Ignify Consulting

Garbage Collection

Author: Douglas Dubier

douglas@ignify.com

Ignify Consulting
13304 Alondra Blvd #201.
Cerritos CA 90703
www.ignify.com

March 2003

Using Resources

Allocate memory for resource.
Set the initial state of the resource.
Use the resource.
Clean up the resource.
Free the memory.

What is Garbage Collection?

Automatic reclamation of unused objects.
Recycling of memory.
Decreases development time.
Prevents memory leaks.
Prevents dangling pointers.

Types of Garbage Collection

Reference Counting
Tracing

Reference Counting

Each object has a count of references to it.
Count increased when an additional reference is added. (B = A)
Count decreased when a reference is removed. (B = null or B = C)
Object is unused when count reaches 0.

Reference Counting

Advantages

- Simple to implement.
- Overhead is small and constant.
- Can be interleaved with normal program operation.

Reference Counting

Disadvantages

Cannot deal with cyclic structures:

$B \rightarrow A$, $A \rightarrow B$

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$

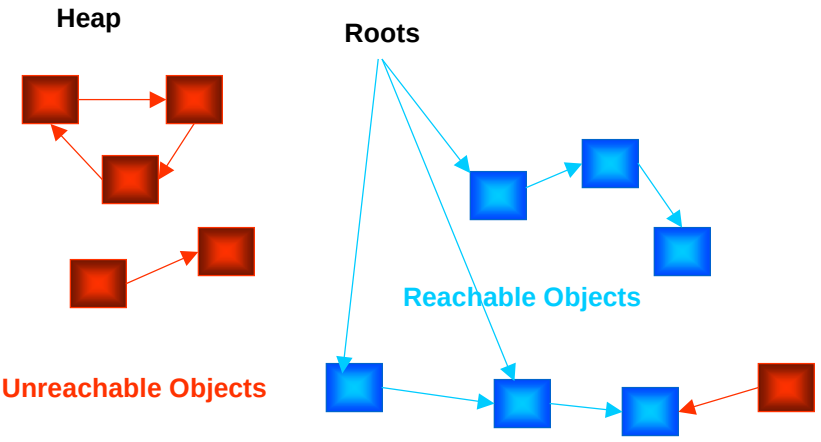
Parents & children with references to each other

Tracing

Roots: Global variables, local variables, instance members, class members, running threads.

Object is unused when unreachable from roots.

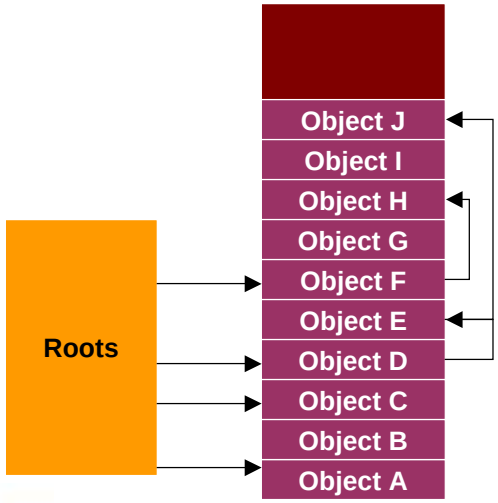
Tracing



Types of Tracing Collector

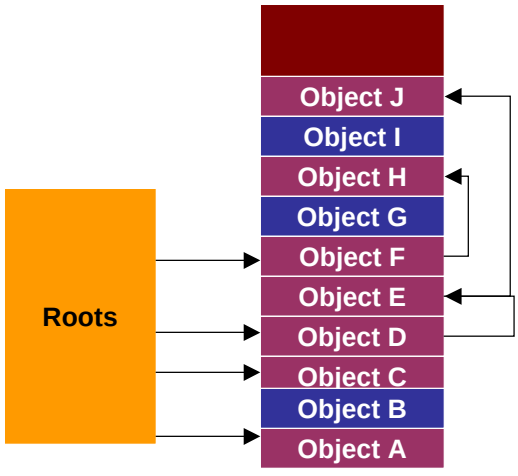
Mark-and-Sweep Collector
Copying Collector

