Session Goals

- User should understand the need of Set and Map Data Structures in Java.
- User should understand when to use HashMap or a HashSet.
- User should understand the need of unique keys.
- User should be able to declare, initialize and perform CRUD Operations on HashMap and HashSet in Java.
- User should understand the concepts of JVM, JDK and JRE in Java.

Java-111- HashMap and HashSet in Java

Session 10

Session Agenda

- HashSet
- HashMap
- JVM/JRE/JDK

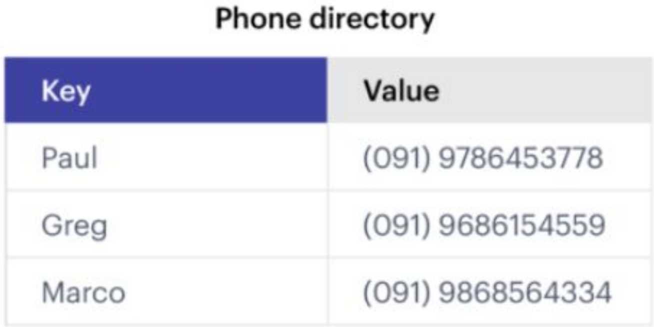
Collection: HashMap

Documentation - https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/HashMap.html

Key & Value. **Lookup by Key** and get its corresponding value as result. **No duplicate Keys.**

Examples

- Name of the month (key) and number of days in month (value)
- Name of the country (key) and its currency (value)
- Word (key) and is meaning (value)
- Roll number (key) and name of the student (value)



Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

Exercise: Create an ArrayList of 5 IDs (Integer type), and another ArrayList of 5 of your friend names. Use HashMap to create a mapping from ID -> Name.

Recap - 6 Step Strategy

1. ***Understand the problem*** *(ask questions and get clarity)*
2. ***Design test data/test cases*** *(input and expected output)*
3. ***Derive the solution - solve the problem*** *(write pseudo code)*
4. ***Test the solution*** *(against the test data/case - dry run)*
5. ***Write the program/code*** *(using Java here)*
6. ***Test the code*** *(syntax errors, run time errors, logical errors)*

Activity: Valid Anagram

Link

What will be your approach to the problem? (Step 3)

Quickly put your answers in the chat!

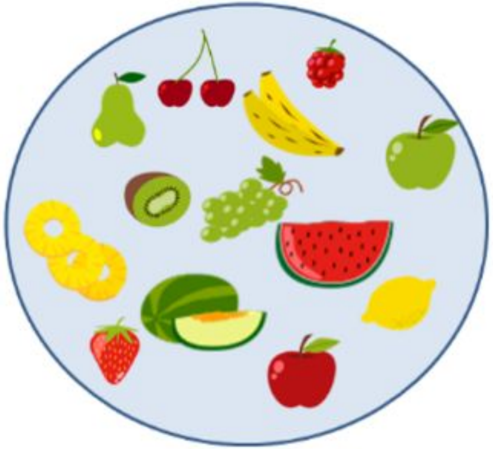
Concept #3 - Collection: HashSet

Documentation - https://docs.oracle.com/javase/8/docs/api/java/util/HashSet.html

Keys only. No Duplicates. Order not important.

Examples

- Names of all countries
- Names of all Oscar winners
- All unique numbers from a given input



Exercise: Create an ArrayList with a few repeating values. Find the unique values in that list using HashSet.

(Use the file in today's repl)

Activity: Calculate the number of distinct values

Link

What will be your approach to the problem? (Step 3)

Quickly put your answers in the chat!

