

스레드 그룹

스레드 그룹

❖ 스레드 그룹

- 관련된 스레드 묶어 관리 목적으로 이용
- 스레드 그룹은 계층적으로 하위 스레드 그룹 가질 수 있음
- 자동 생성되는 스레드 그룹
 - system 그룹: JVM 운영에 필요한 스레드들 포함
 - system/main 그룹: 메인 스레드 포함
- 스레드는 반드시 하나의 스레드 그룹에 포함
 - 기본적으로 자신을 생성한 스레드와 같은 스레드 그룹
 - 스레드 그룹에 포함시키지 않으면 기본적으로 system/main 그룹

스레드 그룹

❖ 스레드 그룹 이름 얻기

```
ThreadGroup group = Thread.currentThread().getThreadGroup();  
String groupName = group.getName();
```

스레드 그룹

❖ 현재 실행 중인 스레드 정보 : ThreadInfoExample.java

```
import java.util.Map;
import java.util.Set;

public class ThreadInfoExample {
    public static void main(String[] args) {
        AutoSaveThread autoSaveThread = new AutoSaveThread();
        autoSaveThread.setName("AutoSaveThread");
        autoSaveThread.setDaemon(true);
        autoSaveThread.start();

        Map<Thread, StackTraceElement[]> map = Thread.getAllStackTraces();
        Set<Thread> threads = map.keySet();
        for (Thread thread : threads) {
            System.out.println("Name: " + thread.getName() +
                               ((thread.isDaemon()) ? "(데몬)" : "(주)"));
            System.out.println("\t" + "소속그룹: " +
                               thread.getThreadGroup().getName());
            System.out.println();
        }
    }
}
```

스레드 그룹

❖ 스레드 그룹 생성

```
ThreadGroup tg = new ThreadGroup(String name);  
ThreadGroup tg = new ThreadGroup(ThreadGroup parent, String name);
```

- 부모(parent) 그룹 지정하지 않으면?
 - 현재 스레드 속한 그룹의 하위 그룹으로 생성

```
Thread t = new Thread(ThreadGroup group, Runnable target);
```

```
Thread t = new Thread(ThreadGroup group, Runnable target, String name);
```

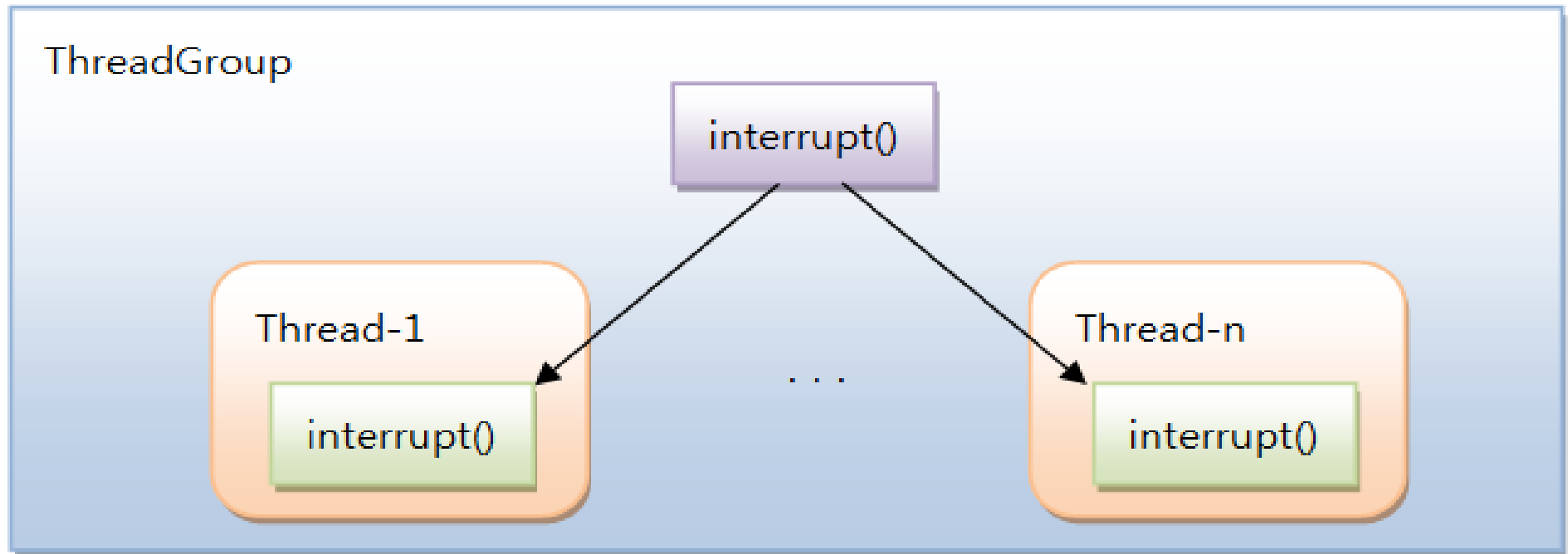
```
Thread t = new Thread(ThreadGroup group, Runnable target, String name, long stackSize);
```

```
Thread t = new Thread(ThreadGroup group, String name);
```

