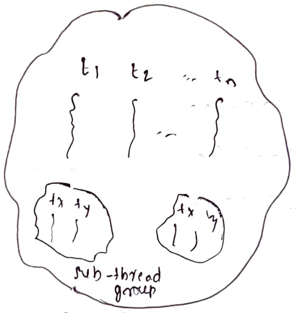
Thread group

Based on functionality we can group threads into a single unit, which is nothing but thread group

i.e thread group contains a group of threads

. in addition to threads, thread group also contains sub-thread groups



Thread group

· the main advantage of maintaining threads in the form of thread group is. we can perform common operations very easily

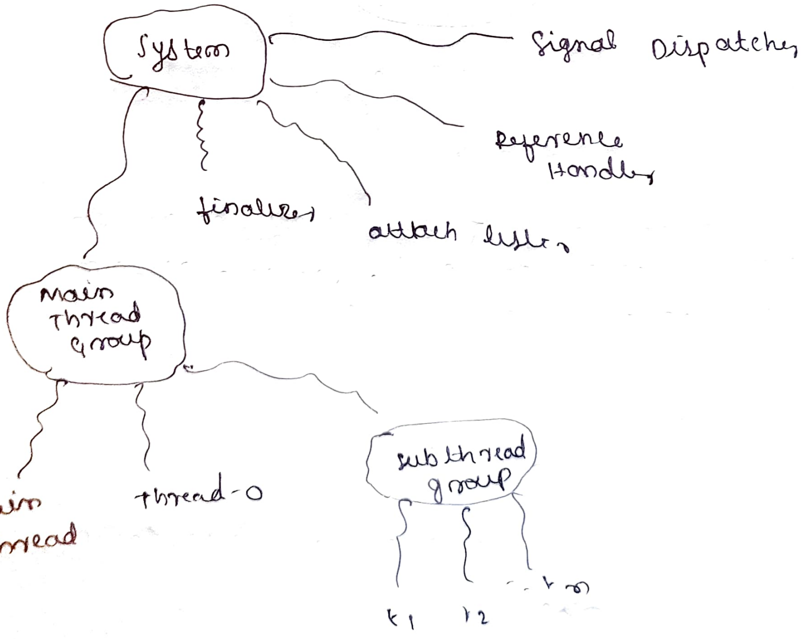
```
Class Test {
  psvm(String[] args) {
    Sop(Thread.currentThread().getThreadGroup().getName());
    //main - group
  }
}
    Sop(Thread.currentThread().getThreadGroup().
                                   getParent().
                                   getName());
```

main thread   main Thread group   System thread group   system

- Every thread in java belongs to some group
- main Thread belongs to main group
- Every thread group in java is the child group of System group, either directly or indirectly
- hence System group acts as root for all thread groups in java

- System group contains several S/m level threads like.

  finalizer
  Reference Handler
  Signal Dispatcher
  Attach Listener



· thread group is a java class present in java.lang package and it is a direct child class of object

· Constructors

① threadGroup g = new Thread Group (String group name);

· creates a new threadgroup with the specified groupName

· The parent of this new thread group is the threadgroup of currently executing thread

Ex: threadGroup g = new thread Group ("first Group");

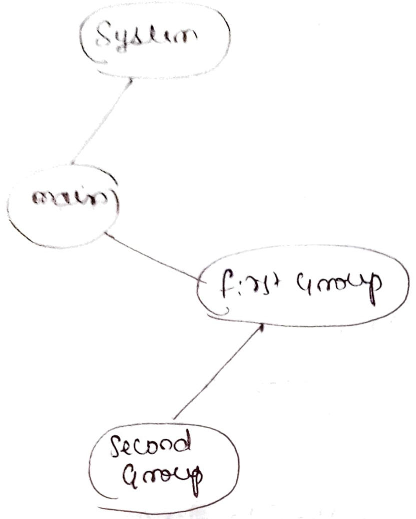
② ThreadGroup g = new ThreadGroup(thread Group PG, String Group");

· creates a new thread group with a specified group name

· the parent of this new thread group is specified parent group

ex. threadGroup g1 = new ThreadGroup(g, "second Group");

```
Class Test{
  psvm( String[] args) {
    Thread Group g1 = new thread Group("first Group");
  sop( g1.getParent() . getName()); //main
  Thread Group g2 = new ThreadGroup (g1, "second
                                        Group");
  sop( g2.getParent() . getName()); ] // first Group
} }
```



important methods of ThreadGroup class

1. **String getName()**

   returns name of the threadGroup

2. **int getMaxPriority()**

   returns max priority of threadGroup

3. **void setMaxPriority(int p)**

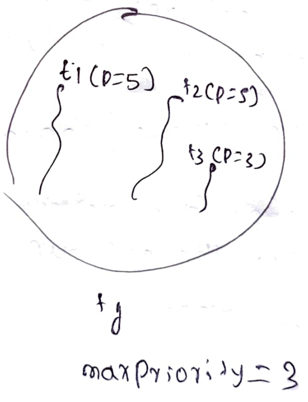
   to set maximum priority of threadGroup

