iOS Dev Camp #3 Week 4
Foundation Framework with collections

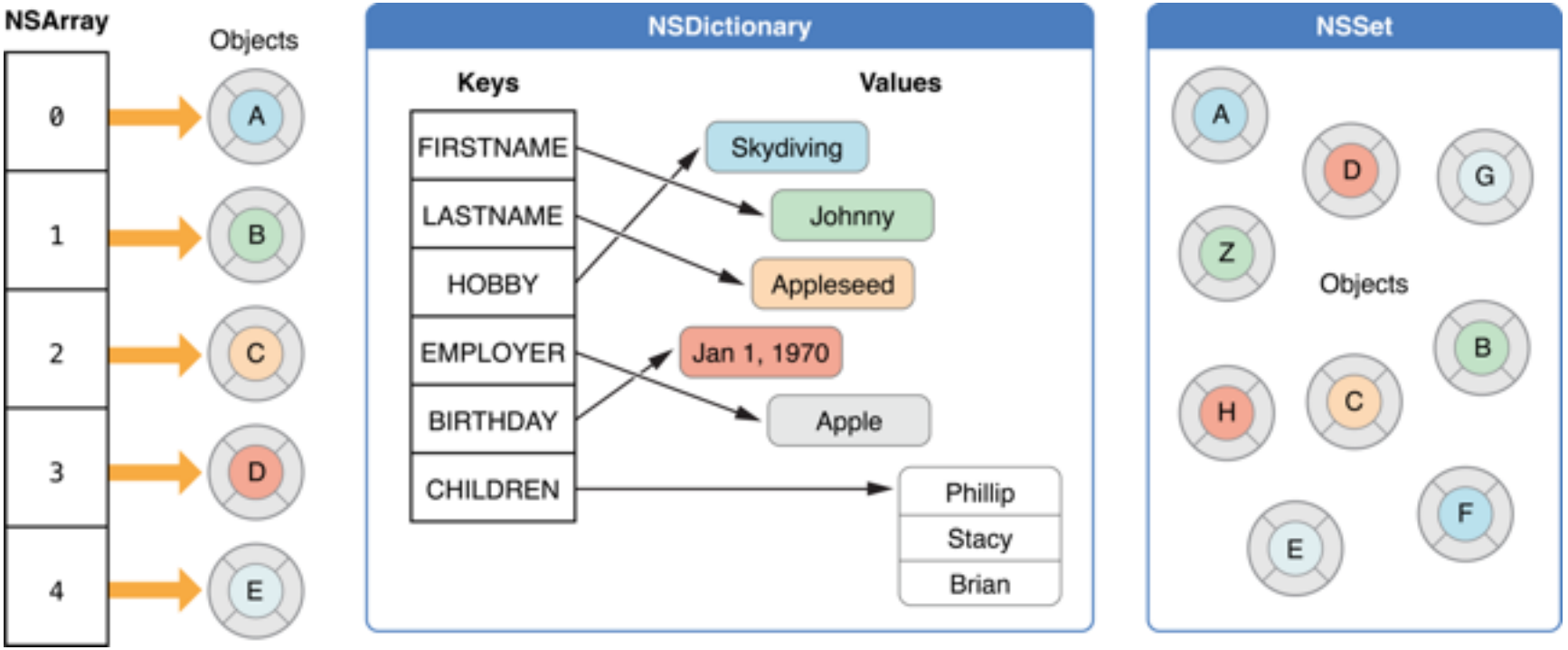
Edward Chiang

2014.10.20 - 2014.10.24

Today

- NSArray
- NSDictionary
- NSSet
- Copying / Enumeration
- NSUserDefaults

Store objects



Common tasks

All collections

- Enumerating the objects in a collection.
- Determining whether an object is in a collection.
- Accessing individual elements in a collection

