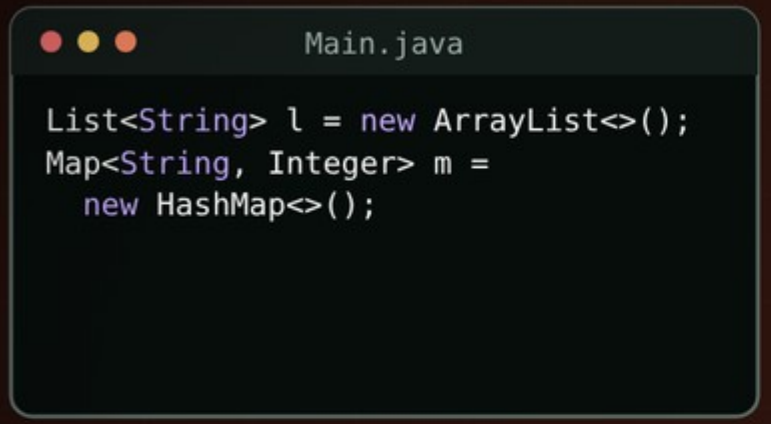
LESSON 8



Java

List · Set · Map

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

JAVA PROGRAMMING

LESSON 8

The Collections Framework

List, ArrayList, Set, HashSet, Map, HashMap, iteration and sorting

Lists, Sets and Maps

Interface	Common class	Use
`List`	`ArrayList`	ordered, by index
`Set`	`HashSet` · `TreeSet`	unique values
`Map`	`HashMap` · `TreeMap`	key → value
`Deque`	`ArrayDeque`	stack or queue

- A List keeps order and allows duplicates. A Set holds unique values. A Map links keys to …
- Declare variables with the interface type, List or Map, and create them with a class like …

01

ArrayList

Lists.java

```
List<String> cities = new ArrayList<>();
cities.add("Kabul");
cities.add("Herat");
cities.add(0, "Balkh");          // insert at front
cities.get(1);                     // "Kabul"
cities.size();                     // 3
cities.contains("Herat");          // true
List<Integer> fixed = List.of(1, 2, 3);   // immutable
```

- ArrayList is a resizable array. add appends, get reads by index, set replaces, remove …
- Collections hold objects, so int values are boxed into Integer automatically.

02

HashMap

Counts.java

```
Map<String, Integer> counts = new HashMap<>();
for (String w : text.split("\\s+")) {
    counts.merge(w.toLowerCase(), 1, Integer::sum);
}
counts.getOrDefault("kabul", 0);
for (var e : counts.entrySet()) {
    System.out.println(e.getKey() + " = " + e.getValue());
}
```

- put adds or replaces a value, get reads it, and getOrDefault avoids null when a key is …
- merge is perfect for counting: add one to the count, or start at one if the key is new.

03

Sets

Sets.java

```
Set<String> seen = new HashSet<>(List.of("a", "b", "a"));
seen.size();                  // 2
seen.add("b");               // false: already there
Set<Integer> sorted = new TreeSet<>(List.of(5, 1, 3));
System.out.println(sorted);  // [1, 3, 5]
```

- A HashSet removes duplicates instantly and checks membership very fast. A TreeSet keeps its …
- For objects in a HashSet or as HashMap keys, equals and hashCode must be correct, which …

04

Sorting with Comparator

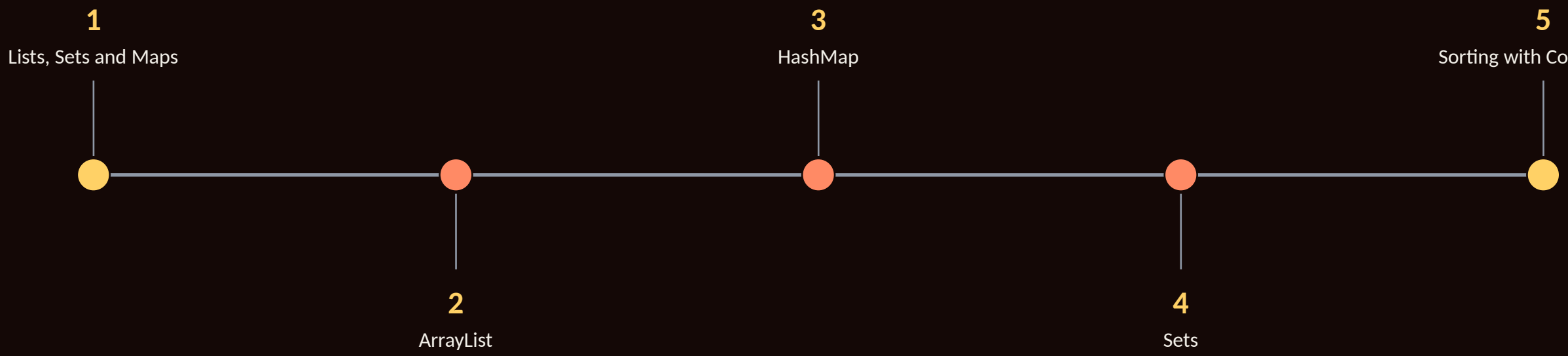
Sort.java

```
students.sort(
    Comparator.comparing(Student::score).reversed()
              .thenComparing(Student::name)
);
Collections.max(scores);
Collections.unmodifiableList(students);
```

- list dot sort takes a Comparator. Comparator dot comparing builds one from a field, and …
- reversed flips the order. This is how you sort students by score, highest first, then by name.

05

Lesson Map


1
Lists, Sets and Maps
2
ArrayList
3
HashMap
4
Sets
5
Sorting with Comparator

Sources and Further Reading

Interface	Common class	Use
List	ArrayList	ordered, by index
Set	HashSet · TreeSet	unique values
Map	HashMap · TreeMap	key → value
Deque	ArrayDeque	stack or queue

JAVA PROGRAMMING · 8

Counts.java

```java
Map<String, Integer> counts = new HashMap<>();
for (String w : text.split("\\s+")) {
    counts.merge(w.toLowerCase(), 1, Integer::sum);
}
counts.getOrDefault("kabul", 0);
for (var e : counts.entrySet()) {
    System.out.println(e.getKey() + " = " + e.getValue());
}
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari