Javrevisted-2017

**How to enable String deduplication in Java 8**

**Important points**

* This option is only available from Java 8 Update 20 JDK release.
* This feature will only work along with G1 garbage collector, it will not work with other garbage collectors e.g. Concurrent Mark Sweep GC.
* You need to provide both -XX:+UseG1GC and -XX:+StringDeduplication JVM options to enable this feature, first one will enable the G1 garbage collector and the second one will enable the String deduplication feature within G1 GC.
* You can optionally use -XX:+PrintStringDeduplicationStatistics JVM option to analyze what is happening through the command-line.
* Not every String is eligible for deduplication, especially young String objects are not visible, but you can control this by using  -XX:StringDeduplicationAgeThreshold=3 option to change when Strings become eligible for deduplication.
* It is observed in general this feature may decrease heap usage by about 10%, which is very good, considering you don't have to do any coding or refactoring.
* String deduplication runs as a background task without stopping your application.  
    
  **Difference between PriorityQueue and TreeSet in Java?**
* Both PriorityQueue and TreeSet provide O(log(N)) time complexity for adding, removing, and searching elements, both are non-synchronized and you can get element from both PriorityQueue and TreeSet in sorted order.
* **TreeSet is a Set and doesn't allow a duplicate** element,
* **PriorityQueue is a queue and doesn't have such restriction**.
* It can contain multiple elements with equal values and in that case head of the queue will be arbitrarily chosen from them.
* PriorityQueue doesn't provide any ordering guarantee.
* Only guarantee PriorityQueue gives that head will always be smallest or largest element.
* TreeSet keeps all elements in sorted order and iterator returned by TreeSet will allow you to access all elements in that sorted order.
* This is one point where you will see both similarity and difference between PriorityQueue and TreeSet in Java i.e. both provides O(log(N)) complexity for adding, removing and searching elements, but
* When you want to remove the highest or lowest priority element then PriorityQueue gives O(1) performance because it always keep the element in head.

**Difference between Transient, Persistent, and Detached Objects in Hibernate**  
A new instance of a persistent class which is not associated with a Session, has no representation in the database and no identifier value is considered ***transient*** by Hibernate:

Person person = new Person();

person.setName("Foobar"); 🡸 person is in a transient state

A ***persistent*** instance has a representation in the database, an identifier value and is associated with a Session. You can make a transient instance ***persistent*** by associating it with a Session:

Long id = (Long) session.save(person); 🡸 Persistent state

// person is now in a persistent state

Now, if we close the Hibernate Session, the persistent instance will become a ***detached*** instance

**Transient - Objects instantiated using the new operator are called transient objects.**

**Persistent - An object that has a database identity associated with it is called a persistent object.**

**Detached - A detached instance is an object that has been persistent, but its Session has been closed.**

## **Difference between @Autowired and @Inject annotation in Spring?** The @Inject annotation also serves the same purpose, but the main difference between them is that @Inject is a **standard annotation** for dependency injection and @Autowired is **spring specific**.