Paste HTML element by inspecting here!

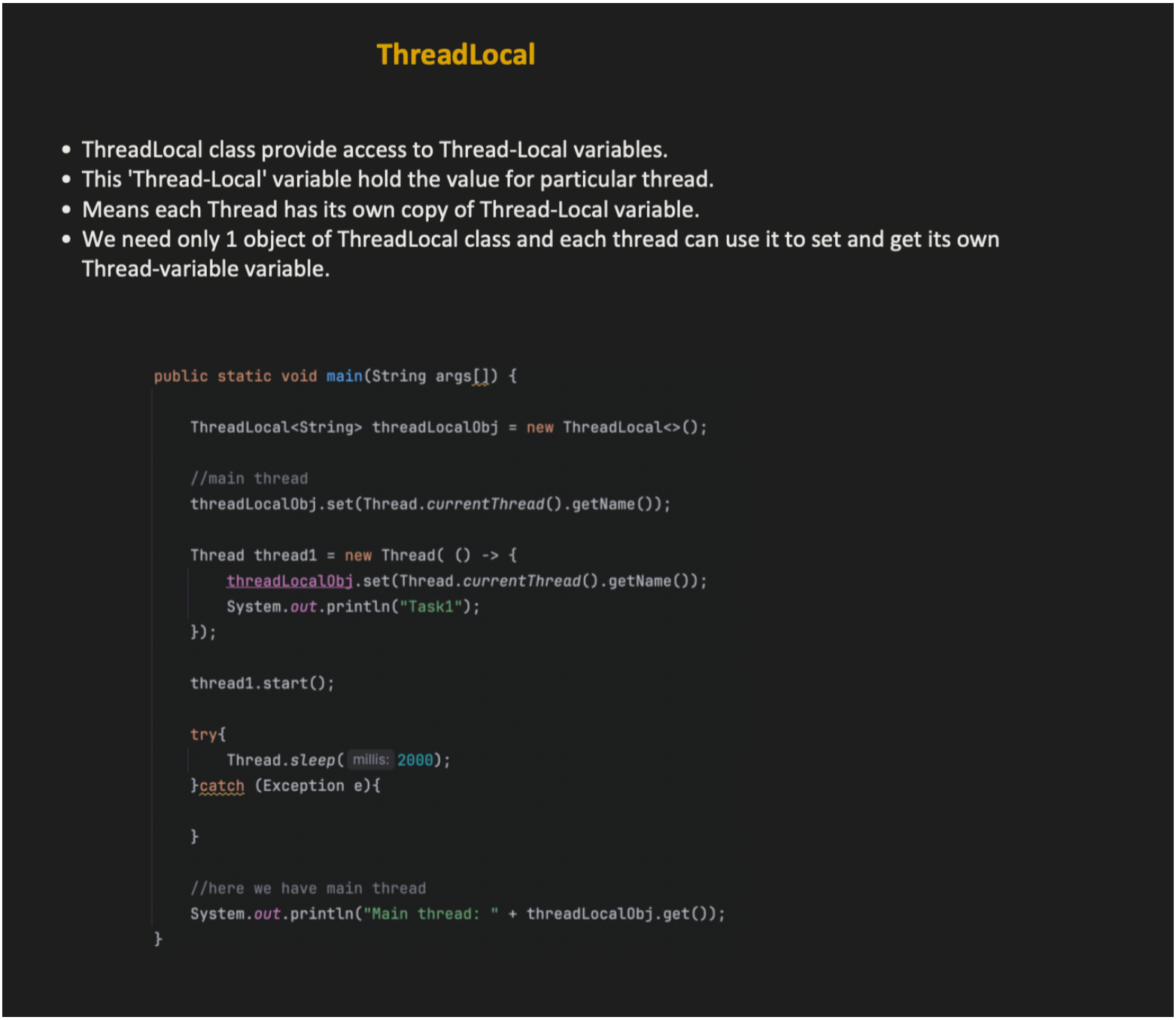
es/74tdoed98f7cfbc5a4c388f3b296c4cf82995" class="notecardImageClass"
d="74tdoed98f7cfbc5a4c388f3b296c4cf82995">

/div>

Download as PDF

Java: VirtualThreads and ThreadLocal

"Concept && Coding" YT Video Notes



Remember to clean up, if reusing the thread

```
public static void main(String args[]) {

    ThreadLocal<String> threadLocalObj = new ThreadLocal<>();

    ExecutorService poolObj = Executors.newFixedThreadPool( nThreads: 5);

    poolObj.submit(() -> {
        threadLocalObj.set(Thread.currentThread().getName());
    });

    for(int i=1; i<15; i++){
        poolObj.submit(() -> {
            System.out.println(threadLocalObj.get());
        });
    }
}
```

Output:

```
null
pool-1-thread-1
null
pool-1-thread-1
null
null
null
null
null
pool-1-thread-1
null
null
null
```

```
public static void main(String args[]) {

    ThreadLocal<String> threadLocalObj = new ThreadLocal<>();

    ExecutorService poolObj = Executors.newFixedThreadPool( nThreads: 5);

    poolObj.submit(() -> {
        threadLocalObj.set(Thread.currentThread().getName());
        //my work completed, now clean up
        threadLocalObj.remove();
    });

    for(int i=1; i<15; i++){
        poolObj.submit(() -> {
            System.out.println(threadLocalObj.get());
        });
    }
}
```

Output:

```
null
null
null
null
null
null
null
null
null
null
null
null
null
null
```

Virtual Thread vs Platform Thread

Moto of Virtual Thread:
To get Higher throughput not Latency

```
Thread th1 = Thread.ofVirtual().start(RunnableTask);
Or
 ExecutorService myExecutorObj = Executors.newVirtualThreadPerTaskExecutor()
myExecutorObj .submit(RunnableTask)
```