Sigma
Software

SIGMA UNIVERSITY
Software

sigma
Software

UNIVERSITY

ЛЕКЦІЯ 5

“Управління памяттю в Swift”

5

SIGMA SOFTWARE
UNIVERSITY

Управління памяттю в Swift

Становлення проблеми управління пам’яттю в програмуванні

Retain counter

4 Основних правила ручного управління памяттю

ARC

NSZombie

Управління памяттю в Swift

Obj-C vs Swift memory Management

Swift memory management principles

Strong Reference Cycles

Resolving Strong Reference Cycles

Capture List

Problem of memory management

“*memory is always a limited resource* “ (Objective C Memory Management Essentials)

“*is the programming discipline of managing the life cycles of objects and freeing them when they are no longer needed. Managing object memory is a matter of performance; if an application doesn’t free unneeded objects, its memory footprint grows and performance suffers.*” (Apple)

SIGMA SOFTWARE
UNIVERSITY

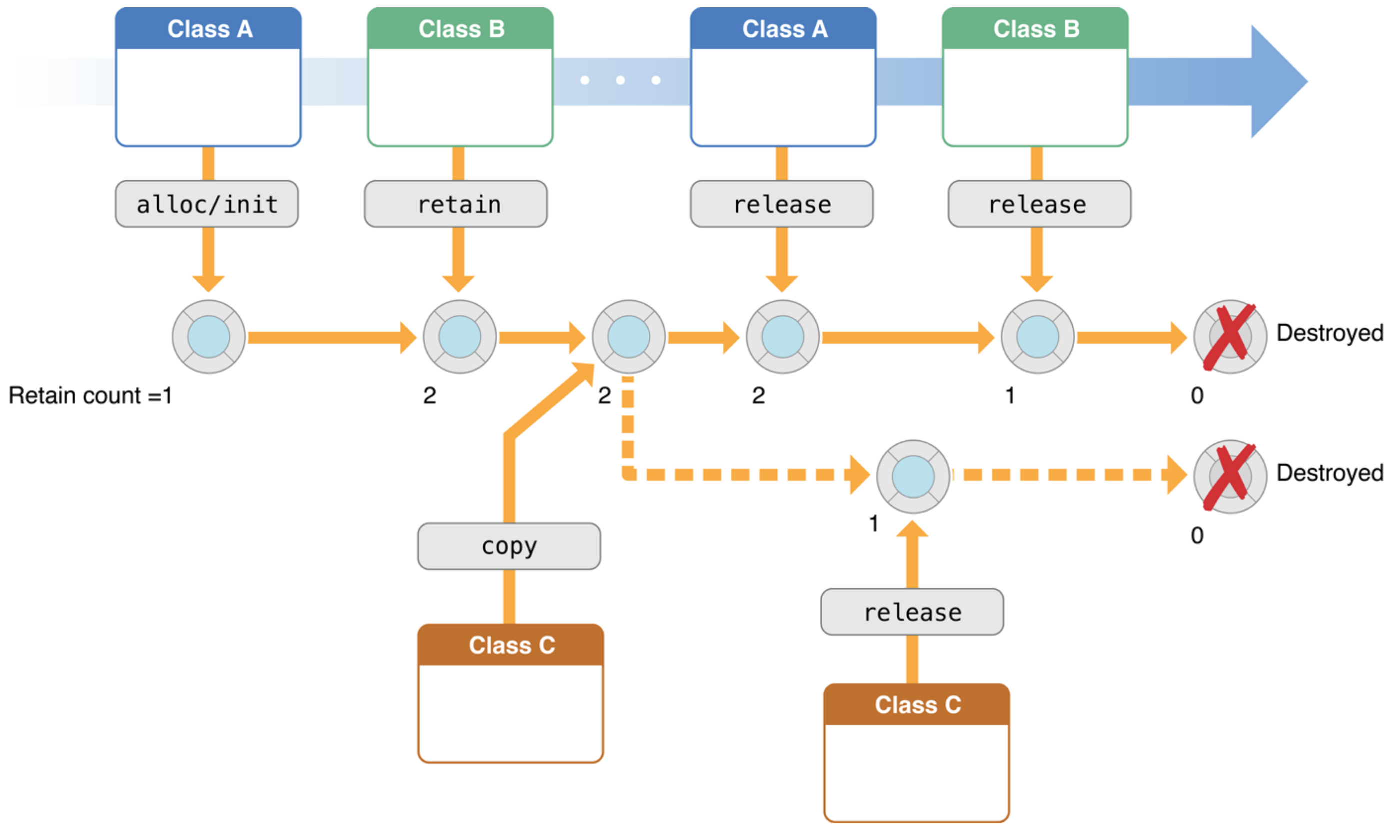
Memory Leak

“*is when your program loses track of a piece of memory that was allocated and has forgotten to release it.*“ (Objective C Memory Management Essentials)

