

BASICS OF JAVA**OOPS CONCEPTS****STRING HANDLING****EXCEPTION HANDLING****JAVA MULTITHREADING**

Introduction to Multithreading

Thread Class

Creating a thread

Joining threads

Sleeping Thread in Java

Naming Thread in Java

Thread Priority in Java

Daemon Thread in Java

Synchronization

Interthread Communication

Thread group

ADVANCED TOPICS**COLLECTION FRAMEWORK****JAVA GUI****REFLECTION API****RMI APPLICATION****INNER CLASS****WRAPPER CLASS****FILE HANDLING****LIST****SET****MAP****QUEUE & DEQUE****JDBC****LAYOUT MANAGERS**

ADVERTISEMENT

Java Naming Thread

ADVERTISEMENT

Each thread in Java has its own name which is set by the JVM by default. Although there are many other attributes associated to the thread like: id, priority etc.

We can get name of a thread by calling `getName()` method of Thread class. If we wish to set new name of the thread then `setName()` method can be used. Both methods belongs to Thread class and even we can set name of thread by passing into constructor during creating object. Lets understand by the examples.

Syntax of the two methods is given below.

```
setName(String name)
getName()
```

Example: Fetch Thread Name

lets first fetch the thread name set by the JVM. It is default name so initially we cannot predict it.

```
public class MyThread extends Thread
{
    public void run()
    {
        System.out.println("thread running...");
    }
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();

        t1.start();
        System.out.println("thread name: "+t1.getName());
    }
}
```

OUTPUT:

thread running...

thread name: Thread-0

we can also use static `currentThread()` method of Thread class. it returns thread name, priority etc. it is static method so object is not require to call it. See the below example.

ADVERTISEMENT

Example: Thread.currentThread()

In this example, we are using `currentThread` method of Thread class to get name of the thread.

```
public class MyThread extends Thread
{
    public void run()
    {
        System.out.println("thread running...");
        System.out.println("Thread name: "+Thread.currentThread());
    }
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();

        t1.start();
    }
}
```

OUTPUT:

thread running...

Thread Name: Thread[Thread-0,5,main]

Example: Setting Name

In this example, we are using `setName()` method to set name of the thread and `getName()` method to get name of the thread.

```
public class MyThread extends Thread
{
    public void run()
    {
        System.out.println("thread running...");
    }
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();
        t1.setName("mythread");
        System.out.println("Thread Name: "+t1.getName());
    }
}
```

OUTPUT:

thread running...

Thread Name: mythread

Since we are extending the Thread class so we call its constructor through the super call and passed thread name to the MyThread class.

Example: Setting Name

We can directly pass the thread name to the constructor by creating thread class object.

```
public class MyThread implements Runnable
{
    public void run()
    {
        System.out.println("thread running...");
    }
    public static void main(String[] args)
    {
        MyThread t1=new MyThread("mythread");
        t1.start();
        System.out.println("Thread Name: "+t1.getName());
    }
}
```

OUTPUT:

thread running...

Thread Name: mythread

In this topic, we learned to fetch thread name using `getName()` and `currentThread()` methods, and to set the name using `setName()` method and passing name to the constructor.

← Prev

Next →

ADVERTISEMENT

Java MCQ Tests

Prepare for Java Interview in TCS, Infosys, etc. companies.

Explore

Java Programs

Java programs with code and output for practice.

Explore

Spring Framework

Learn the most widely used Java framework in the World.

Explore

ADVERTISEMENT

ADVERTISEMENT

No compatible source was found for this media.

HelloXrends