**Julio Pochet - M8 Discussion Post – Thread Class and StringBuilder vs StringBuffer**

This week I finally got what multithreading is about: it lets your program do multiple things at the same time. Java has a few options, and the **Thread class** is one of the simplest to use.

To use it, you create a class that *extends* Thread and put your code in the run() method. Then in main(), you create instances and call .start(), not .run(). It felt a bit weird at first, but here's a quick example:

public class MyThread extends Thread {

public void run() {

System.out.println("Running: " + Thread.currentThread().getName());

}

public static void main(String[] args) {

MyThread t1 = new MyThread();

MyThread t2 = new MyThread();

t1.start();

t2.start();

}

}

This prints both threads doing their jobs, and the order changes each run, showing they run at the same time.  
**👍 Pros:** Super easy to set up for basics.  
**👎 Cons:** Since you're extending Thread, you can’t extend any other class, which is a limitation.

Next up: **StringBuilder vs. StringBuffer**. These classes help build strings, especially useful when you're adding text in loops.

* StringBuilder is **faster** but **not thread-safe**.
* StringBuffer is **thread-safe** (uses synchronization) but a bit slower.

Here’s a quick test:

public class StringExample {

public static void main(String[] args) {

StringBuilder sb = new StringBuilder();

sb.append("Hello");

sb.append(" ");

sb.append("World!");

System.out.println("StringBuilder: " + sb.toString());

StringBuffer sbf = new StringBuffer();

sbf.append("Java");

sbf.append(" ");

sbf.append("Threads!");

System.out.println("StringBuffer: " + sbf.toString());

}

}

It shows both do the job, but if multiple threads were modifying your string, StringBuffer would protect you from race issues. If there's only one thread, StringBuilder is way faster.

**❓ Question for the class:**

Have you ever had a situation where using StringBuilder caused issues in a multi-threaded context? Would you always choose speed or pick safety?

**📚 Sources:**

* GeeksforGeeks – [Java Multithreading Tutorial](https://www.geeksforgeeks.org/java/java-multithreading-tutorial/)
* Oracle Docs – [StringBuffer (Java SE 8)](https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuffer.html)
* Oracle Docs – [StringBuilder (Java SE 8)](https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html)