

We can implement method overriding with the help of inheritance.

Dynamic object binding, using parent class in left side and using child class in right side.

## **Static**

Static variable – static variable will belong to the class not to the object, so if we declare two objects for a class and change the value of static variable from first class, it will be reflected in the second class also.

Static method is also same to the variable. Static method will only access the static variables.

If we use final with the static variable it will become constant.

Static blocks of codes called only once when class is loaded, irrespective of no. of objects loaded for that class. Static variable also initialized the same when the class is loaded, irrespective of no. of objects.

## **Nested class**

A class within a class.

Two categories: Static classes, Static nested classes, Non-static nested classes (inner class).

A more for encapsulation.

## **Packages**

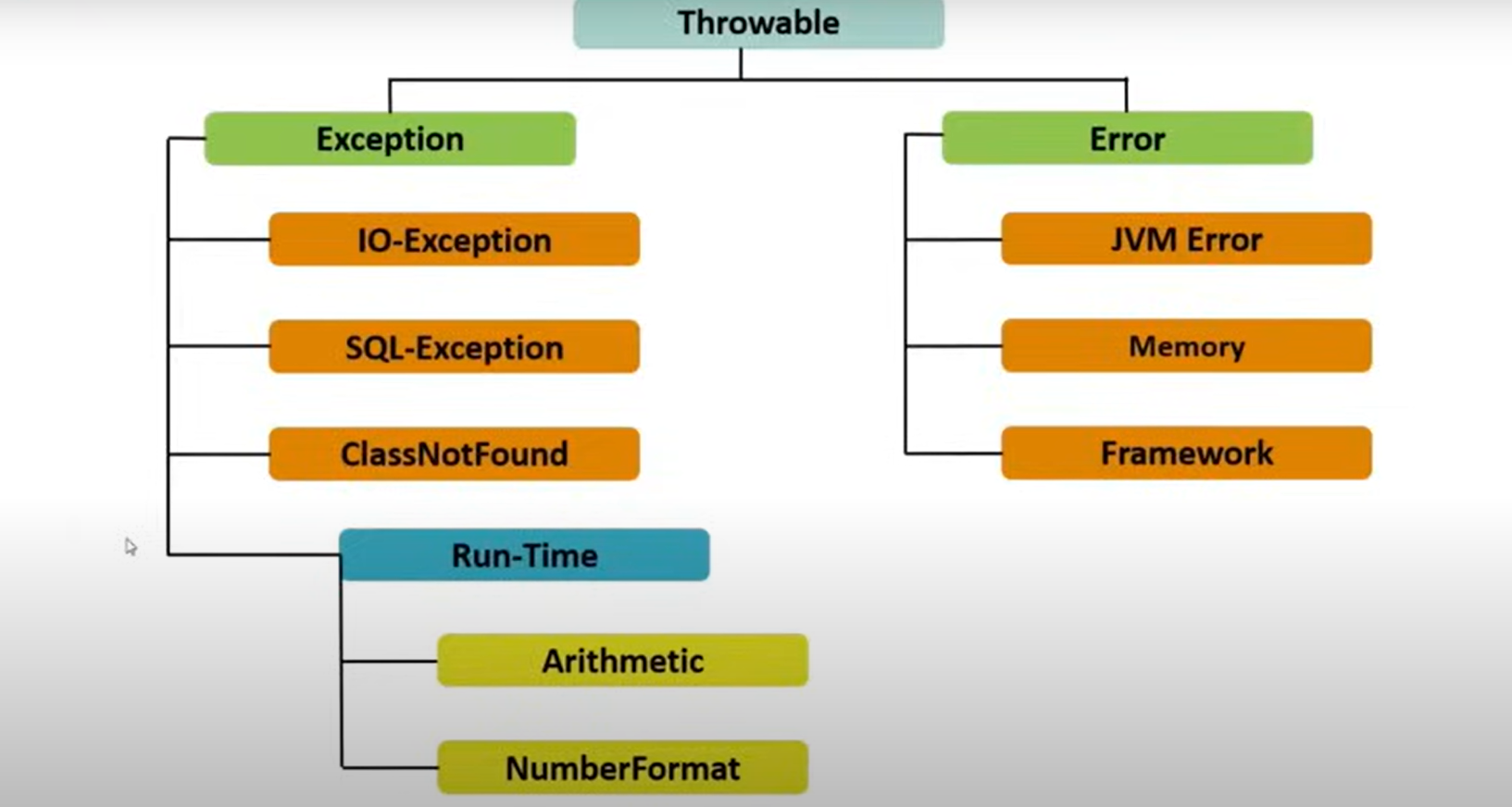
Group of related types, types can be classes, interfaces, abstracts, nested classes or anything.

We can create nested packages using . while creating packages.

## **Recursion**

A method should call itself is called recursion. Recursion method static based execution.

## **Exception**



Run time exceptions are unchecked exceptions which are thrown at run time. Checked exceptions are throws at compile time, if we are using filehandler in java we need to handle exceptions thrown by filehandler.

## **Equals and hashcode**

Every object has a hascode, hascode is nothing but a memory footprint.

To implement equal between two custom classes need to override both equal and hashcode.

## **Comparable**

Is needed to sort our collections. To use this for our custom classes we need to implement comparable interface to our class.

## **Comparator**

Comparator used to sort custom collection on multiple methods. For example we need to sort on roll number or marks or name or address, we can sort on multiple attribute on same collection.