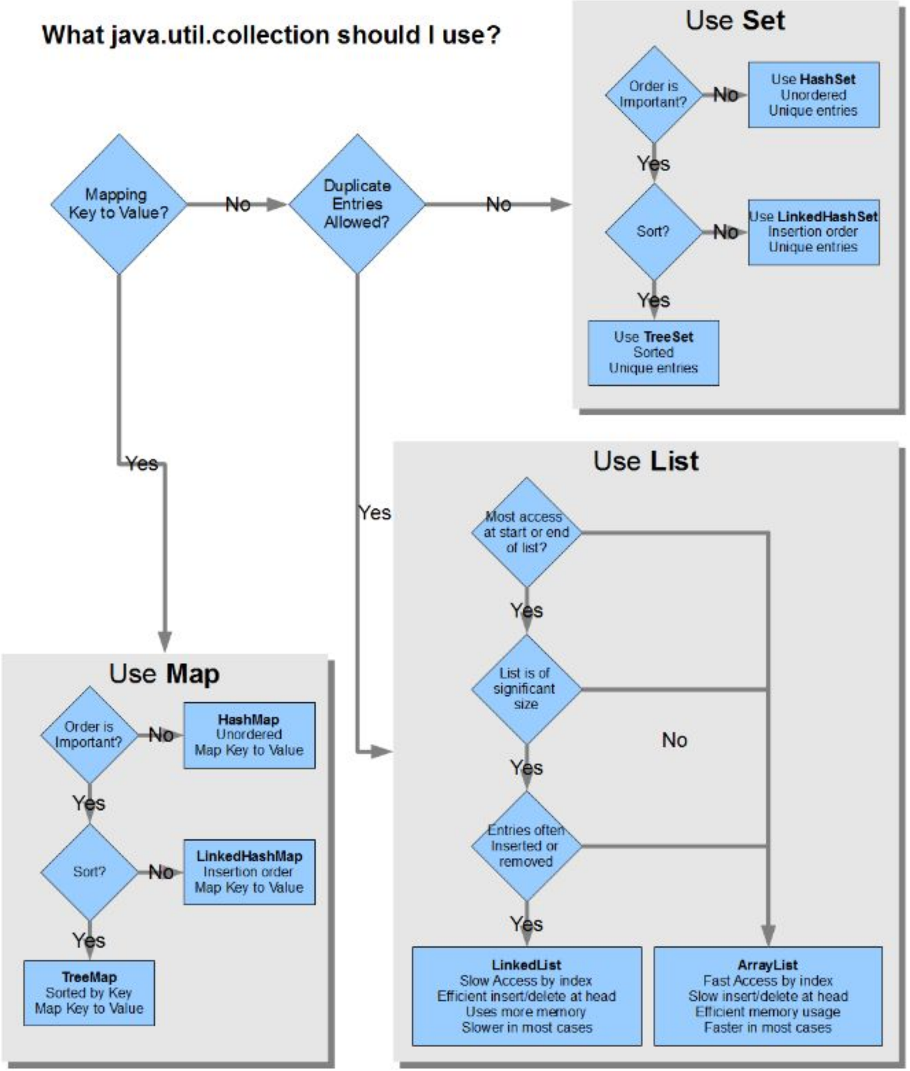
When to use what?

- If you need to store **mapping** of some key to value, use **HashMap**
- Else if you need to allow **duplicates** or maintain **order**, use **ArrayList**
- Else use **HashSet**

What Collection would you use here?

1. You are keeping track of the order in which you see vehicles go by, on the road
2. Store each product name currently offered by Apple
3. You want to store all TODO tasks and look up the TO-DO at a particular position
4. Keep track of number of 90+ scores by each student and be able to pull this up for any student
5. Track all names of visitors to a store (visitors can repeat)

Takeaway


What java.util.collection should I use?
Mapping Key to Value?
No
Duplicate Entries Allowed?
No
Use Set
Order is Important?
No
Use HashSet Unordered Unique entries
Yes
Sort?
No
Use LinkedHashSet Insertion order Unique entries
Yes
Use TreeSet Sorted Unique entries
Yes
Yes
Use List
Most access at start or end of list?
Yes
List is of significant size
Yes
Entries often Inserted or removed
Yes
No
LinkedList Slow Access by index Efficient insert/delete at head Uses more memory Slower in most cases
ArrayList Fast Access by index Slow insert/delete at head Efficient memory usage Faster in most cases
Use Map
Order is Important?
No
HashMap Unordered Map Key to Value
Yes
Sort?
No
LinkedHashMap Insertion order Map Key to Value
Yes
TreeMap Sorted by Key Map Key to Value

- Source: https://stackoverflow.com/a/21974362
- Other reading: http://www.javapractices.com/topic/TopicAction.do?Id=65

Collection: Map | Debrief

- **Map** represents a **relationship between a key and value**.
  - The key and value can be any object.
- Implementations are **HashMap**, **HashTable**

Important methods -

- get(key) : Retrieve value for the given key
- put(key, value), putAll(map) : Puts values for the key(s)
- remove(key) : Removes mapping for key
- values(), keySet() : Returns collection of values and keys respectively

