Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

havanagrawal / **c2c2017**

Watch 2 Star 9 Fork 4

Code Issues 1 Pull requests 0 Projects 0 Insights

Branch: master c2c2017 / **Session11 - StuffWeLeftOut** / Create new file Find file History

havanagrawal committed on **GitHub** Add section for immutable classes Latest commit 48c3055 10 hours ago

..

README.md	Add section for immutable classes	10 hours ago

README.md

Session 11 - Stuff We Left Out

Placement preparation is difficult, especially in a constrained timeline. What follows in this README is a set of topics that we did not cover, only because other topics took priority.

Given the strong foundation in Java, Data Structures and Algorithms that you now possess, understanding these topics should not be too challenging.

Table of Contents

Data Structures And Aglorithms

Heaps

1. Heaps - HackerEarth Tutorial
2. Binary Heaps - CMU Notes
3. Heap Topics - GeeksForGeeks
4. HeapSort - HackerEarth Tutorial

Self Balancing Trees

1. Self Balancing Tree Problems - GeeksForGeeks
2. Self Balancing BST Tutorial - Cal Poly Pomona

Java

Enums

> An enum type is a special data type that enables for a variable to be a set of predefined constants. The variable must be equal to one of the values that have been predefined for it. Common examples include compass directions (values of NORTH, SOUTH, EAST, and WEST) and the days of the week.

> Because they are constants, the names of an enum type's fields are in uppercase letters.

Enums - Oracle Tutorial

Packages

> To make types easier to find and use, to avoid naming conflicts, and to control access, programmers bundle groups of related types into packages.

> A package is a grouping of related types providing access protection and name space management. Note that types refers to classes, interfaces, enumerations, and annotation types.

Packages - Oracle Tutorial

Immutable Classes

An immutable class is one where an instance of that class cannot be modified in any way after construction. Immutable objects simplify multi-threaded code, since it eliminates the problem of shared read-write access.

Making an immutable class can be trickier than at first sight. The following resource lists down a few guidelines for writing an immutable class.

Immutable Classes - Oracle Tutorial

Multi Threading

Remember that:

1. Multi-threading is good, because it can take advantage of multi-core architectures, and utilizes the CPU to its fullest.
2. Multi-threading is tricky, because of issues like deadlocks, shared memory, synchronizing access, etc.

In Java, it is sufficient to understand:

1. How to create and start threads
2. The `Runnable` interface and the `Thread` class, and their methods.
3. The `sleep`, `start`, `run` and `join` methods.
4. The `volatile` keyword
5. The `synchronized` keyword

Nested Classes

> Nested classes are divided into two categories: static and non-static. Nested classes that are declared static are called static nested classes. Non-static nested classes are called inner classes.

Nested Classes - Oracle Tutorial