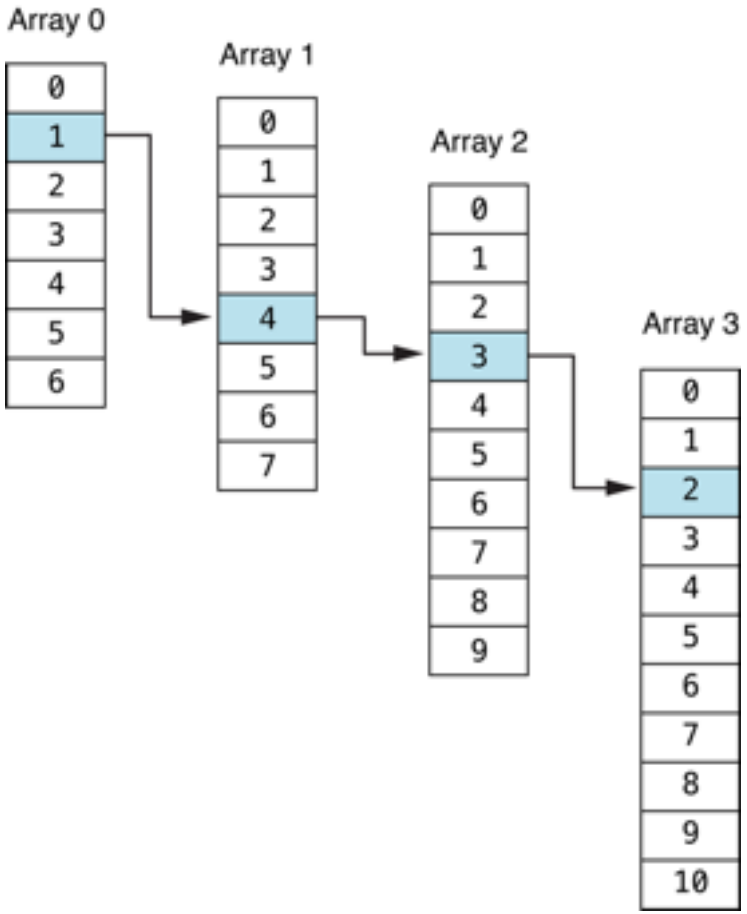
Mutable collections' tasks

- Adding objects to a collection.
- Removing objects from a collection.

Collections with particular task

- Array - Accessing indexes and easily enumerating elements.
- Dictionaries - Associating data with arbitrary keys.
- Sets - Offering fast insertion, deletion, and membership checks.
- Index Sets - Storing subsets fo arrays

Index Paths


Array 0
0
1
2
3
4
5
6
Array 1
0
1
2
3
4
5
6
7
Array 2
0
1
2
3
4
5
6
7
8
9
Array 3
0
1
2
3
4
5
6
7
8
9
10

NSArray

Array Fundamentals

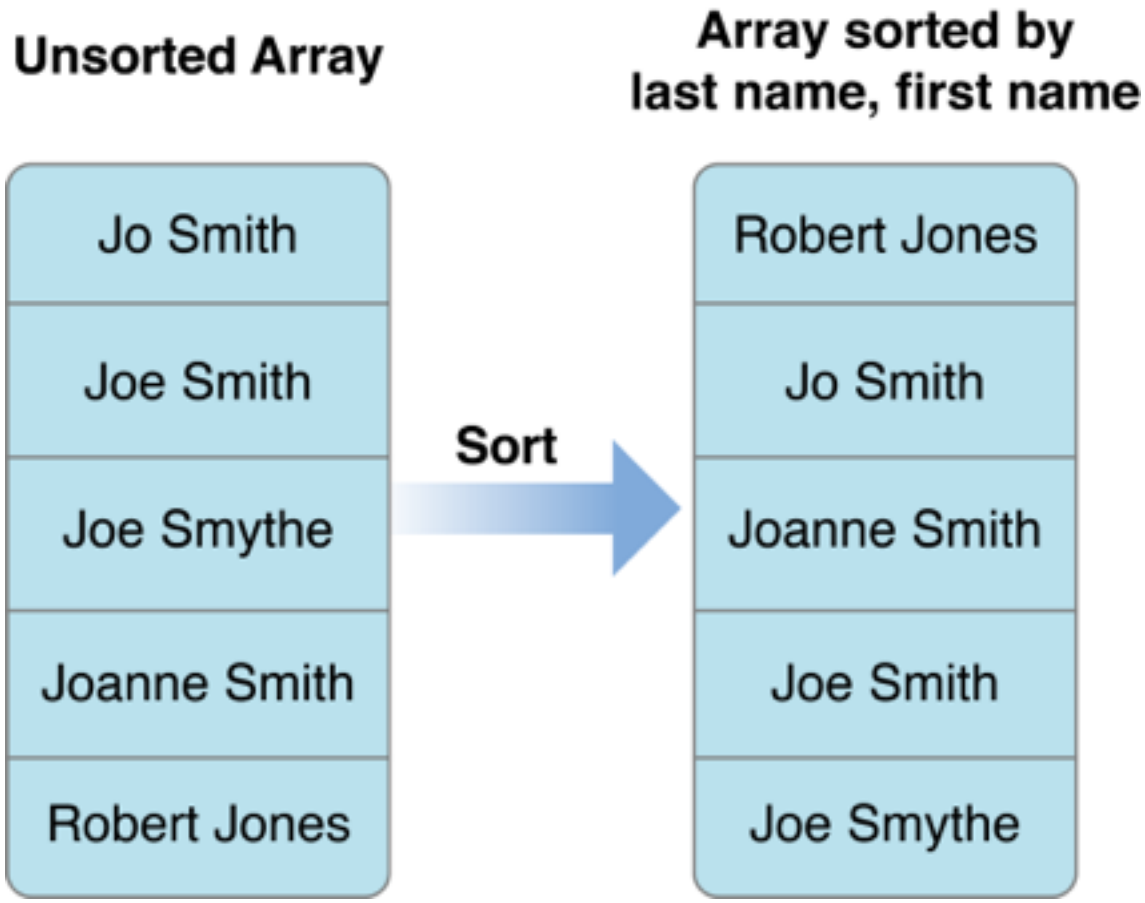
- An NSArray object manages an immutable array—that is, after you have created the array, you cannot add, remove, or replace objects.
- An NSMutableArray object manages a mutable array, which allows the addition and deletion of entries, allocating memory as needed.
- You can easily create an instance of one type of array from the other using the initializer initWithArray: or the convenience constructor arrayWithArray:.
- count returns the number of elements in the array.
- objectAtIndex: gives you access to the array elements by index, with index values starting at 0.

