**Thread 方法**

下表列出了Thread类的一些重要方法：

序号 方法描述

1 public void start()

使该线程开始执行；Java 虚拟机调用该线程的 run 方法。

2 public void run()

如果该线程是使用独立的 Runnable 运行对象构造的，则调用该 Runnable 对象的 run 方法；否则，该方法不执行任何操作并返回。

3 public final void setName(String name)

改变线程名称，使之与参数 name 相同。

4 public final void setPriority(int priority)

更改线程的优先级。

5 public final void setDaemon(boolean on)

将该线程标记为守护线程或用户线程。

6 public final void join(long millisec)

等待该线程终止的时间最长为 millis 毫秒。

7 public void interrupt()

中断线程。

8 public final boolean isAlive()

测试线程是否处于活动状态。

上述方法是被 Thread 对象调用的，下面表格的方法是 Thread 类的静态方法。

序号 方法描述

1 public static void yield()

暂停当前正在执行的线程对象，并执行其他线程。

2 public static void sleep(long millisec)

在指定的毫秒数内让当前正在执行的线程休眠（暂停执行），此操作受到系统计时器和调度程序精度和准确性的影响。

3 public static boolean holdsLock(Object x)

当且仅当当前线程在指定的对象上保持监视器锁时，才返回 true。

4 public static Thread currentThread()

返回对当前正在执行的线程对象的引用。

5 public static void dumpStack()

将当前线程的堆栈跟踪打印至标准错误流。

实例

如下的ThreadClassDemo 程序演示了Thread类的一些方法：

DisplayMessage.java 文件代码：

// 文件名 : DisplayMessage.java

// 通过实现 Runnable 接口创建线程

public class DisplayMessage implements Runnable {

private String message;

public DisplayMessage(String message) {

this.message = message;

}

public void run() {

while(true) {

System.out.println(message);

}

}

}

GuessANumber.java 文件代码：

// 文件名 : GuessANumber.java

// 通过继承 Thread 类创建线程

public class GuessANumber extends Thread {

private int number;

public GuessANumber(int number) {

this.number = number;

}

public void run() {

int counter = 0;

int guess = 0;

do {

guess = (int) (Math.random() \* 100 + 1);

System.out.println(this.getName() + " guesses " + guess);

counter++;

} while(guess != number);

System.out.println("\*\* Correct!" + this.getName() + "in" + counter + "guesses.\*\*");

}

}

ThreadClassDemo.java 文件代码：

// 文件名 : ThreadClassDemo.java

public class ThreadClassDemo {

public static void main(String [] args) {

Runnable hello = new DisplayMessage("Hello");

Thread thread1 = new Thread(hello);

thread1.setDaemon(true);

thread1.setName("hello");

System.out.println("Starting hello thread...");

thread1.start();

Runnable bye = new DisplayMessage("Goodbye");

Thread thread2 = new Thread(bye);

thread2.setPriority(Thread.MIN\_PRIORITY);

thread2.setDaemon(true);

System.out.println("Starting goodbye thread...");

thread2.start();

System.out.println("Starting thread3...");

Thread thread3 = new GuessANumber(27);

thread3.start();

try {

thread3.join();

}catch(InterruptedException e) {

System.out.println("Thread interrupted.");

}

System.out.println("Starting thread4...");

Thread thread4 = new GuessANumber(75);

thread4.start();

System.out.println("main() is ending...");

}

}

运行结果如下，每一次运行的结果都不一样。

Starting hello thread...

Starting goodbye thread...

Hello

Hello

Hello

Hello

Hello

Hello

Goodbye

Goodbye

Goodbye

Goodbye

Goodbye

.......