Java 线程

Thread

Thread

```
public class MyThread extends Thread{

    @Override
    public void run() {
        // 要执行的代码
        System.out.println(Thread.currentThread().getName());
    }
}
```

```
Thread t = new MyThread();
t.setName("T-1");
t.start();      //这启动了一个新线程，会打印线程名字   T-1
```

Thread

```java
public class MyThread extends Thread{

    @Override
    public void run() {
        // 要执行的代码
        System.out.println(Thread.currentThread().getName());
    }
}
```

```java
Thread t = new MyThread();
t.setName("T-1");
t.start();      //这启动了一个新线程，会打印线程名字   T-1
t.run();        //没有启动新线程，还是在当前线程内运行；不会打印 T-1
```

Thread

```java
public class MyThread extends Thread{

    @Override
    public void run() {
        // 要执行的代码
        System.out.println(Thread.currentThread().getName());
    }
}
```

```java
Thread t = new MyThread();
t.setName("T-1");
t.start();      //这启动了一个新线程，会打印线程名字  T-1
t.run();        //没有启动新线程，还是在当前线程内运行；不会打印 T-1
```

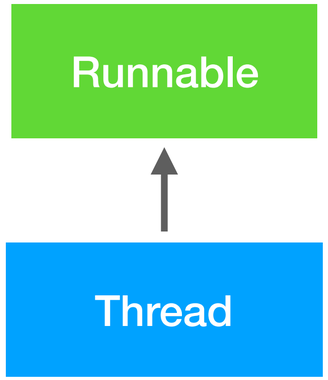
Thread

```java
public class MyThread extends Thread{

    @Override
    public void run() {
        // 要执行的代码
        System.out.println(Thread.currentThread().getName());
    }
}
```

```java
Thread t = new MyThread();
t.setName("T-1");
t.start();      //这启动了一个新线程，会打印线程名字   T-1
t.run();        //没有启动新线程，还是在当前线程内运行；不会打印 T-1
```

Runnable

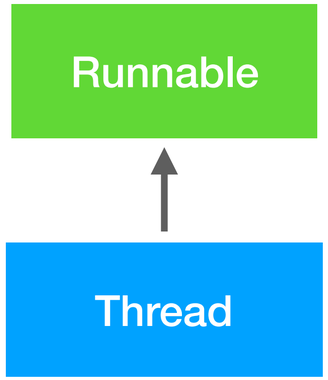



```
public class MyTask implements Runnable{

    @Override
    public void run(){
        // 要执行的任务写这里
        System.out.println(Thread.currentThread().getName());
    }
}

MyTask task = new MyTask();
Thread t1 = new Thread(task, “T1”);
t1.start();
t1.run();
task.run();
```

Runnable

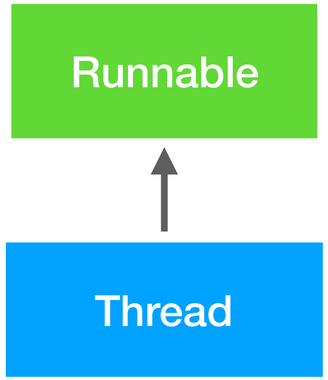



```
public class MyTask implements Runnable{

    @Override
    public void run(){
        // 要执行的任务写这里
        System.out.println(Thread.currentThread().getName());
    }
}

MyTask task = new MyTask();
Thread t1 = new Thread(task, “T1”);
t1.start();
t1.run();
task.run();
```

Runnable

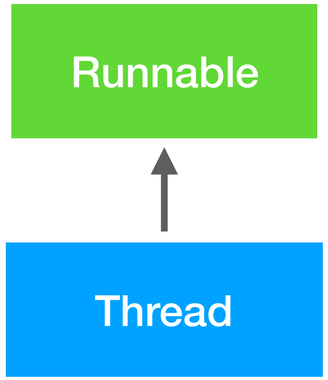



```
public class MyTask implements Runnable{

    @Override
    public void run(){
        // 要执行的任务写这里
        System.out.println(Thread.currentThread().getName());
    }
}

MyTask task = new MyTask();
Thread t1 = new Thread(task, “T1”);
t1.start();
t1.run();
task.run();
```

Runnable




```
public class MyTask implements Runnable{

    @Override
    public void run(){
        // 要执行的
        System.out
    }
}

MyTask task = new
Thread t1 = new Th
t1.start();
t1.run();
task.run();
```

```
Thread t1 = new Thread(new Runnable() {
    @Override
    public void run() {
        //线程内执行的代码写这
    }
}, "线程名字");
```

```
Thread t2 = new Thread(() -> {
    //线程内执行的代码写这
}, "线程名字");
t2.start();
```

Thread vs. Runnable

```
Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++) {
            list.add(Thread.currentThread().getName() + i);
        }
        System.out.println(String.join(",", list));
    }

};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
```