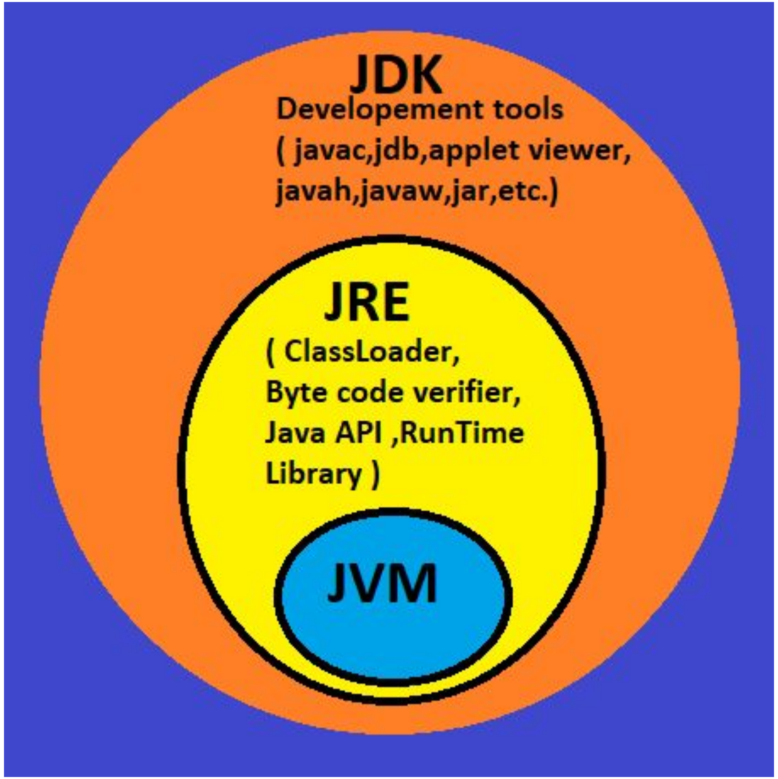
Collection: Set | Debrief

- Most important property of elements in a **Set** is: **no duplicates.**
  - Useful when we have to **track unique presence/occurrence** of elements
- Implementations are **HashSet**, **TreeSet** and **LinkedHashSet**.

Important methods

- equals() - two sets can be compared
- size() - number of elements in the set
- Common methods - add(), remove(), contains(), isEmpty(), clear()

JVM, JRE, JDK



JDK

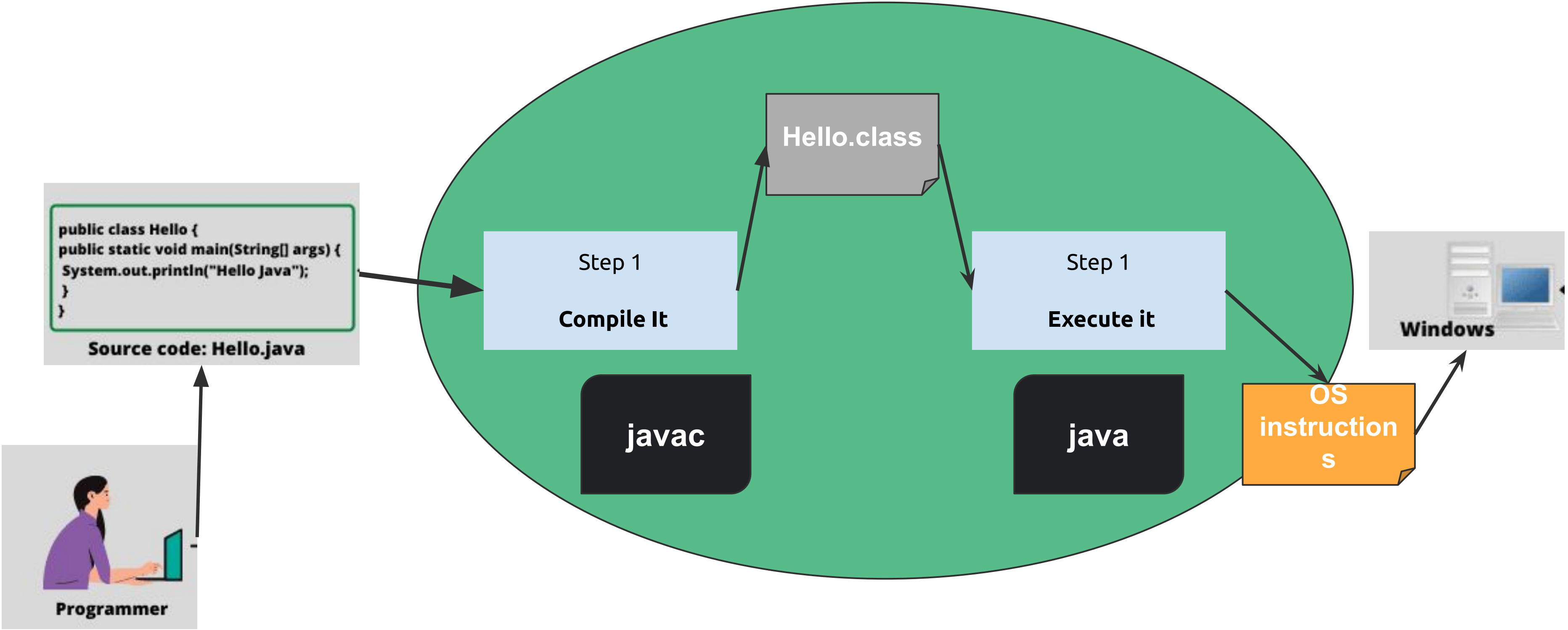
Developement tools
(javac,jdb,applet viewer,
javah,javaw,jar,etc.)

