Java Threads - Notes

1What is a Thread?

A thread is a lightweight process that allows concurrent execution in Java.

2Ways to Create a Thread

1. Extending `Thread` class

```
class MyThread extends Thread {
    public void run() {
        System.out.println("Thread is running...");
    }
}
MyThread t1 = new MyThread();
t1.start();
```

2. Implementing `Runnable` interface

```
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Thread is running...");
    }
}
Thread t1 = new Thread(new MyRunnable());
t1.start();
```

3. Using Anonymous Class

```
Thread t1 = new Thread(new Runnable() {
    public void run() {
        System.out.println("Thread is running...");
    }
});
t1.start();
```

4. Using Lambda Expression (Java 8+)

```
Thread t1 = new Thread(() -> System.out.println("Thread is running..."));
t1.start();
```

3Thread Methods

- `start()`: Starts the thread.
- `run()`: Defines the task inside the thread.
- `sleep(ms)`: Pauses thread for `ms` milliseconds.
- `join()`: Waits for the thread to finish execution.
- `yield()`: Allows other threads to execute first.

- `setName(name)`: Sets thread name.
- `getName()`: Gets thread name.
- `setPriority(int)`: Sets priority (1 to 10).
- `getPriority()`: Gets thread priority.
- `isAlive()`: Checks if thread is running.
- `setDaemon(true)`: Sets thread as daemon.
- `isDaemon()`: Checks if thread is daemon.
- `interrupt()`: Interrupts a sleeping or waiting thread.

4Example: `sleep()`, `join()`

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
            System.out.println(i);
        }
    }
}
MyThread t1 = new MyThread();
t1.start();
t1.join();
```

5Thread Priority

- `MIN_PRIORITY = 1`
- `NORM_PRIORITY = 5` (default)
- `MAX_PRIORITY = 10`

```
Thread t1 = new Thread();
t1.setPriority(Thread.MAX_PRIORITY);
```

6Daemon Threads

- Daemon threads run in the background.
- Example: Garbage Collector.

```
class DaemonExample extends Thread {
    public void run() {
        System.out.println("Daemon thread running...");
    }
}
DaemonExample t1 = new DaemonExample();
t1.setDaemon(true);
t1.start();
```

7Thread Synchronization

Without Synchronization (Race Condition)

```
class Counter {
    int count = 0;
    void increment() { count++; }
}
```

With Synchronization

```
class Counter {
    int count = 0;
    synchronized void increment() { count++; }
}
```

- **Creating Threads:** `Thread` class or `Runnable` interface.
- **Methods:** `start()`, `sleep()`, `join()`, `yield()`.
- **Priorities:** `setPriority()`, `getPriority()`.
- **Daemon Threads:** `setDaemon(true)`.
- **Synchronization:** Use `synchronized` to avoid race conditions.