volatile keyword in Java

why, when, and how to use it

Alexandru Nastase · Follow

Published in CodeX · 3 min read · Nov 8, 2022

114

Modifiers are specific keywords present in Java using that we can make changes to the characteristics of a variable, method, or class and limit its scope. Java programming language has a rich set of Modifiers.

Modifiers in Java are divided into two types — **Access Modifiers** and **Non-Access modifiers.**

Non-Access modifiers

Non-access modifiers provide information about the characteristics of a class, method, or variable to the JVM. Seven types of Non-Access modifiers are present in Java. They are –

1. **final**
2. **static**
3. **abstract**
4. **synchronized**
5. **volatile**
6. **transient**
7. **native**

This article will go into the details of the **volatile** keyword. Let’s explore first, why do we use it, when should we use it if we decide to, and last but not least, how to use it.

the WHY:

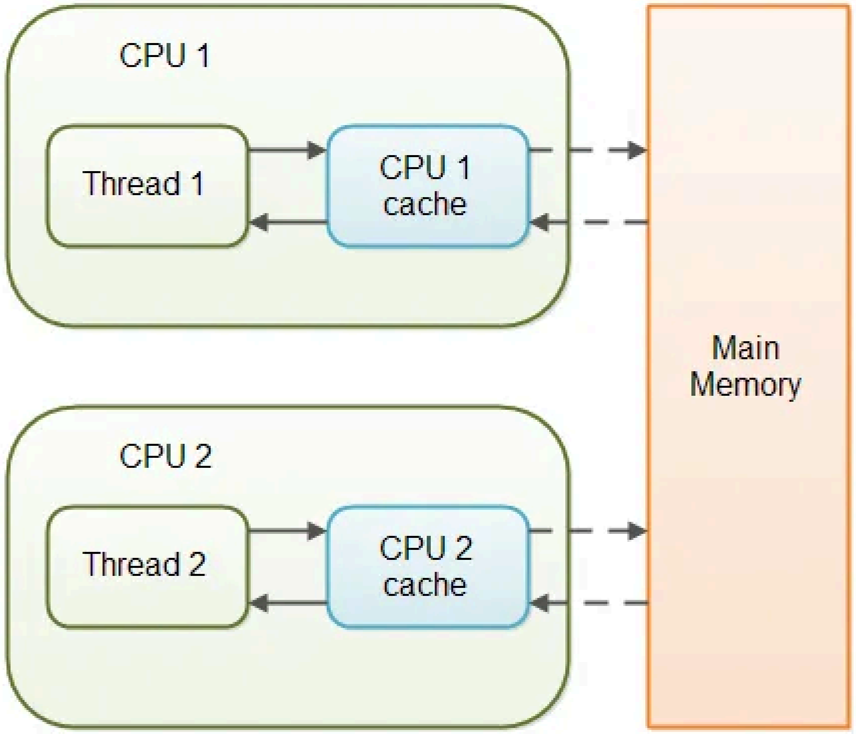
The volatile keyword in Java is used to mark a Java variable as “being stored in main memory”.

Every thread that accesses a volatile variable will read it from main memory, and not from the CPU cache. This way, all threads see the same value for the volatile variable.

The volatile keyword can be used with variables of any kind, including primitive types and object references. When used with objects, all threads will see the most up-to-date values for all fields in the object.

The volatile keyword is often used with flags that indicate that a thread needs to stop running. For example, a thread might have a boolean flag called “done”, and when another thread sets this flag to “true”, the first thread will know to stop running. Without the volatile keyword, the first thread might keep running indefinitely, because it would never see the updated value of the “done” flag.

There are other uses for the volatile keyword as well, but this is one of the most common. In general, if you need to make sure that all threads see the same value for a certain variable, then you should mark that variable as being volatile.



the WHEN and HOW:

1. volatile field

```java
public class Singleton {
    private static volatile Singleton _instance; // volatile variable
    public static Singleton getInstance() {
        if (_instance == null) {
            synchronized (Singleton.class) {
                if (_instance == null)
                    _instance = new Singleton();
            }
        }
        return _instance;
    }
}
```

volatile_field.java hosted with ❤ by **GitHub** **view raw**

In the above example, we are creating instance lazily at the time the first request comes.

If we do not make the `_instance` variable `volatile` then the Thread which is creating the instance of `Singleton` is not able to communicate to the other thread. So if Thread A is creating Singleton instance and just after creation, the CPU corrupts etc, all other threads will not be able to see the value of `_instance` as not null and they will believe it is still assigned null.

Why does this happen? Because reader threads are not doing any locking and until the writer thread comes out of a synchronized block, the memory will not be synchronized and value of `_instance` will not be updated in main memory. With the Volatile keyword in Java, this is handled by Java itself and such updates will be visible by all reader threads.

Knowing best why, how and when to use **volatile**, as well as other modifiers, comes with experience, and experience comes with time.

Enjoy the journey and happy coding!

> Thank you for your time, dear reader!

> Some of my articles:

10 best practices in Java
Hashing in JAVA
final Keyword in Java
The struggle of getting my first job as a software developer
Will Java still be worth in 2023? ☕

The BLUB Paradox

Things I wish I knew as a beginner programmer

Spring Boot: The GOOD and the BAD

Why do we love coding? 💻

Stay motivated

AI is bullshit

100 followers done. What’s next?

Is work from home going to end?

