DAY-60 MULTI\_THREADING-3

------ --------------------------------

which is the best approch to create the thread?

--> There are two approaches to crate the thread.

1. by extending thread class

2. by implementing runnable interface

among this two the best option is implementing runnable interface because:

case-1

------

class MyThread extends Thread

{

}

case-2

-------

class MyThread implements Runnable

{

}

-->

Advantage of implementing runnable interface over the extending thread class is:

If we inherit thread class then we wont be able to extend any other class.But If we implement runnable interface still we can extend any other class

so, This helps us to achive multiple- inheritance.

THREAD LIFE CYCLE:

-----------------

human life cycle:

-----------------

baby ---> child --> teenager ---> adult ---> old age

refer fig:1

1. New/Born state: when we create Thread class object in java application then Thread will come to new/born state.

2. Ready/Runnable state : When we access start() method Thread schedular has to assign system resources like memory and time here before calling

the run method.

3. Running state : calling the start() method will internally calls run method and run method will execute to perform job of the thread.

4. Dead/Destroy state : accessin the stop() method to stop the running thread is called as destroying the thread where as if the thread completes

its job then it will destory itself callled as dead state.

5. Blocked state : Any interruption to the thread execution leads to blocking of thread

few reasons to this blocking state is :

1. thread is in wait mode

2. system resorces are not available

3. when ever we use sleep() , wait (), notify () and so on ...

Thread class Library:

----------------------

1. Thread class constructor

// EXAMPLE-1

-------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

Thread t= new Thread();

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[Thread-0,5,main]

Thread-0 is name of the thread

5 is thread priority

main is thread group name

// EXAMPLE-2

--------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

Thread t= new Thread("myThread");

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[myThread,5,main]

myThread is name of the thread

5 is thread priority

main is thread group name

\*/

// EXAMPLE-3

--------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

Runnable r = new Thread();

Thread t= new Thread(r);

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[Thread-1,5,main]

Thread-1 is name of the thread

5 is thread priority

main is thread group name

\*/

// EXAMPLE-4

--------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

Runnable r = new Thread();

Thread t= new Thread(r,"myThread");

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[myThread,5,main]

myThread is name of the thread

5 is thread priority

main is thread group name

\*/

// EXAMPLE-5

---------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

ThreadGroup tg = new ThreadGroup("STUDY ONLINE");

Runnable r = new Thread();

Thread t= new Thread(tg,r);

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[Thread-1,5,STUDY ONLINE]

Thread-1 is name of the thread

5 is thread priority

STUDY ONLINE is thread group name

\*/

// EXAMPLE-6

-------------

class MyThread //extends Thread // implements Runnable

{

public MyThread()

{

}

public static void main(String[] args)

{

//MyThread mt = new MyThread();

ThreadGroup tg = new ThreadGroup("STUDY ONLINE");

Runnable r = new Thread();

Thread t= new Thread(tg,r,"java\_course");

System.out.println(t);

}

}

/\*

OUTPUT

----------

Thread[java\_course,5,STUDY ONLINE]

java\_course is name of the thread

5 is thread priority

STUDY ONLINE is thread group name

\*/

IN-BUILT METHODS AVAILABLE IN THREAD CLASS:

--------------------------------------------

1. public void setName(String name)

--> ()It can be used to set a particular name to the Thread explicitly.

2. public String getName()

--> It can be used to get Thread name explicitly

3. public void setPriority(int priority)

--> It can be used to set a particular value to the thread but here the priority value must be provided within 1 to 10.

apart form this any number is given as prioorty then jvm will give us "illegalargumentException".

NOTE: min priority is : 1

avg priority is : 5

max prioroty is : 10

4. public int getpriority()

--> It can be used to get the prioroty value of the thread

EXAMPLE:

---------

class Demo

{

public static void main(String[] args)

{

Demo d = new Demo();

Thread t = new Thread();

System.out.println(t.getPriority());

t.setPriority(7);

System.out.println(t.getPriority());

t.setPriority(Thread.MAX\_PRIORITY-2);

System.out.println(t.getPriority());

t.setPriority(12);

}

}

/\*

OUTPUT:

---------

5

7

8

Exception in thread "main" java.lang.IllegalArgumentException

at java.lang.Thread.setPriority(Thread.java:1089)

at Demo.main(Demo.java:12)

5. public static int activeCount()

--> It will return the number of threads which are active/running/executing.

EXAMPLE:

-----------

class Demo1

{

public static void main(String[] args)

{

Thread t1 = new Thread();

Thread t2 = new Thread();

t1.start();

t2.start();

System.out.println(Thread.activeCount());

}

}

OUTPUT:

-------

3

6. public boolean isAlive()

--> This method can be used to check whether a thread is alive or not

EXAMPLE:

---------

class Demo1

{

public static void main(String[] args)

{

Thread t1 = new Thread();

System.out.println(t1.isAlive()); --> false

t1.start();

System.out.println(t1.isAlive()); --> true

}

}

7. public static thread currentThread()

--> is can be used to get thread object ref which is in active at present

EXAMPLE:

---------

class MyThread extends Thread

{

public void run()

{

System.out.println(Thread.currentThread().getName());

}

}

class Demo1

{

public static void main(String[] args)

{

MyThread t1 = new MyThread();

MyThread t2 = new MyThread();

MyThread t3 = new MyThread();

t1.setName("t1");

t2.setName("t2");

t3.setName("t3");

t1.start();

t2.start();

t3.start();

}

}

OUPUT:

------

t2

t1

t3

THREAD PRIORITY:

----------------

EXAMPLE:

---------

class ChildThread extends Thread

{

public void run()

{

for (int i =1 ;i<=10 ;i++ )

{

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

}

}

}

class Demo1

{

public static void main(String[] args)

{

ChildThread ct = new ChildThread();

ct.start();

ct.setPriority(10);

for (int i =1 ;i<=10 ;i++ )

{

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

}

System.out.println(ct.getPriority());

}

}

/\*

OUTPUT:

--------

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Child Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

Main Thread

10

NOTE: The default thread is main thread and its prioroty is 5

the user deifiend thread is originated from main thread hence its default prioroty will also be 5

always parent priority is given to its child threads

we can change the child prioirty by using setPriority method.

// case-1

------------

class childThread extends Thread

{

}

class Demo2

{

public static void main(String[] args)

{

Thread t = Thread.currentThread(); // w.r.t main thread

childThread ct = new childThread();

t.setPriority(10); // main thread

System.out.println(t.getPriority()); // 10

System.out.println(ct.getPriority()); // 5

}

}

// case-2

-----------

class childThread extends Thread

{

}

class Demo2

{

public static void main(String[] args)

{

Thread t = Thread.currentThread(); // w.r.t main thread

t.setPriority(10); // main thread

childThread ct = new childThread();

System.out.println(t.getPriority()); // 10

System.out.println(ct.getPriority()); // 10

}

}