Mark-and-Sweep Collector



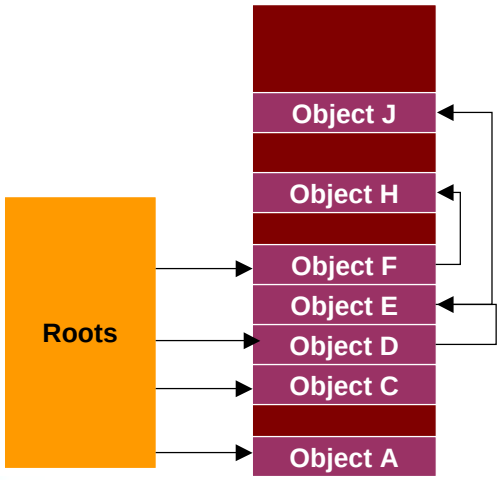
Ignify Consulting

Mark-and-Sweep Collector



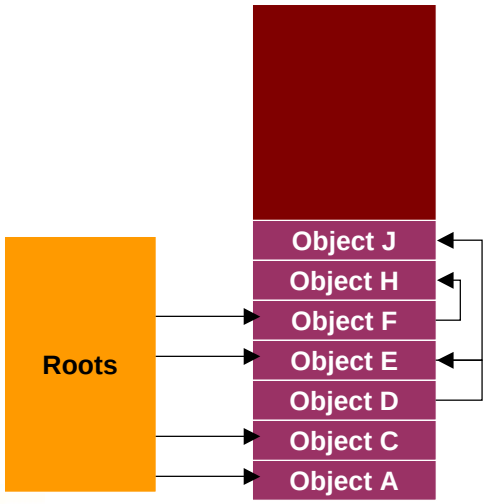
Ignify Consulting

Mark-and-Sweep Collector



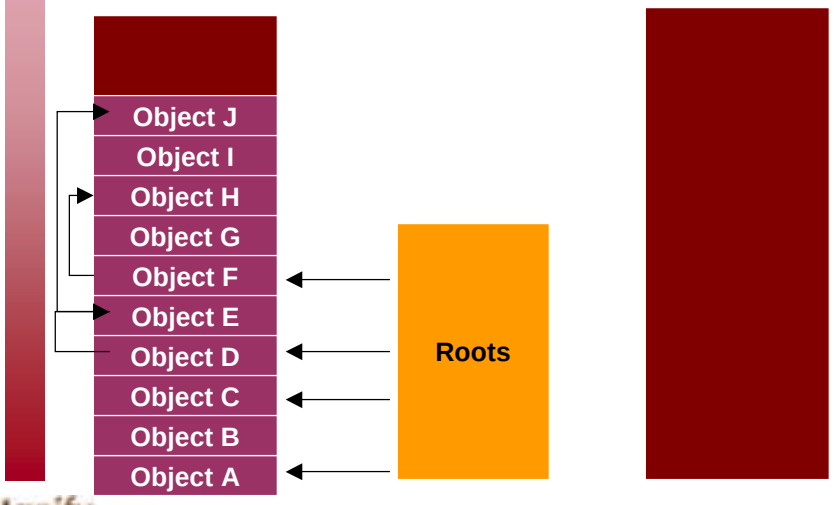
Ignify Consulting

Mark-and-Sweep Collector



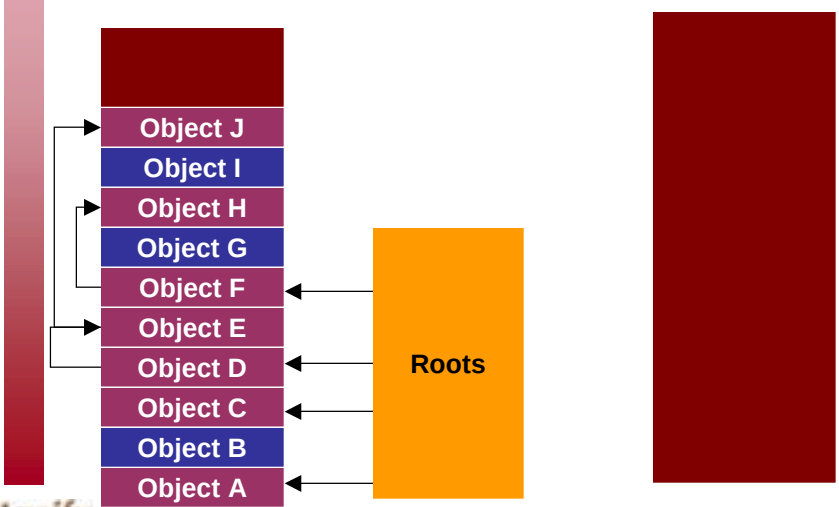
Ignify Consulting