Java Tech Technology Volatile Programming

Written by Alexandru Nastase

Follow

429 Followers · Writer for CodeX

Java guy. Life newbie. Maybe smart maybe funny. Personal stuff here, too. WELCOME ! ❤️

More from Alexandru Nastase and CodeX

Alexandru Nastase

Hashing in JAVA

What is hashing?

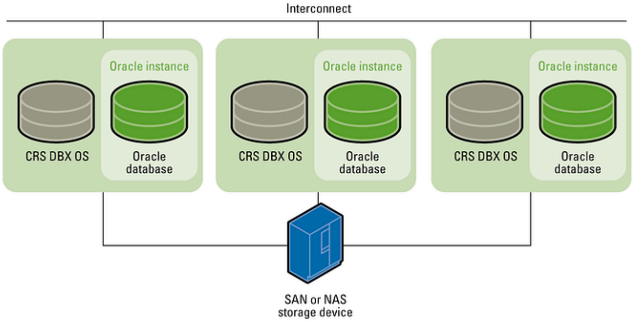
Sep 12, 2022 205 1

Enes Talay in CodeX

Stop Using find() Method in JavaScript

Forget the find() Method in JavaScript: Alternative Approaches for Cleaner Code

Apr 1 2.4K 38



Devansh in CodeX

How Pinterest Scaled to 11 Million Users With Only 6 Engineers

Scaling Pinterest—From 0 to 10s of Billions of Page Views a Month in Two Years

May 13 👏 1.1K 💬 8

Alexandru Nastase

10 best practices in Java

Java is a reliable and time-tested language. However, even the most experienced...

Aug 1, 2022 👏 710 💬 9

See all from Alexandru Nastase

See all from CodeX

Recommended from Medium

Firas Ahmed in Javarevisited

The difference between Interface & Abstract Class in Java

Interfaces and abstract classes are two key components in Java and Object Oriented...

✦ Apr 5 32 1

Gain Java Knowledge

When to Use a Parallel Stream in Java ?

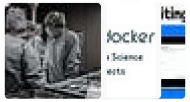
Using a parallel stream in Java can significantly enhance the performance of a...

✦ May 21 13

Lists

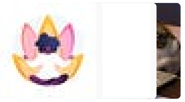
General Coding Knowledge

20 stories · 1339 saves



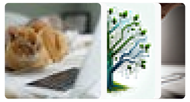
Coding & Development

11 stories · 677 saves



Stories to Help You Grow as a Software Developer

19 stories · 1167 saves



ChatGPT prompts

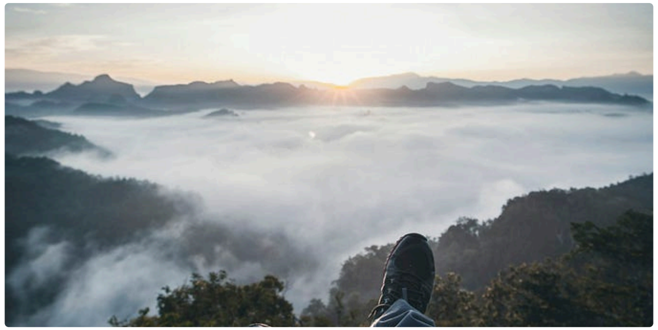
48 stories · 1737 saves

Sea Breeze in Towards Dev

Mastering Java Concurrent Programming(3): Java Thread...

In this article, we'll take a deep dive into the six states of Java threads and how they...

Feb 1 31

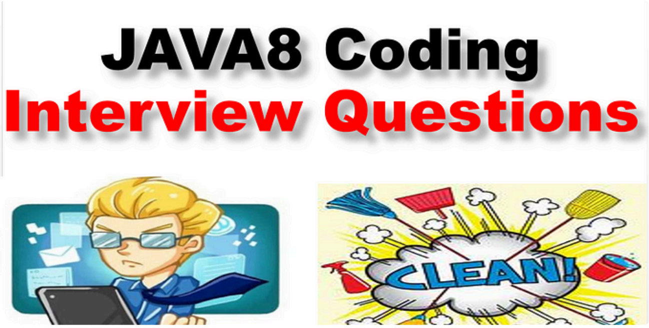


Oliver Foster in Stackademic

JAVA: How to Optimize 10,000 If-Else Statements in a Project?

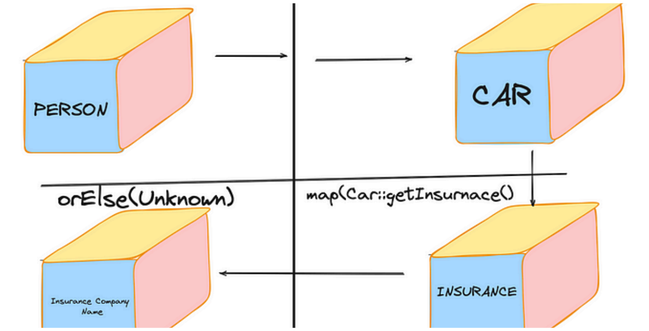
My article is open to everyone; non-member readers can click this link to read the full text.

May 24 370 7



Gain Java Knowledge

Java 8 coding interview questions



Vikas Taank

Essentials Java Interview Questions

In this tutorial, I am sharing the some Java 8 interview coding questions with answers. I...

✦ May 4 33

In this article I am trying to consolidate one of the most important questions and concepts...

✦ Jun 15 72

See more recommendations