

Multithreading in Java

Take a look at [this article](#) by Oracle for more information about concurrency and threads. Below you will find information important for robot programs written in WPILib that will cause unexplained errors.

Threads

The code below will never be able to exit! It will only stop when the entire JVM stops. There is no way in Java to stop a thread unless the thread exits by itself.

Bad Example

```
public class Robot extends IterativeRobot {
    public void robotInit() {
        Thread t = new Thread(() -> {
            while (true) {
                // We are stuck here
            }
        });
        t.start();
    }
}
```

We can solve this problem by setting a flag. In Java every thread has a flag designed for this. We just need to modify our code to check that flag. Take a look at this example:

```
public class Robot extends IterativeRobot {
    public void robotInit() {
        Thread t = new Thread(() -> {
            while (!Thread.interrupted()) {
                // Not stuck anymore!
            }
        });
        t.start();
    }
}
```

Multithreading in Java

```
}
```

Everytime we run the loop, we check the interrupted flag to see if we should continue to execute.