Programación Orientada a Objetos con Cocoa

Manejo de Memoria

Amilcar Meneses Viveros

Sección Computación
Departamento de Ingeniería Eléctrica
CINVESTAV-IPN México, D.F.
(2003)

Introducción

- Parte de un *"Run Time System"* es el colector de basura.
- `C++` no tiene colector de basura. Los objetos son almacenados en pila.
- Java tiene colector de basura. Los objetos son almacenados en el *"heap"*.
- C++ no tiene colector de basura. Los objetos son almacenados en pila.
- Cocoa mantiene objetos por numero de referencia. Los objetos son almacenados en un área de memoria llamada *"Pool"*. Es responsabilidad del usuario eliminar los objetos que retiene.

Instanciando objetos

- Instanciación de objetos con los métodos `alloc`, `allocWithZone:`, `new`, `copy`, `copyWithZone:`, `multableCopy` o `mutablecopyWithZone:`.

```
id a, b;
Persona *l;
a = [[Persona alloc] initWithName:@''Pancho''];
l = [[Persona allocWithZone:[self zone]] init];
b = [a copy];
...
[a release];
[b autorelease];
[l release];
```

- Instanciar objetos con estos métodos incrementa el contador de referencias.

Instanciando objetos

- Es responsabilidad del programador liberar estos objetos para decrementar su contador de referencias. Esto se hace con los métodos `release` y `autorelease`.
- Cada objeto contiene su numero de referencias. Cuando el contador es cero, el objeto se elimina.
- Los métodos `release` o `autorelease` decrementan el contador de referencia. Esto es responsabilidad del usuario si instancializa objetos con los metodos del punto anterios.

Creando objetos transitorios

- Objetos que estén poco tiempo en memoria.
- Objetos que no necesiten que el usuario decremente su contador de referencia.
- Por simplicidad de codificación

```
- (void) hazAlgo: (NSArray *)a con: (int) i
{
  id unObjeto;
  NSColor *c = [NSColor blackColor];
  unObjeto = [a objectAtIndex:i];
  [unObjeto coloreate: c];
}
```

Reteniendo Objetos

- Los objetos que se instancian con algún mensaje que no sea `alloc, new, copy`...
- Retener un objeto transitorio cuando se requiere que su tiempo de vida sea mayor

```
Persona *p;
[p tuNombreEs: @''Pancho Lopez''];

...

@implementation Persona
- (id)tuNombreEs: (NSString *)n
{
   nombre = [n retain];  // Porque n es transitorio
                         // Que pasa si nombre!=nil?
}
```

Regresando objetos

- Se puede violar el encapsulamiento cuando se regresa un objeto:

```
Persona *p;
NSString *s;
p = [[[Persona alloc] init] tuNombreEs:''Pancho Lopez''];
s = [persona nombre];
s = @''Juan Perez'';          // !!!!!Que paso????!!!!
```

- Se debe regresar la copia del objeto requerido

```
// Mala implantacion:
- (NSString *) nombre
{
   return nombre;
}
```

```
// Implantacion adecuada
- (NSString *)nombre
{
  return [NSString stringWithString:nombre];
}
```

Regresando objetos

- De forma general

```
- (atributo *) atributo
{
    id = c;
    c = [atributo copy];
    return [c autorelease];
}
```

- Se debe buscar la manera de regresar una copia del objeto (atributo) que solicitan, a través de una instanciación por algún constructor (por ejemplo `+stringWithString:`), creando e inicializando el objeto con el valor deseado, o copiando y dejando en cero el contador de referencias.

En breve: reglas para el manejo de memoria

- Si instancías un objeto con los métodos `alloc, new, copy,`..., o si envías a una instancia el mensaje `retain`, eventualmente, debes liberarlo (mensajes `release, autorelease`).
- Si se obtiene la instancia del objeto por algún otro método, entonces no debes liberarlo (si necesitas retenerlo, enviar mensaje `retain` y aplicar la regla 1).
- `autorelease` singnifica libera después... cualquier cosa que signifique después...

Bibliografía

- Apple Documentation; *"Memory Management"*;
  `http://developer.apple.com/documentation/Cocoa/Conceptual/MemoryMgmt/index.html`;
  Apple Inc.; Nov. 2002
- Manu Iyengar; *"Memory Management with Cocoa/WebObjects"*;
  `http://www.stepwise.com/Articles/Technical/MemoryManagement.html`;
  `stepwise.com`; 06- 2001.
- Michael Beam; *"Memory Management in Objective-C"*;
  `http://www.macdevcenter.com/pub/a/mac/2001/07/27/cocoa.html`;
  `www.macdevcenter.com`; 07-2002;