NSMutableArray

- addObject:
- insertObject:atIndex:
- removeLastObject
- removeObjectAtIndex:
- replaceObjectAtIndex:withObject:

- If you do not need an object to be placed at a specific index or to be removed from the middle of the collection, you should use the addObject: and removeLastObject methods because it is faster to add and remove at the end of an array than in the middle.

Sorting Arrays



Sorting with Sort Descriptors

- Sort descriptors (instances of NSSortDescriptor) provide a convenient and abstract way to describe a sort ordering. Sort descriptors provide several useful features. You can easily perform most sort operations with minimal custom code. You can also use sort descriptors in conjunction with Cocoa bindings to sort the contents of, for example, a table view. You can also use them with Core Data to order the results of a fetch request.
- If you use the methods sortedArrayUsingDescriptors: or sortUsingDescriptors:, sort descriptors provide an easy way to sort a collection of objects using a number of their properties. Given an array of dictionaries (custom objects work in the same way), you can sort its contents by last name then first name.

Sorting with Blocks

- The sortedArrayUsingComparator: method of NSArray sorts the array into a new array, using the block to compare the objects. NSMutableArray's sortUsingComparator: sorts the array in place, using the block to compare the objects.

NSDictionary

Dictionary Fundamentals

- An NSDictionary object manages an immutable dictionary—that is, after you create the dictionary, you cannot add, remove or replace keys and values.
- An NSMutableDictionary object manages a mutable dictionary, which allows the addition and deletion of entries at any time, automatically allocating memory as needed.
- You can easily create an instance of one type of array from the other using the initializer initWithDictionary: or the convenience constructor dictionaryWithDictionary:.
- In mutable, to add a single key-value pair, or to replace the object for a particular key, use the setObject:forKey: instance method

Using Custom Keys

- It may be necessary to use custom objects as keys in a dictionary.
- Dictionary copies each key, keys must conform to the NSCopying protocol. Bear this in mind when choosing what objects to use as keys. Although you can use any object that adopts the NSCopying protocol and implements the hash and isEqual: methods, it is typically bad design to use large objects, such as instances of NSImage, because doing so may incur performance penalties.