Copying Collector



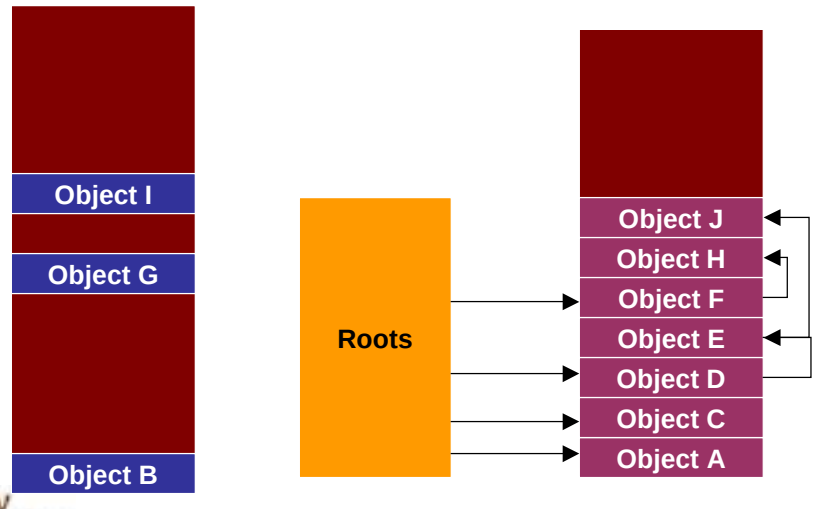
Ignify Consulting

Copying Collector



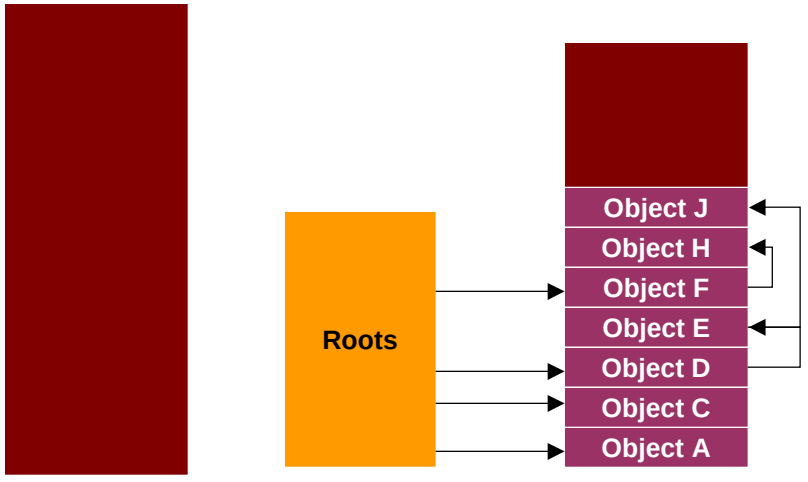
Ignify Consulting

Copying Collector



Ignify Consulting

Copying Collector



Ignify Consulting

Generational Collector

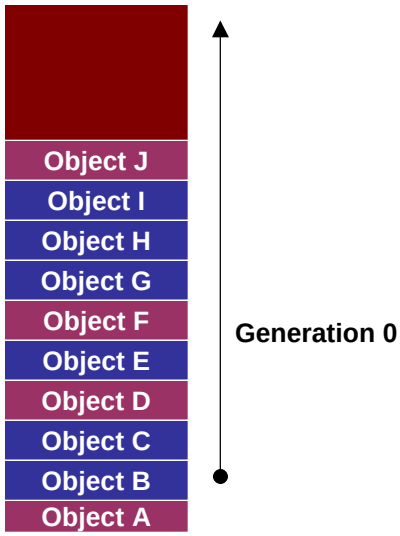
Most objects have very short lives.