스레드 그룹

❖ 스레드 그룹의 일괄 interrupt()

- 스레드 그룹의 interrupt() 호출 시 소속된 모든 스레드의 interrupt() 호출



스레드 그룹

❖ 스레드 그룹 관련 메서드

메소드		설명
int	activeCount()	현재 그룹 및 하위 그룹에서 활동 중인 모든 스레드의 수를 리턴한다.
int	activeGroupCount()	현재 그룹에서 활동 중인 모든 하위 그룹의 수를 리턴한다.
void	checkAccess()	현재 스레드가 스레드 그룹을 변경할 권한이 있는지 체크한다. 만약 권한이 없으면 SecurityException을 발생시킨다.
void	destroy()	현재 그룹 및 하위 그룹을 모두 삭제한다. 단, 그룹 내에 포함된 모든 스레드들이 종료 상태가 되어야 한다.
boolean	isDestroyed()	현재 그룹이 삭제되었는지 여부를 리턴한다.
int	getMaxPriority()	현재 그룹에 포함된 스레드가 가질 수 있는 최대 우선순위를 리턴한다.

스레드 그룹

❖ 스레드 그룹 관련 메서드

메소드		설명
void	setMaxPriority(int pri)	현재 그룹에 포함된 스레드가 가질 수 있는 최대 우선순위를 설정한다.
String	getName()	현재 그룹의 이름을 리턴한다.
ThreadGroup	getParent()	현재 그룹의 부모 그룹을 리턴한다.
boolean	parentOf(ThreadGroup g)	현재 그룹이 매개값으로 지정한 스레드 그룹의 부모인지 여부를 리턴한다.
boolean	isDaemon()	현재 그룹이 데몬 그룹인지 여부를 리턴한다.
void	setDaemon(boolean daemon)	현재 그룹을 데몬 그룹으로 설정한다.
void	list()	현재 그룹에 포함된 스레드와 하위 그룹에 대한 정보를 출력한다.
void	interrupt()	현재 그룹에 포함된 모든 스레드들을 interrupt한다.

스레드 그룹

❖ InterruptedException이 발생할 때 스레드가 종료되도록 함: WorkThread.java

```
public class WorkThread extends Thread {
    public WorkThread(ThreadGroup threadGroup, String threadName) {
        super(threadGroup, threadName); // 스레드 그룹과 스레드 이름을 설정
    }

    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                // InterruptedException 발생시 스레드 종료
                System.out.println(getName() + " interrupted");
                break;
            }
        }
        System.out.println(getName() + " 종료됨");
    }
}
```

스레드 그룹

❖ 스레드 그룹을 이용한 일괄 종료 예제 : ThreadGroupExample.java

```
public class ThreadGroupExample {  
    public static void main(String[] args) {  
        ThreadGroup myGroup = new ThreadGroup("myGroup");  
        WorkThread workThreadA = new WorkThread(myGroup, "workThreadA");  
        WorkThread workThreadB = new WorkThread(myGroup, "workThreadB");  
  
        workThreadA.start();  
        workThreadB.start();  
  
        System.out.println("[ main 스레드 그룹의 list() 메소드 출력 내용 ]");  
        ThreadGroup mainGroup = Thread.currentThread().getThreadGroup();  
        mainGroup.list();  
        System.out.println();  
  
        try {  
            Thread.sleep(3000);  
        } catch (InterruptedException e) {  
        }  
  
        System.out.println("[ myGroup 스레드 그룹의 interrupt() 메소드 호출 ]");  
        myGroup.interrupt();  
    }  
}
```