JRE

(ClassLoader,
Byte code verifier,
Java API ,RunTime
Library)

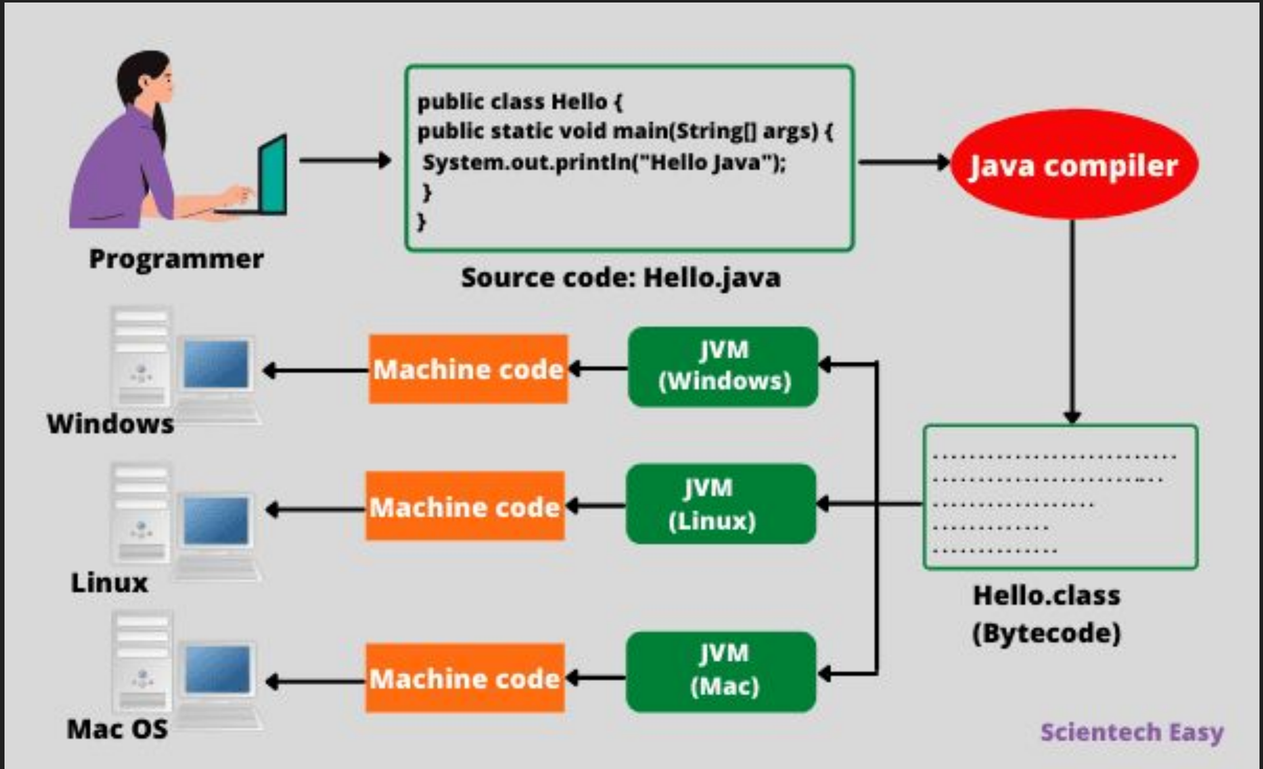
JVM

JVM, JRE, JDK



```
public class Hello {
public static void main(String[] args) {
 System.out.println("Hello Java");
 }
}
```

JVM/ JDK/ JRE



Source code: Hello.java

```
public class Hello {
public static void main(String[] args) {
 System.out.println("Hello Java");
 }
}
```

Scientech Easy

Fig: Java Programming Model

Further Reading

- Collections
- Working of HashMap in Java
- HashMap
- HashSet
- Scanner class
- JVM/JRE/JDK

Keep
Learning,
Keep
Coding.