

线程的优先级指的是，线程的优先级越高越有可能先执行，但仅仅是有可能而已。

1. 设置优先级

```
public final void setPriority(int newPriority)
```

```
class MyThread implements Runnable {
    private boolean flag = true;
    @Override
    public void run() {
        System.out.println("hello");
    }
}

public class Test {
    public static void main(String[] args) throws InterruptedException {
        MyThread myThread = new MyThread();
        Thread thread = new Thread(myThread);
        // 设置线程的优先级为 5
        thread.setPriority(5);
        thread.start();
    }
}
```

我们可以使用 Thread类 提供的几个常量进行优先级的设置。

```
// 最高优先级
public final static int MAX_PRIORITY = 10;
// 中等优先级
public final static int NORM_PRIORITY = 5;
// 最低优先级
public final static int MIN_PRIORITY = 1;
```

2. 取得优先级

```
public final int getPriority()
```

```
public class Test12 {
    public static void main(String[] args) throws InterruptedException {
        MyThread myThread = new MyThread();
        Thread thread = new Thread(myThread);
        thread.setPriority(Thread.MAX_PRIORITY);
        // 取得线程的优先级并打印
        System.out.println(thread.getPriority());
    }
}
```

主方法也是一个线程，主线程的优先级为5。

```
int priority = Thread.currentThread().getPriority();
```

3. 线程具有继承性

线程之间是具有继承关系的，比如在A线程中启动B线程，那么此时启动的B线程的优先级与A优先级一样。

```
class A implements Runnable{
    @Override
    public void run() {
        System.out.println("A的优先级为:" + Thread.currentThread().getPriority());
        Thread thread = new Thread(new B());
        thread.start();
    }
}

class B implements Runnable {
    @Override
    public void run() {
        System.out.println("B的优先级为:" + Thread.currentThread().getPriority());
    }
}

public class Test {
    public static void main(String[] args) throws InterruptedException {
        Thread thread = new Thread(new A());
        thread.setPriority(Thread.MAX_PRIORITY);
        thread.start();

        Thread.sleep(1000);
        System.out.print("单独启动的线程B: ");
        new Thread(new B()).start();
    }
}
```

运行结果为：

