

Priority of a Thread (Thread Priority)

Each thread has a priority. Priorities are represented by a number between 1 and 10. In most cases, the thread scheduler schedules the threads according to their priority (known as preemptive scheduling). But it is not guaranteed because it depends on JVM specification that which scheduling it chooses. Note that not only JVM a Java programmer can also assign the priorities of a thread explicitly in a Java program.

Setter & Getter Method of Thread Priority

Let's discuss the setter and getter method of the thread priority.

public final int getPriority(): The java.lang.Thread.getPriority() method returns the priority of the given thread.

public final void setPriority(int newPriority): The java.lang.Thread.setPriority() method updates or assign the priority of the thread to newPriority. The method throws IllegalArgumentException if the value newPriority goes out of the range, which is 1 (minimum) to 10 (maximum).

3 constants defined in Thread class:

- 1. public static int MIN_PRIORITY
- 2. public static int NORM_PRIORITY
- 3. public static int MAX_PRIORITY

Default priority of a thread is 5 (NORM_PRIORITY). The value of MIN_PRIORITY is 1 and the value of MAX_PRIORITY is 10.

Example of priority of a Thread:

FileName: ThreadPriorityExample.java

```
// Importing the required classes
import java.lang.*;
public class ThreadPriorityExample extends Thread
{
// Method 1
// Whenever the start() method is called by a thread
// the run() method is invoked
public void run()
// the print statement
System.out.println("Inside the run() method");
}
// the main method
public static void main(String argvs[])
// Creating threads with the help of ThreadPriorityExample class
ThreadPriorityExample th1 = new ThreadPriorityExample();
ThreadPriorityExample th2 = new ThreadPriorityExample();
ThreadPriorityExample th3 = new ThreadPriorityExample();
// We did not mention the priority of the thread.
// Therefore, the priorities of the thread is 5, the default value
// 1st Thread
// Displaying the priority of the thread
// using the getPriority() method
System.out.println("Priority of the thread th1 is: " + th1.getPriority());
// 2nd Thread
// Display the priority of the thread
```

```
System.out.println("Priority of the thread th2 is: " + th2.getPriority());
// 3rd Thread
// // Display the priority of the thread
System.out.println("Priority of the thread th2 is: " + th2.getPriority());
// Setting priorities of above threads by
// passing integer arguments
th1.setPriority(6);
th2.setPriority(3);
th3.setPriority(9);
// 6
System.out.println("Priority of the thread th1 is: " + th1.getPriority());
// 3
System.out.println("Priority of the thread th2 is: " + th2.getPriority());
// 9
System.out.println("Priority of the thread th3 is: " + th3.getPriority());
// Main thread
// Displaying name of the currently executing thread
System.out.println("Currently Executing The Thread: " + Thread.currentThread().getName());
System.out.println("Priority of the main thread is: " + Thread.currentThread().getPriority());
// Priority of the main thread is 10 now
Thread.currentThread().setPriority(10);
System.out.println("Priority of the main thread is: " + Thread.currentThread().getPriority());
}
}
```

Output:

```
Priority of the thread th1 is : 5
Priority of the thread th2 is : 5
```

```
Priority of the thread th2 is : 5

Priority of the thread th1 is : 6

Priority of the thread th2 is : 3

Priority of the thread th3 is : 9

Currently Executing The Thread : main

Priority of the main thread is : 5

Priority of the main thread is : 10
```

We know that a thread with high priority will get preference over lower priority threads when it comes to the execution of threads. However, there can be other scenarios where two threads can have the same priority. All of the processing, in order to look after the threads, is done by the Java thread scheduler. Refer to the following example to comprehend what will happen if two threads have the same priority.