Some objects have very long lives.

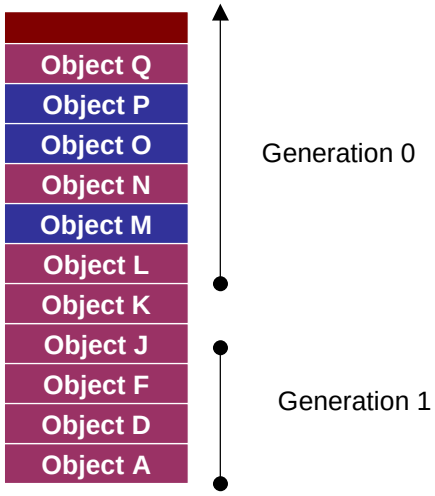
Younger objects frequently reference older objects; the reverse is rare.

Compacting a portion of the heap is faster than compacting the whole heap.

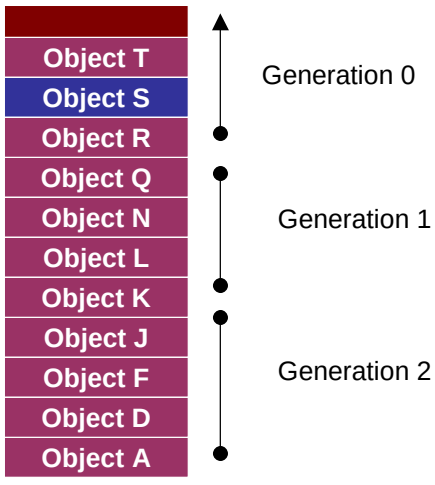
Generational Collector



Generational Collector



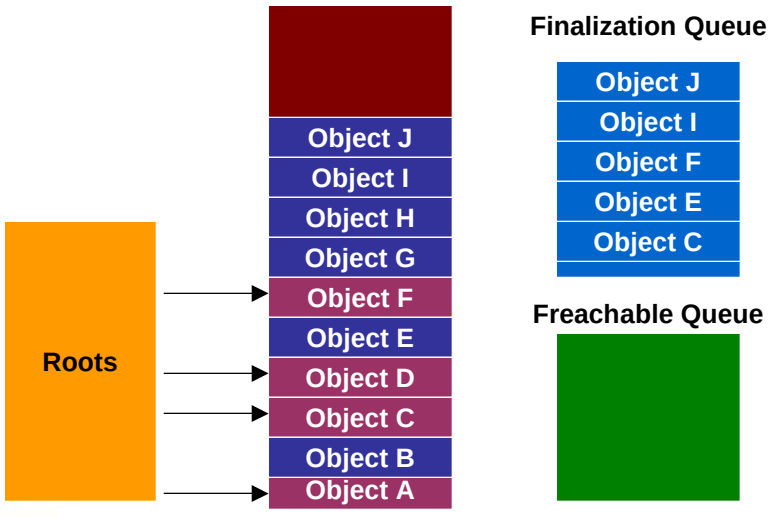
Generational Collector



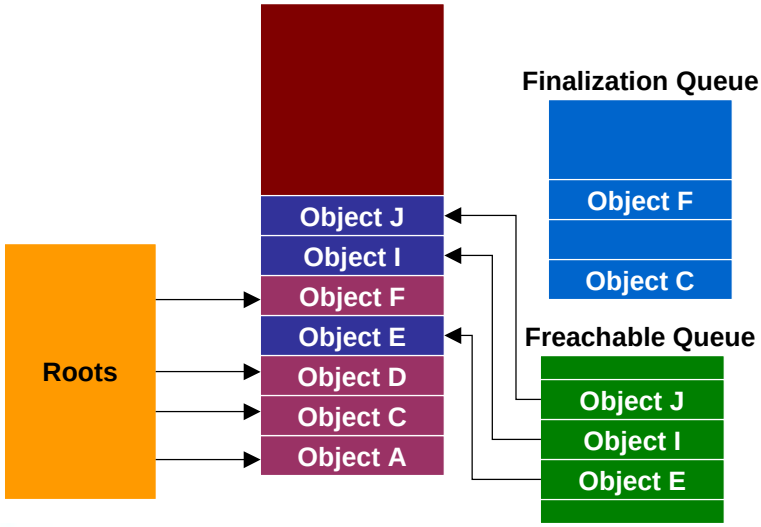
Finalization

Allows a resource to clean up when collected.
Class must implement finalize().
Not like a destructor.

Finalization



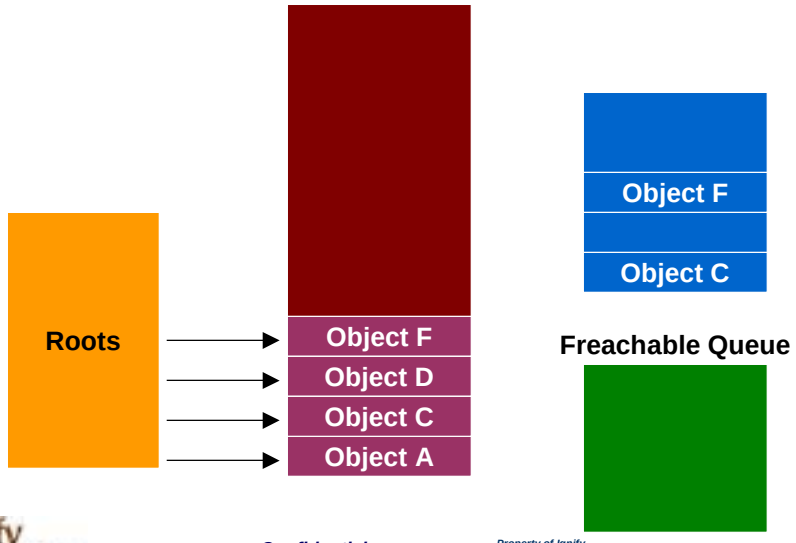
Finalization



Ignify Consulting

Confidential

Finalization



Ignify Consulting

Confidential

Finalization

Finalizable objects take longer to allocate.
Lifetime of object, and all dependant objects may be prolonged.
Increases memory pressure.
Errors during finalization are ignored.

Finalization

No control over when finalization happens.
Finalization may not happen.
Objects may be resurrected.

Weak References

Allow app to reference the object.
Allow garbage collector to collect the object.
Allow objects to be cached.
Destroys cached objects if memory is required.

Drawbacks of Garbage Collection

Significant performance hit.
Program execution halts during collection.
Little control over scheduling of collection.
No control over finalization.

Online Resources

The Garbage Collection List

http://www.iecc.com/gclist/

Memory Management in the .NET Framework

http://msdn.microsoft.com/msdnmag/issues/1100/GCI/GCI.asp

Online Resources

Reference Objects and Garbage Collection

http://developer.java.sun.com/developer/technicalArticles/ALT/RefObj/

Inside the Java 2 Virtual Machine - Garbage Collection

http://www.artima.com/insidejvm/ed2/ch09GarbageCollection01.html

Los Angeles Office

13304 Alondra Blvd #201, Cerritos CA 90703

Email: sales@Ignify.com

Tel: 562-404-8089

India Office

7 Madhuban, North Main Road, Koregaon Park, Pune

Tel: +91-20-612-0778

Email: India@Ignify.com

San Francisco Bay Area

4800 Great America Pkwy, Suite 310, Santa Clara, CA 95054

Tel: 408-480-3289

Global Website: http://www.Ignify.com

Ignify Consulting