SCIENCES SORBONNE UNIVERSITÉ

Objective-C, autorelease pool

Fabrice.Kordon@lip6.fr

LIP6

F. Kordon - Sorbonne Université - CC2018

As an introduction

Dilemma when a method returns an object

- Situation 1

```
- (NSString*) identitySituation1 {
    NSString *res = [[NSString alloc]
                          initWithFormat:@"%d - %@", _number, _[illegible]e];
    return res; // no balance !!!
}
```

As an introduction

Dilemma when a method returns an object

- Situation 1

```
– (NSString*) identitySituation1 {
    NSString *res = [[NSString alloc]
                          initWithFormat:@"%d – %@", _number, _   e];
    return res; // no balance !!!
}
```

- Situation 2

```
– (NSString*) identitySituation2 {
    NSString *res = [[NSString alloc]
                          initWithFormat:@"%d – %@", _number,
    [res release];
    return res; // res's counter = 0!
}
```

No solution!

But «autorelease pool»

F. Kordon - Sorbonne Université - CC2018

Autorelease

Method = autorelease

- Puts the counter to 0
- Puts the object in a «temporary safe» place
- Until when? the end of the current event catch

How to solve our dilemma?

```
- (NSString*) identitySolution {
   NSString *res = [[NSString alloc]
                    initWithFormat:@"%d - %@", _number, _name];
   [res autorelease];
   return res; // res exist a little more... (caller may perform a retain)
}
```

- Let a chance to the caller to claim ownership

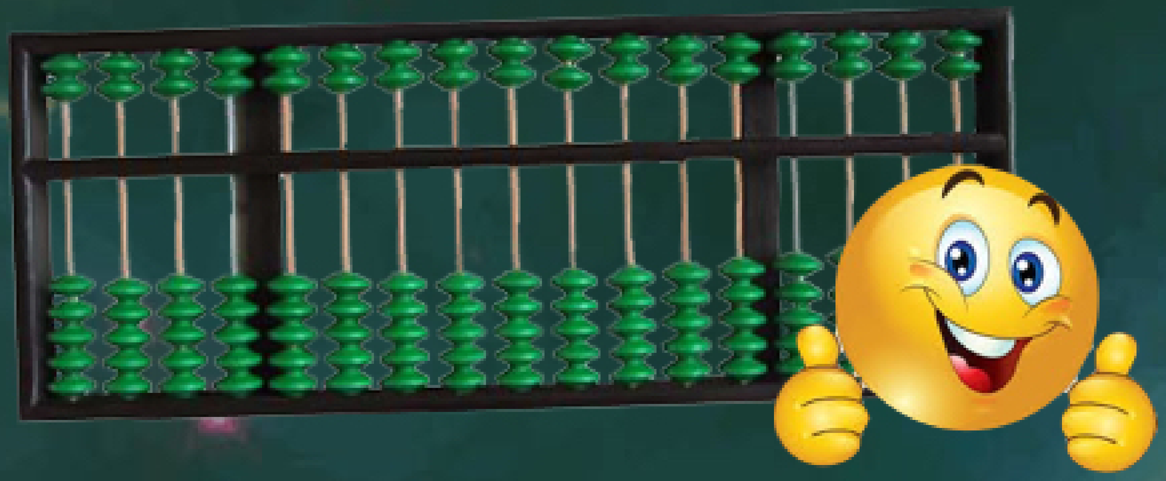
Some guidelines

- Important
  - Avoid retains (risks for leaks)
  - Solution, follow conventions
- Counter of an object returned by a method
  - Id the name contains **alloc**, **init** or **copy**
    - **Counter set to 1 (caller may release if needed)**
  - Otherwise
    - **Object is on autorelease pool**
- You must respect this convention
  - An autoreleased object can be retained
  - By the way, ARC respect such conventions

F. Kordon - Sorbonne Université - CC2018

As a conclusion…

- You know (almost) everything on this topic
  - Yet quite simple is'n't it?
- You just need to count



- You also need to think a bit
  - Thanks to your neurons