**Naming a Thread on Multithreading**

**Definition:** The Thread class provides methods to change and get the name of a thread. By default, each thread has a name, i.e. thread-0, thread-1 and so on. By we can change the name of the thread by using the setName() method. The syntax of setName() and getName() methods are given below:

1. **public** String getName(): is used to **return** the name of a thread.
2. **public** **void** setName(String name): is used to change the name of a

thread.

We can also set the name of a thread directly when we create a new thread using the constructor of the class.

**Feature:**

* Naming a thread is very useful in identifying a thread and in debugging a code
* Every thread has a name for identification purposes. More than one thread may have the same name. If a name is not specified when a thread is created, a new name is generated for it.

**Question and Answers:**

1. **How to set the name of a thread?**

Ans: We can name a thread by using the method known as setName()

replacing default naming which was Thread-0, Thread-1, and so on.

*thread\_class\_object.setName(“Name\_thread\_here”);*

1. **How to retrieve a name of the primary thread or main thread?**

Ans: First, get the reference of the main thread by using currentThread()

method of thread class. currentThread() method returns the reference

of currently executing thread. After, getting the reference of the main

thread, use the getName() method to retrieve the name of the thread.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** ThreadsInJava  {  **public** **static** **void** main(String[] args)      {          Thread t = Thread.currentThread();            System.out.println(t.getName());      }  } |

Output: main

1. **Can we change the name of the main thread?**

Ans: Yes, we can change the name of the main thread. It can be done by first

getting the reference of the main thread by using currentThread()

method and then calling setName() method on it.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | **public** **class** ThreadsInJava  {  **public** **static** **void** main(String[] args)      {          Thread t = Thread.currentThread();            t.setName("main-thread-name-modified");            System.out.println(t.getName());      }  } |

Output: main-thread-name-modified

1. **Another method of Naming a thread in java?**

Ans: You can pass name of the thread while creating the object of thread.

There is a constructor in thread class which takes name of the thread

as an argument.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | **class** MyThread **extends** Thread  {  **public** MyThread(String name)      {  **super**(name);      }        @Override  **public** **void** run()      {          System.out.println(getName());      }  }    **public** **class** ThreadsInJava  {  **public** **static** **void** main(String[] args)      {          MyThread t = **new** MyThread("My Thread"); //passing the name while creating  the thread          t.start();      }  } |

Output: My Thread

1. **Can we change the name of a thread at any state of the thread?**

Ans: Yes, we can change the name of a thread at any state of the thread.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | **class** MyThread **extends** Thread  {      @Override  **public** **void** run()      {  **for**(**int** i = 0; i < 1000; i++)          {              System.out.println(i);          }      }  }    **public** **class** ThreadsInJava  {  **public** **static** **void** main(String[] args)      {          MyThread thread = **new** MyThread();            thread.setName("My Thread");        //Changing the name of the thread before  starting the thread          thread.start();            thread.setName("My Thread");        //Changing the name of the thread when  thread is active    **try**          {              thread.sleep(5000);          }  **catch** (InterruptedException e)          {              e.printStackTrace();          }            thread.setName("My Thread");     //Changing the name of the thread when thread  is sleeping      }  } |