asciiville by J.T.Presta
Copyright 2011, J.T.Presta. All rights reserved.
J.T.PRESTA 5/29/11
“Hey! Your application has a memory leak.”

SIGMA SOFTWARE UNIVERSITY

Retain counter



4 Основних правила ручного управління памяттю

If you own it, release it.

If you don’t own it, don’t release it.

Override dealloc in your classes to release the fields that you own.

Never call dealloc directly.

ARC

Automatic Reference Counting представлене в iOS5

Не треба використовувати retain/release

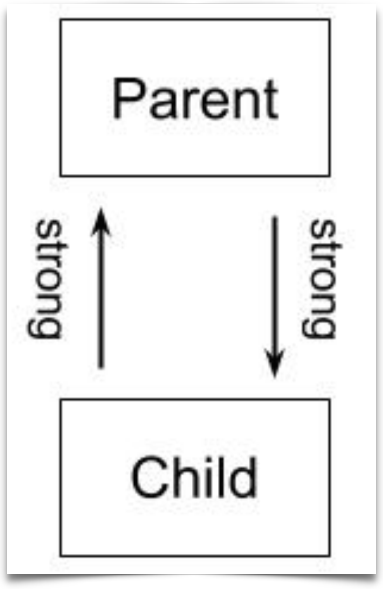
Компілятор додає retain/release в код автоматично

ARC звільняє об’єк як тільки всі strong посилання зникають

Це не Garbage Collector

Retain cycles

коли обєкти тримають посилання один на один



Retain cycles - Правила для уникання

Об’єкт ніколи не має тримати strong посилатися на «батьківський» обєкт

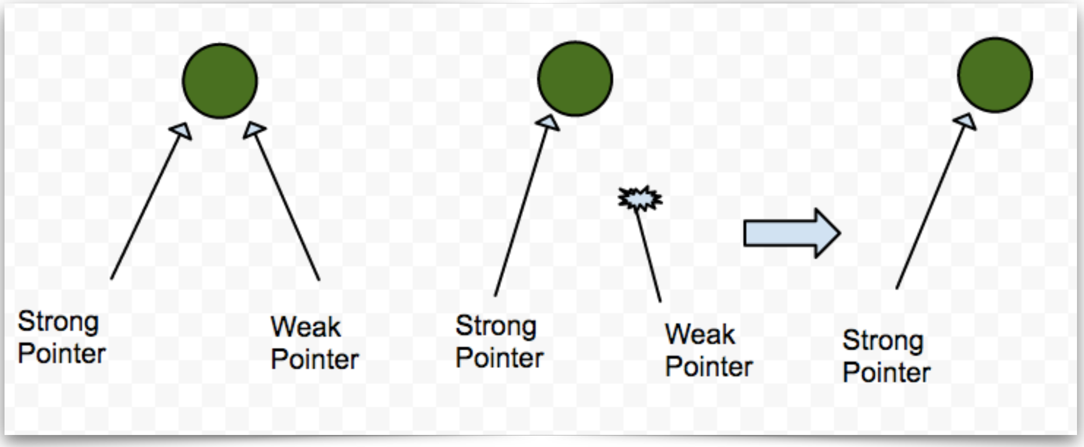
Object не має тримати strong посилання на будь-який об’єкт який стоїть вище в ієрархії
"Connection" об’єкти не повинні тримати strong посилання на їх цілі (delegate, outlets, observers)

