

Table [クラスThreadのフィールド]

フィールド	説明
static final int MAX_PRIORITY	スレッドに設定できる最高優先順位
static final int MIN_PRIORITY	スレッドに設定できる最低優先順位
static final int NORM_PRIORITY	スレッドに設定されるデフォルトの優先順位

Table [クラスThreadのメソッド]

	メソッド	説明	
コンストラクタ	Thread() Thread(String name)	スレッドオブジェクトを生成する。nameが指定されたときには、nameという名前を持つスレッドオブジェクトを生成する。	
	Thread(Runnable target) Thread(Runnable target, String name)	Runnable型オブジェクトをラップするスレッドオブジェクトを生成する。スレッドの名前をnameで指定することもできる	
	Thread(ThreadGroup group, Runnable target) Thread(ThreadGroup group, Runnable target, String name) Thread(ThreadGroup group, Runnable target, String name, long stackSize) Thread(ThreadGroup group, String name)	スレッドグループgroupに属するスレッドオブジェクトを生成する。targetが指定された場合は、targetをラップする。nameが指定された場合は、名前nameを持つスレッドオブジェクトを生成する。stackSizeを指定された場合は、生成されたスレッドのスタックサイズはstackSizeとなる。	
	基本	void start()	このスレッドのrun()メソッドを呼び出し、このスレッドの実行を開始する。すでに実行中なら、IllegalThreadStateException例外を投げる。
		void run()	このスレッドのstart()メソッドがこのメソッドを呼び出すことで、スレッドの実行が始まる。Threadのサブクラスは、このメソッドをオーバーライドすることで、スレッドの処理内容を記述する。publicであることに注意。
		static Thread currentThread()	現在実行中のスレッドを返す
属性	String getName()	このスレッドの名前を返す	
	void setName(String name)	このスレッドの名前をnameに設定。設定が許されないならSecurityException例外を投げる	
	String toString()	このスレッドの名前、優先順位、スレッドグループ名を含む文字列を返す	
	boolean isDaemon()	このスレッドがデーモンスレッドの場合にtrueを返す	
	void setDaemon(boolean on)	onがtrueならこのスレッドをデーモンスレッドとし、falseならこのスレッドをユーザスレッドとする。スレッド実行開始前に呼び出さなければならず、実行開始後に呼ぶとIllegalThreadStateException例外が投げられる。このスレッドの変更が許されない場合、SecurityException例外を投げる。	
同期	static boolean holdsLock(Object obj)	現在実行中のスレッドがobjに対するロックを持っている場合にtrueを返す	
実行制御とスケジューリング	boolean isAlive()	このスレッドが生存している場合にtrueを返す	
	void join() void join(long millis) void join(long millis, int nanos)	このスレッドの終了まで最高millisミリ秒+nanosナノ秒待つ。時間が無指定か0なら待機し続ける。他スレッドに割り込まれるとInterruptedException例外が投げられ、本スレッドの割り込みステータスはクリアされる。	
	void destroy()	後始末(クリーンアップ)せずに、このスレッドを終了する(未実装のまま)。	
	static void sleep(long millis) static void sleep(long millis, int nanos)	現在実行中のスレッドをmillisミリ秒(+nanosナノ秒)だけ一時停止する。他のスレッドに割り込まれるとInterruptedException例外が投げられ、このスレッドの割り込みステータスはクリアされる。	
	void interrupt()	このスレッドに対して割り込みをかける。割り込みが許されない場合は、SecurityException例外が投げられる。	
	static boolean interrupted()	現在実行中のスレッドが割り込まれている場合はtrueを返す。現在実行中のスレッドの割り込みステータスはクリアされる。	
	boolean isInterrupted()	このスレッドが割り込まれている場合はtrueを返す。このスレッドの割り込みステータスはクリア「されない」。	
	static void yield()	現在実行中のスレッドを停止し、他のスレッドに実行の機会を与える	
	int getPriority()	このスレッドの優先順位を返す	
	void setPriority(int newPriority)	このスレッドの優先順位を、newPriorityとこのスレッドの属するスレッドグループの最高許容優先順位の低い方に設定する	
グループ	ThreadGroup getThreadGroup()	このスレッドの属するスレッドグループを返す	
	static int activeCount()	現在実行中のスレッドの属するスレッドグループで、いくつかのスレッドがアクティブかを返す。	
その他	ClassLoader getContextClassLoader()	このスレッドのコンテキストClassLoaderを返す。許可されない場合は、SecurityException例外を投げる。	
	void setContextClassLoader(ClassLoader cl)	このスレッドのコンテキストClassLoaderを設定する。許可されない場合は、SecurityException例外を投げる。	
	void checkAccess()	現在実行中のスレッドがこのスレッドを変更するアクセス権を持っているかどうかを検査する。アクセス権限を持っていない場合は、SecurityException例外を投げる	
	static int enumerate(Thread[] tarray)	現在実行中のスレッドのスレッドグループのアクティブなスレッドを、配列targetにコピー。許されない場合、SecurityException例外を投げる。	
	static void dumpStack()	現在実行中のスレッドのスタック情報を出力する	