Thread vs. Runnable

```
Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++) {
            list.add(Thread.currentThread().getName() + i);
        }
        System.out.println(String.join(",", list));
    }

};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
```

A0,A1,A2　　B0,B1,B2　　C0,C1,C2　　中的一组, list.size() == 3

Thread vs. Runnable

```
Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++) {
            list.add(Thread.currentThread().getName() + i);
        }
        System.out.println(String.join(",", list));
    }

};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
```

A0,A1,A2　　B0,B1,B2　　C0,C1,C2　　中的一组, list.size() == 3

A0,A1,A2,B0,B1,B2,C0,C1,C2 共9个，但不保证ABC的顺序, list.size() == 9

Thread vs. Runnable

```
Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++) {
            list.add(Thread.currentThread().getName() + i);
        }
        System.out.println(String.join(",", list));
    }

};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
```

A0,A1,A2　　B0,B1,B2　　C0,C1,C2　　中的一组, list.size() == 3

A0,A1,A2,B0,B1,B2,C0,C1,C2 共9个，但不保证ABC的顺序, list.size() == 9

Thread vs. Runnable

```
Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++) {
            list.add(Thread.currentThread().getName() + i);
        }
        System.out.println(String.join(",", list));
    }

};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
```

A0,A1,A2　　B0,B1,B2　　C0,C1,C2　　中的一组, list.size() == 3

A0,A1,A2,B0,B1,B2,C0,C1,C2 共9个，但不保证ABC的顺序, list.size() == 9

ThreadName

- thread.setName(String threadName)
- new Thread(Runnable runnable, String threadName)
- 日志里可以打印出来ThreadName，方便我们调试

ThreadName

- thread.setName(String threadName)
- new Thread(Runnable runnable, String threadName)
- 日志里可以打印出来ThreadName，方便我们调试

logback

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appender name="out" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
    <!- 其余略 ->
</configuration>
```

ThreadName

```
2019-03-05 09:43:58.488  INFO 2297 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized
2019-03-05 09:43:58.590  INFO 2297 --- [  restartedMain] o.s.web.context.ContextLoader            : Root WebApplication
2019-03-05 09:43:58.862  INFO 2297 --- [  restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing Execut
2019-03-05 09:43:58.873  INFO 2297 --- [  restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing Execut
2019-03-05 09:43:59.157  INFO 2297 --- [  restartedMain] o.s.s.c.ThreadPoolTaskScheduler          : Initializing Execut
2019-03-05 09:43:59.184  INFO 2297 --- [  restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing Execut
2019-03-05 09:43:59.972  INFO 2297 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server i
2019-03-05 09:44:00.007  INFO 2297 --- [  restartedMain] o.s.m.s.b.SimpleBrokerMessageHandler     : Starting...
2019-03-05 09:44:00.007  INFO 2297 --- [  restartedMain] o.s.m.s.b.SimpleBrokerMessageHandler     : BrokerAvailabilityE
 ], registry[0 sessions]]]]
2019-03-05 09:44:00.008  INFO 2297 --- [  restartedMain] o.s.m.s.b.SimpleBrokerMessageHandler     : Started.
2019-03-05 09:44:00.017 TRACE 2297 --- [MessageBroker-2] cn.devmgr.cp.ScheduledTasks              : reportCurrentTimeTo
2019-03-05 09:44:00.017 TRACE 2297 --- [MessageBroker-1] cn.devmgr.cp.ScheduledTasks              : current time is 09:
2019-03-05 09:44:00.047  INFO 2297 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on p
2019-03-05 09:44:00.053  INFO 2297 --- [  restartedMain] cn.devmgr.cp.Application                 : Started Application
2019-03-05 09:44:21.461  INFO 2297 --- [       Thread-8] o.s.m.s.b.SimpleBrokerMessageHandler     : Stopping...
2019-03-05 09:44:21.462  INFO 2297 --- [       Thread-8] o.s.m.s.b.SimpleBrokerMessageHandler     : BrokerAvailabilityE
 ], registry[0 sessions]]]]
2019-03-05 09:44:21.462  INFO 2297 --- [       Thread-8] o.s.m.s.b.SimpleBrokerMessageHandler     : Stopped.
2019-03-05 09:44:21.465  INFO 2297 --- [       Thread-8] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down Execu
2019-03-05 09:44:21.466  INFO 2297 --- [       Thread-8] o.s.s.c.ThreadPoolTaskScheduler          : Shutting down Execu
2019-03-05 09:44:21.468  INFO 2297 --- [       Thread-8] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down Execu
2019-03-05 09:44:21.468  INFO 2297 --- [       Thread-8] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down Execu
```

```
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
    <!- 其余略 ->
</configuration>
```

ThreadName

- thread.setName(String threadName)
- new Thread(Runnable runnable, String threadName)
- 日志里可以打印出来ThreadName，方便我们调试

logback

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appender name="out" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
    <!- 其余略 ->
</configuration>
```