SIGMA SOFTWARE UNIVERSITY

weak vs strong

strong - protects the referred object from getting deallocated by ARC

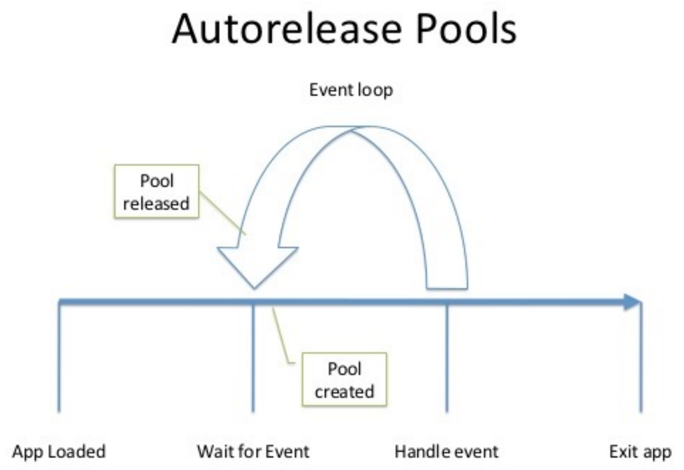
weak - don't protects the referred object from getting deallocated by ARC



“Use a **weak** reference whenever it is valid for that reference to become nil at some point during its lifetime. Conversely, use an unowned reference when you know that the reference will never be nil once it has been set during initialization.” (Apple)

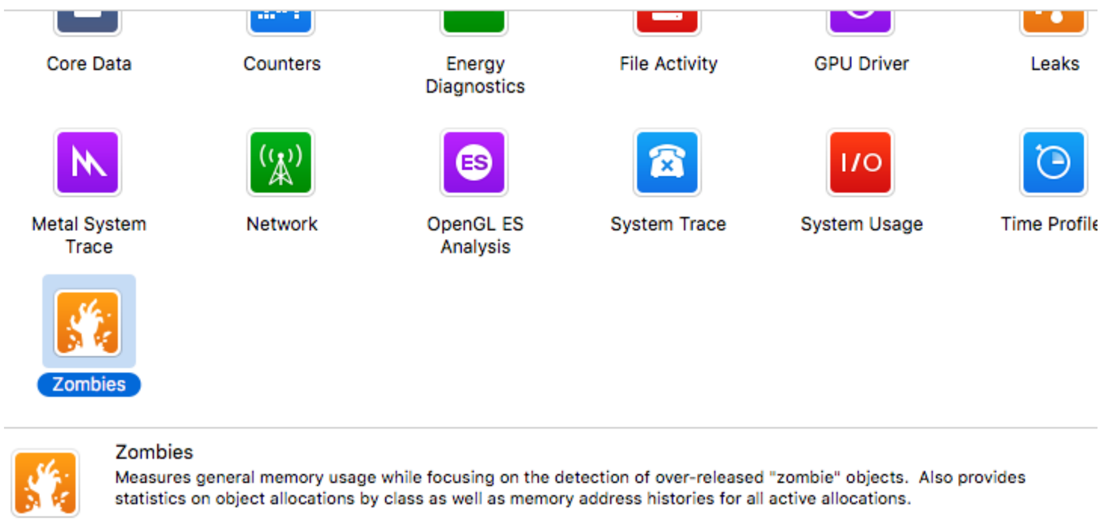
Autorelease POOL

“a mechanism whereby you can relinquish ownership of an object, but avoid the possibility of it being deallocated immediately”



NSZombie

"is a memory debugging aid which can help you debug subtle over-release/ autorelease problems."



Zombies
Measures general memory usage while focusing on the detection of over-released "zombie" objects. Also provides statistics on object allocations by class as well as memory address histories for all active allocations.

Obj-C vs Swift memory Management

is similar, but Swift use ARC

Swift is strongly typed and type safe, so all variables must have a known type and non-nil value (unless declared optional)

Swift memory management

“in Swift, memory management is made to be as painless as possible“

“this mean that memory management just work in Swift “ (Apple)

How ARC Work in Swift

Create object - ARC allocate chunk of memory with additional information about object