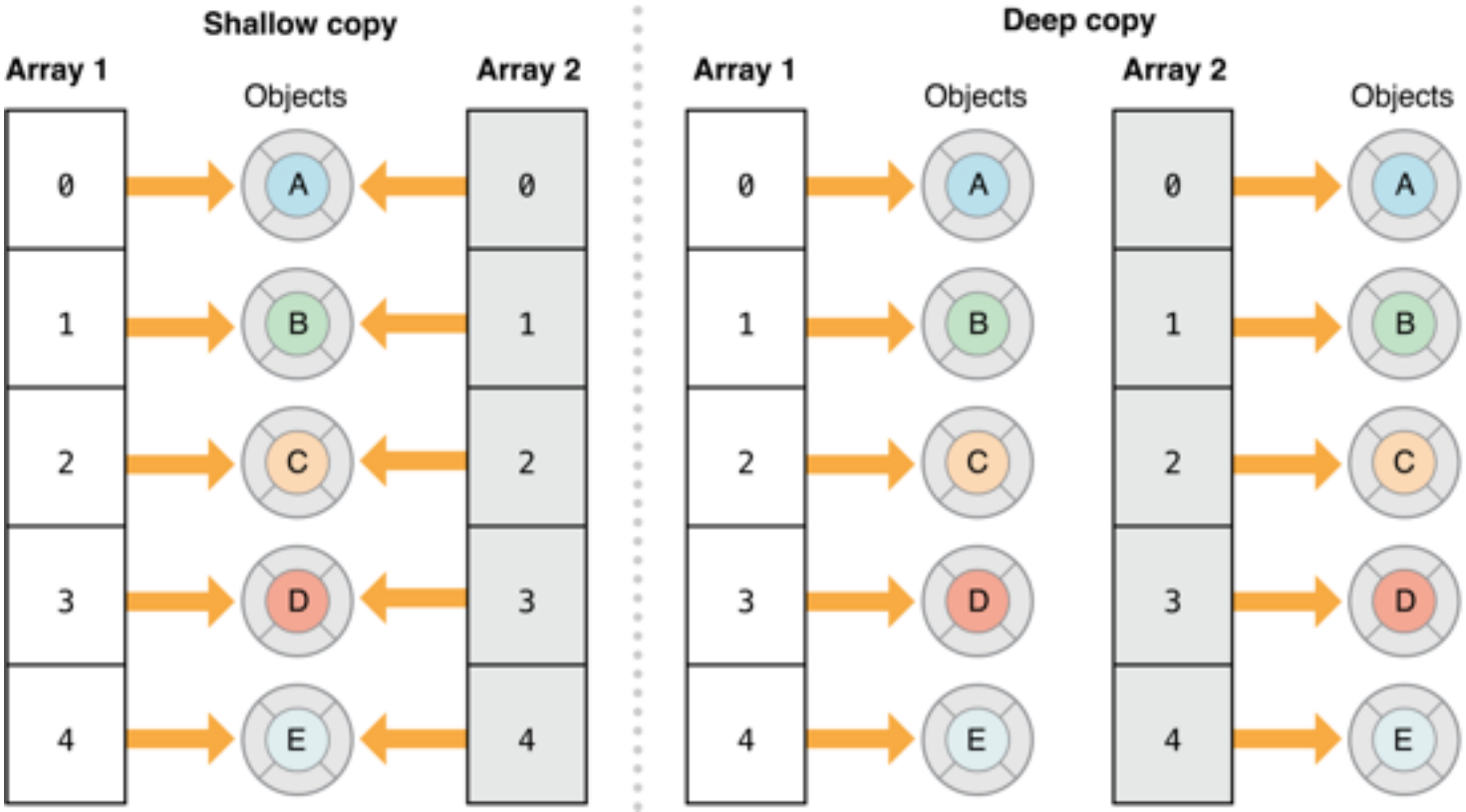
NSSet

Set Fundamentals

- An NSSet object manages an immutable set of distinct objects.
- NSMutableSet, a subclass of NSSet, is a mutable set of distinct objects, which allows the addition and deletion of entries at any time, automatically allocating memory as needed.
- NSCountedSet, a subclass of NSMutableSet, is a mutable set to which you can add a particular object more than once; in other words, the elements of the set aren't necessarily distinct. A counted set is also known as a bag. The set keeps a counter associated with each distinct object inserted. NSCountedSet objects keep track of the number of times objects are inserted and require that objects be removed the same number of times to completely remove the object from the set.

Copying Collections

Copying Collections



Enumeration

Fast Enumeration

- Fast enumeration is the preferred method of enumerating the contents of a collection because it provides the following benefits:
  - The enumeration is more efficient than using NSEnumerator directly.
  - The syntax is concise.
  - The enumerator raises an exception if you modify the collection while enumerating.
  - You can perform multiple enumerations concurrently.

Other two ways

- Using Block-based enumeration
- Using an Enumerator

Todays Homework

- Use NSArray, NSDictionary to
  - Enumerating the objects.
  - Determining whether an object is.
  - Accessing individual elements.
- Use NSMutableArray, NSMutableDictionary
  - Add, remove objects.
- Try out more copy and enumeration

~ END ~

http://www.alphacamp.tw