- the default maxpriority is 10

Threads in the threadGroup that ~~have~~ have already higher priority won't be effect but, for newly added thread this maxpriority is applicable

```
class ThreadGroupDemo2 {
  psvm( String[] args) {
    ThreadGroup g1 = new ThreadGroup("tg");
    Thread t1 = new Thread (g1, "Thread1");    // 5
    Thread t2 = new Thread (g1, "thread2");  // 5
    g1.setmaxpriority(3);
    Thread t3 = new Thread(g1, "Thread3");  // 3
    sop( t1.getPriority());
    sop( t2.getpriority());
    sop( t3.getpriority());
  }
}
```

o/p :
```
5
5
3
```



4. ThreadGroup getParent()

returns ParentGroup of current thread

5. void list()

it prints information about threadgroup to the console

6. int activeCount()

returns number of active threads present in the threadGroup

7. int ~~[illegible]~~ activeGroupCount()

it returns number of active groups present in the current threadGroup

8. int enumurate(Thread[] t)

to copy all active threads of this threadGroup into provided thread array

In this case sub thread group also thread will be considerd

9. int enumerate(ThreadGroup[] g)

to copy all active sub thread groups into threadGroup array

10 boolean is Daemon ()

to check wether the threadgroup is daemon or not

11. void setDaemon (boolean b)

12. void interrupt()

to interrupt all waiting or sleeping threads present in the threadgroup

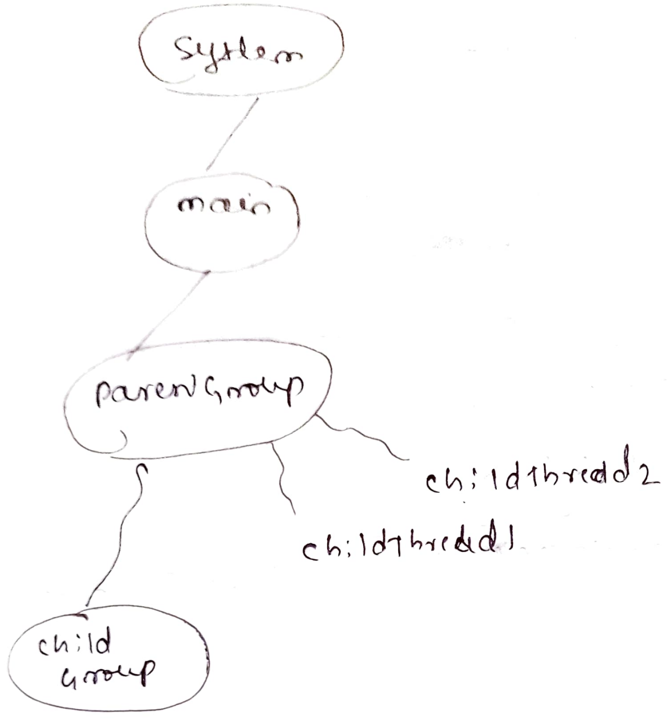
13. void destroy ()

to destroy threadgroup and it's sub threadGroups

```
class MyThread extends Thread {
  MyThread ( threadGroup g, String name ) {
        super( g, name );
     }
      public void run() {
      sop("child Thread");
      try {
        thread.sleep( 5000 );
      }
      catch ( InterruptedException e) { }
    }
  }

class   ThreadGroupDemo3 {
  psvm (String[] args) { throws Exception
   ThreadGroup  pg = new ThreadGroup("parentGroup");
   Thread Group  cg = new ThreadGroup (pg, "childGroup"
    Mythread  t1 = new mythread ( pg, "child thread1");
  Mythread t2 = new Mythread (pg, "child thread2");
   t1.start();
   t2.start();

   sop( pg.activeCount());   //2
   sop( pg. activeGroupCount());  // 1
   pg. list();
   Thread. sleep(10000);
   sop( pg. activeCount());
   sop( pg. activeGroupCount());
   pg. list();
 }
}
```



• Write a program to display all active thread names belongs to System group and its childgroups.

```
class ThreadGroupDaemon4 {
    psvm (String[] args){
    ThreadGroup system = Thread.currentThread().
                    getThreadGroup().getParent();
   Thread[]t = new Thread[system.activeCount()];
   system.enumerate(t) ;
    for( Thread t1 : t) {
          sop( t1.getName() +"... "+ t1.isDaemon());
     }
   }
}
```

```
O/P :  Reference Handler   true
       Finalizer         true
       Signal Dispatcher   true
       Attach Listener     true
       ~~main ...~~        false
       Notification Thread ..  true
       main
                           false
       Common - Cleaner      true;
```