Just Java

Sofiullah Iqbal Kiron

April 29, 2021

1 Beginning

Java is an open source high-level, general-purpose(hard & used to develop any kind of programs), object oriented language. If you wanna be a software developer than you must learn that 4 languages:-

1. Python
2. Java
3. JavaScript
4. C / C++

In those, Java is second dominant (and popular also) language that you must learn if you wanna be a platform-independent developer. Also:-

→ Java has huge online community for getting help.

→ Can be used in android development that is most preferable thing today.

Gosling explains his motivation for creating JAVA: James Gosling, father of **JAVA**. It is one of the world's most widely used programming language. It used over ten billion of devices and become central to the development of Android at Google. According to **Oracle**, there are 51 billion active Java Virtual Machines deployed globally.

Updated list (13 Nov, 2020)

1. Python
2. JavaScript
3. Java
4. C / C++

Java has two types of error. Syntax error and Semantic error. Syntax error is grammatical error and semantic error is that the line has no meaning.

→ ”I are playing” - that's the syntax error.

→ ”He is hello” - this line has no meaning, semantic error.

2 Mosh

2.1 Behind

In programming, there are many type of paradigms(ways of formatting/writing code) like: Procedural, Functional, Object-oriented, Event-driven, Logic, Aspect-oriented. From those, Functional(JavaScript often use) and Object-oriented are two most popular programming paradigms. In OOP, everything based on object. An object has two main features, data/state/field and function/method/behavior.

3 JAR

Means Java Archive like as zip files. That encapsulate and compress project files into one file. Package file format typically used to aggregate many java classes and associated metadata[1] and resources(text, images) for distribution. The contents in a jar file can be decompressed and extracted using any standard decompression software or jar command line utility: "jar -xf file_name.jar". Some operating systems can run when "Just Click"ed. Command line argument: "java -jar file_name.jar". To create native launcher[2] at Windows, JSmooth, Launch4J(free), WinRun4J etc. For further information, Google IT.
To work with JAR files, use Java Archive Tool that provided as part of the Java Development Tool(JDK). Invoke using "jar" command utility.

Creating a jar: `jar cf jar-file.jar input-file(s)`[3]

View contents: `jar tf jar-file.jar`

Extract contents: `jar xf jar-file.jar`

Run: Run an application packaged as JAR (requires the Main-Class manifest header). `java -jar app.jar` The `-jar` flag tells the launcher that the application is packaged in the JAR file format. You can only specify one JAR file, which must contain all of the application-specific code. Before you execute this command, make sure that the runtime environment has information about which class within the JAR file is the application's entry point.

Entry Point: To setting up the applications entry point,

4 The Process API, Since 9

API: Application Programming Interface.
The process API lets you start, retrieve information about and manage native system processes. The process API consist of ProcessBuilder class, Process abstract class, ProcessHandle interface and ProcessHandle.Info interface.

5 Queue, Since 1.5

Queue interface is a member of Java Collections Framework (JCF). A collection designed for holding elements prior to processing. Beside basic Collection operations, queues provide additional Insertion, Extraction and Inspection operations. We can use all of Collection methods to a queue. Declaration in JDK:

```
public interface Queue<E>
```

[1] Data that provides information about another data, data of data.
[2] executable software at current OS
[3] Space separated names with dot extension

Where E is the type parameter.
Queues typically but do no necessarily, order elements in First-In-First-Out, FIFO manner. All new elements are inserted at the tail of the queue. Every queue implementations must specify it's ordering properties. Using boolean offer(E e) method is preferable to Collection.add(E e) cause, it inserts an element if possible, otherwise returns false. The E remove() and E poll() methods remove the head after return it. If the queue is empty, remove() method throw an exception where poll() return null. I think poll() is good enough.
The element() and peek() method return head without removing it. peek() is special because it return null if the queue is empty where element() throws exception.
Queue implementations not allows null insertion, generally. Although some implementations like LinkedList, do not prohibit insertion of null but we should not insert null because poll() method returns null to indicate that the queue is empty. I would use LinkedList implementation.
Other methods in java.util.Collection useful for queue: addAll, clear, contains, equals, isEmpty, size, toArray.

6 Deque: Member, Java Collection Framework

The deque(usually pronounced as deck) is double-ended-queue is a linear collection of elements that supports the insertion and deletion from both ended point. Predefined classes like **ArrayDeque** and **LinkedList** implements **Deque**. It is rich because it implements both **stack** and **queue** at the same time.

Deque: Insert Methods available, **addFirst()**, **offerFirst()** and **addLast()**, **offerLast()**.
offerFirst() and **offerLast()** methods are preferable for insertion.

Deque: Remove Methods: **removeFirst()**, **pollFirst()** and **removeLast()**, **pollLast()** removes element from the collection.
pollFirst() and **pollLast()** are preferable because they return null if the collection is empty wheres **removeFirst()** and **removeLast()** throws exception(Exceptions are not easy to handling, that's why).

Deque: Retrieve Methods: **getFirst()**, **peekFirst()** and **getLast()**, **peekLast()**.
peekFirst() and **peekLast()** are preferable because they will return null if the collection is empty wheres the other two methods throws exception.

Some other useful methods: removeFirstOccurrence(Object o), removeLastOccurrence(Object o), contains(Object o), size(), removeAll(), addAll(), isEmpty(), toArray() etc.

7 Map

Map is an interface representing (Key, Value) pair. Map does not contain duplicate key. Declaration

```
public interface Map<Key, Value>
```

Some implementing classes: AbstractMap, HashMap, LinkedHashMap, TreeMap(Sorted implementations). To see full list of implementing classes, browse on the oracle's site.
The Map interface provides three collection views:

1. Set of keys
2. Collection of values
3. Set of key-value pair

8 JToolBar

JToolBar is a component class of javax.swing . I love it. JToolBar is a group of buttons that provides commonly used actions to the users. It can be moved around runtime (floatable). Create a horizontal toolbar

```
JToolBar toolbar\_name = new JToolBar(String name, int orientation)
```

Orientations:

1. SwingConstants.HORIZONTAL, Value = 0

2. SwingConstants.VERTICAL. Value = 1

Add some buttons to the toolbar. The buttons in toolbar need to be smaller than usual buttons. We can make button smaller in size by setting its margin to zero (0). We should add tooltip to all buttons to give quick hint to the user.

9 Throwable and Exception

The Throwable class is the super class of all errors and exceptions in Java. Only objects that are instances of this class are thrown by Java Virtual Machine with throw statement. Other descriptions will be added later...

```
public void printStackTrace()
public void printStackTrace(PrintStream s)
```

This method used to print this throwable along with other details like class name and line numbers where the exception occurs means it's backtrace. Thsi method prints a stack trace on the standard error output stream. A great handy method to find out errors with details explanation.

```
public class ThrowableDemo {
    public static void main(String[] args) throws Exception {
        try {
            getException();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // This method will throw exception
    private static void getException() throws Exception {
        ArrayIndexOutOfBoundsException ae = new ArrayIndexOutOfBoundsException();
        Exception e = new Exception();

        // Initialize the cause and throw the exception
        e.initCause(ae);
        throw e;
    }
}
```

Exceptions are represented as an object in Java. Exceptions are subclass/childclass of Throwable.

10 Streams

Streams are not data type, it is pipeline of flow of data from a data storage/collection. Like a tank has water(collection) and flow of water through pipe from the tank is streams. All the classes that implements Collection class has stream method to make a data flow pipe. Like ArrayList, List has stream and utility class Arrays has stream() method to convert an array to stream.