Module java.base
Package java.lang.ref

Class Cleaner

java.lang.Object
    java.lang.ref.Cleaner

```
public final class Cleaner
extends Object
```

`Cleaner` manages a set of object references and corresponding cleaning actions.

Cleaning actions are `registered` to run after the cleaner is notified that the object has become phantom reachable. The cleaner uses `PhantomReference` and `ReferenceQueue` to be notified when the reachability changes.

Each cleaner operates independently, managing the pending cleaning actions and handling threading and termination when the cleaner is no longer in use. Registering an object reference and corresponding cleaning action returns a `Cleanable`. The most efficient use is to explicitly invoke the `clean` method when the object is closed or no longer needed. The cleaning action is a `Runnable` to be invoked at most once when the object has become phantom reachable unless it has already been explicitly cleaned. Note that the cleaning action must not refer to the object being registered. If so, the object will not become phantom reachable and the cleaning action will not be invoked automatically.

The execution of the cleaning action is performed by a thread associated with the cleaner. All exceptions thrown by the cleaning action are ignored. The cleaner and other cleaning actions are not affected by exceptions in a cleaning action. The thread runs until all registered cleaning actions have completed and the cleaner itself is reclaimed by the garbage collector.

The behavior of cleaners during `System.exit` is implementation specific. No guarantees are made relating to whether cleaning actions are invoked or not.

Unless otherwise noted, passing a `null` argument to a constructor or method in this class will cause a `NullPointerException` to be thrown.

API Note:

The cleaning action is invoked only after the associated object becomes phantom reachable, so it is important that the object implementing the cleaning action does not hold references to the object. In this example, a static class encapsulates the cleaning state and action. An "inner" class, anonymous or not, must not be used because it implicitly contains a reference to the outer instance, preventing it from becoming phantom reachable. The choice of a new cleaner or sharing an existing cleaner is determined by the use case.

If the CleaningExample is used in a try-finally block then the `close` method calls the cleaning action. If the `close` method is not called, the cleaning action is called by the Cleaner when the CleaningExample instance has become phantom reachable.

```
public class CleaningExample implements AutoCloseable {
       // A cleaner, preferably one shared within a library
       private static final Cleaner cleaner = <cleaner>;

       static class State implements Runnable {

           State(...) {
               // initialize State needed for cleaning action
           }

           public void run() {
               // cleanup action accessing State, executed at most once
           }
       }

       private final State state;
       private final Cleaner.Cleanable cleanable;

       public CleaningExample() {
           this.state = new State(...);
           this.cleanable = cleaner.register(this, state);
       }

       public void close() {
           cleanable.clean();
       }
   }
```

The cleaning action could be a lambda but all too easily will capture the object reference, by referring to fields of the object being cleaned, preventing the object from becoming phantom reachable. Using a static nested class, as above, will avoid accidentally retaining the object reference.

Cleaning actions should be prepared to be invoked concurrently with other cleaning actions. Typically the cleaning actions should be very quick to execute and not block. If the cleaning action blocks, it may delay processing other cleaning actions registered to the same cleaner. All cleaning actions registered to a cleaner should be mutually compatible.

Since:

9

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
static interface	Cleaner.Cleanable	Cleanable represents an object and a cleaning action registered in a Cleaner.

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
static Cleaner	create()	Returns a new Cleaner.
static Cleaner	create(ThreadFactory threadFactory)	Returns a new Cleaner using a Thread from the ThreadFactory.
Cleaner.Cleanable	register(Object obj, Runnable action)	Registers an object and a cleaning action to run when the object becomes phantom reachable.

Methods declared in class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Details

create

```
public static Cleaner create()
```

Returns a new Cleaner.

The cleaner creates a daemon thread to process the phantom reachable objects and to invoke cleaning actions. The context class loader of the thread is set to the system class loader. The thread has no permissions, enforced only if a SecurityManager is set.

The cleaner terminates when it is phantom reachable and all of the registered cleaning actions are complete.

Returns:

a new Cleaner

Throws:

SecurityException - if the current thread is not allowed to create or start the thread.

create

```
public static Cleaner create(ThreadFactory threadFactory)
```

Returns a new Cleaner using a Thread from the ThreadFactory.

A thread from the thread factory's newThread method is set to be a daemon thread and started to process phantom reachable objects and invoke cleaning actions. On each call the thread factory must provide a Thread that is suitable for performing the cleaning actions.

The cleaner terminates when it is phantom reachable and all of the registered cleaning actions are complete.

Parameters:

threadFactory - a ThreadFactory to return a new Thread to process cleaning actions

Returns:

a new Cleaner

Throws:

IllegalThreadStateException - if the thread from the thread factory was not a new thread.

SecurityException - if the current thread is not allowed to create or start the thread.

register

```
public Cleaner.Cleanable register(Object obj,
                                  Runnable action)
```

Registers an object and a cleaning action to run when the object becomes phantom reachable. Refer to the API Note above for cautions about the behavior of cleaning actions.

Parameters:

`obj` - the object to monitor

`action` - a `Runnable` to invoke when the object becomes phantom reachable

Returns:

a `Cleanable` instance