Deallocate object - ARC frees memory

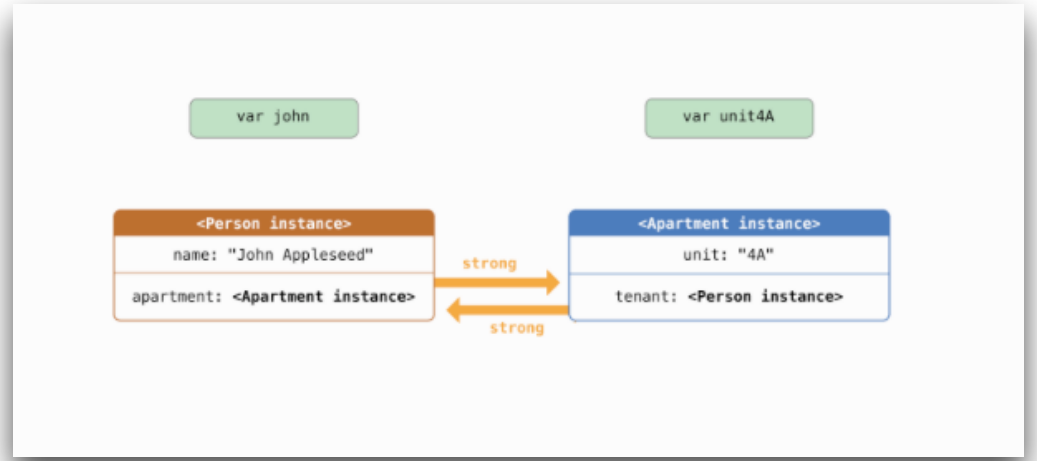
Tell what type of relationship between your classes

Access to deallocated obj crash app

Deallocated obj has referenceCount equal to 0

Strong reference cycle

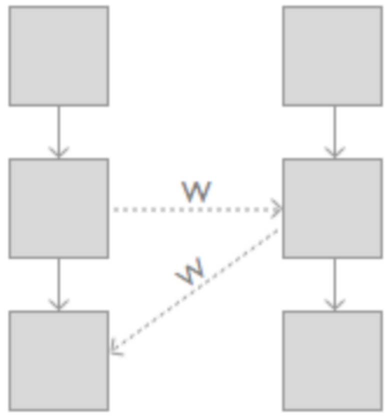
when instance of a class never get reference count equal to 0

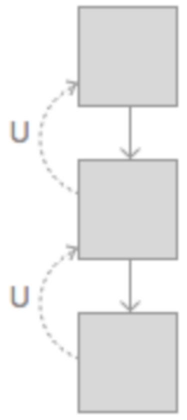


Resolving Strong reference cycle

weak reference

unowned reference





Resolving Strong reference cycle

*“Use a **weak** reference whenever it is valid for that reference to become nil at some point during its lifetime. Conversely, use an **unowned** reference when you know that the reference will never be nil once it has been set during initialization.” (Apple)*

SIGMA SOFTWARE
UNIVERSITY

Weak reference

“*a reference that does not keep a strong hold on an object and so does not stop ARC from disposing*” (Apple)

weak variable can be optional

Indicate as:

```swift
weak var myVariable: SomeObject?
```

SIGMA SOFTWARE UNIVERSITY

Unowned reference

“*a reference that does not keep a strong hold on an object and so does not stop ARC from disposing, but it assumed to always have a value*” (Apple)

unowned variable should always have value

Indicate as:

```
unowned var person: Person
```

Strong reference cycles for closures

“Define a capture in a closure as an unowned reference when the closure and the instance it captures will always refer to each other, and will always be deallocated at the same time.” (Apple)

```
closure = {
    self.string = "Hello, World! I'm immortal string and cause memory leak"
}


closure = { [unowned self] in
    self.string = "Hello, World! I will die soon :("
}
```

SIGMA SOFTWARE
UNIVERSITY

Capture list

“Each item in a capture list is a pairing of the weak or unowned keyword with a reference to a class instance (such as self) or a variable initialized with some value (such as delegate = self.delegate!)” (Apple)

```
lazy var someClosure: ( Int, String) ->  String = {
    [unowned self, weak delegate = self.delegate!] (index:  Int , stringToProcess:  String) ->  String in
    // closure body goes here - do some action here
}
```

SIGMA SOFTWARE UNIVERSITY

Список корисних ресурсів

About memory management (URL).

Мэтт Гэлловей - Сила Objective-C 2.0 - 2014. (Chapter 5).

Objective-C Memory Management Essentials By Gibson Tang, Maxim Vasilkov (978-1-84969-712-5).

А. Махер - "Программирование для iPhone. Высшый уровень" (розділ 1.3)

Список корисних ресурсів

Useful explanation week vs unowned (URL).

Objective-C Memory Management Essentials By Gibson Tang, Maxim Vasilkov (978-1-84969-712-5).

The Swift Programming Language - 2015, Apple Inc,
Chapter Automatic Reference Counting

Weak vs Unowned (URL)

sigma Software | UNIVERSITY