FileName: ThreadPriorityExample1.java

```
// importing the java.lang package
import java.lang.*;
public class ThreadPriorityExample1 extends Thread
{
// Method 1
// Whenever the start() method is called by a thread
// the run() method is invoked
public void run()
// the print statement
System.out.println("Inside the run() method");
}
// the main method
public static void main(String argvs[])
// Now, priority of the main thread is set to 7
Thread.currentThread().setPriority(7);
// the current thread is retrieved
// using the currentThread() method
```

```
// displaying the main thread priority
// using the getPriority() method of the Thread class
System.out.println("Priority of the main thread is:" + Thread.currentThread().getPriority());

// creating a thread by creating an object of the class ThreadPriorityExample1
ThreadPriorityExample1 th1 = new ThreadPriorityExample1();

// th1 thread is the child of the main thread
// therefore, the th1 thread also gets the priority 7

// Displaying the priority of the current thread
System.out.println("Priority of the thread th1 is:" + th1.getPriority());
}
}
```

Output:

```
Priority of the main thread is : 7
Priority of the thread th1 is : 7
```

Explanation: If there are two threads that have the same priority, then one can not predict which thread will get the chance to execute first. The execution then is dependent on the thread scheduler's algorithm (First Come First Serve, Round-Robin, etc.)

Example of IllegalArgumentException

We know that if the value of the parameter *newPriority* of the method getPriority() goes out of the range (1 to 10), then we get the IllegalArgumentException. Let's observe the same with the help of an example.

FileName: IllegalArgumentException.java

```
// importing the java.lang package
import java.lang.*;
public class IllegalArgumentException extends Thread
{
// the main method
public static void main(String argvs[])
// Now, priority of the main thread is set to 17, which is greater than 10
Thread.currentThread().setPriority(17);
// The current thread is retrieved
// using the currentThread() method
// displaying the main thread priority
// using the getPriority() method of the Thread class
System.out.println("Priority of the main thread is: " + Thread.currentThread().getPriority());
}
}
```

When we execute the above program, we get the following exception:

```
Exception in thread "main" java.lang.IllegalArgumentException
    at java.base/java.lang.Thread.setPriority(Thread.java:1141)
    at IllegalArgumentException.main(IllegalArgumentException.java:12)
```

← Prev

 $Next \rightarrow$

Youtube For Videos Join Our Youtube Channel: Join Now

Feedback

Send your Feedback to feedback@javatpoint.com

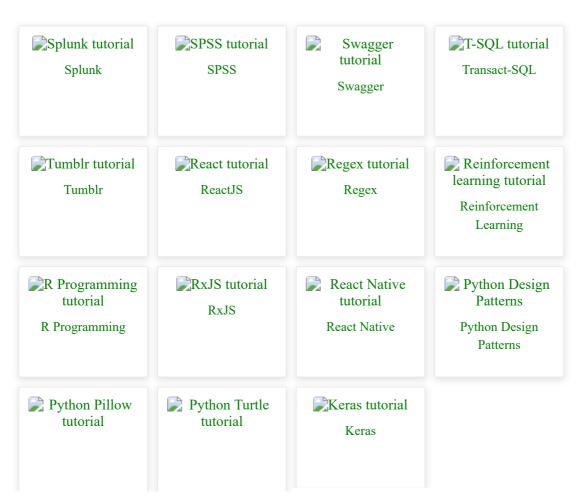
Help Others, Please Share







Learn Latest Tutorials

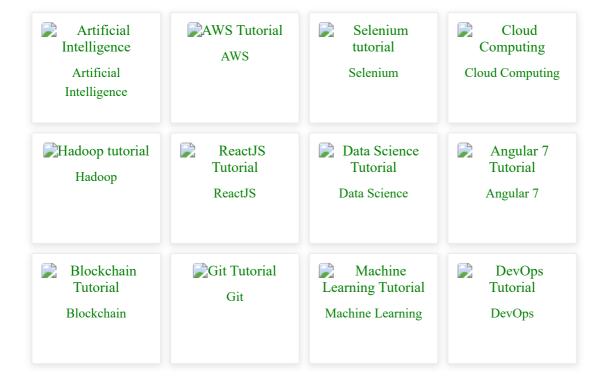


Python Turtle

Preparation



Trending Technologies



B.Tech / MCA

