



Lập trình iOS

Bài 7. *Lưu trữ dữ liệu dạng tập hợp*

Ngành Mạng & Thiết bị di động





Nội dung

1. NSSet

- NSSet là gì?
- Mục đích sử dụng trong ứng dụng
- Các phương thức thường sử dụng

2. NSDictionary

3. NSEnumerator



1.1 NSSet là gì?

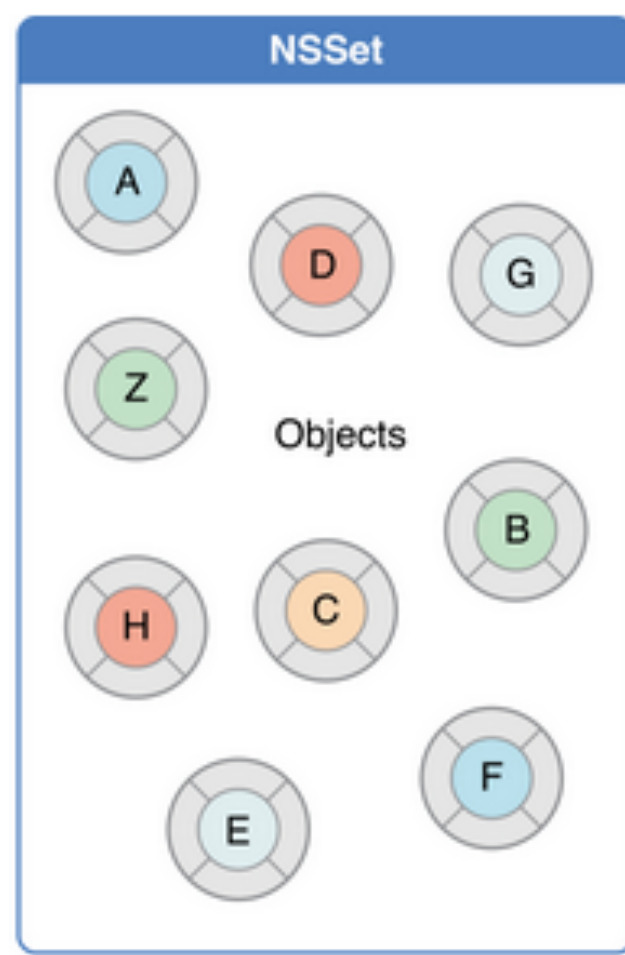
- ❑ Lớp NSSet và NSMutableSet là một tập hợp của các đối tượng không được sắp xếp.
- ❑ NSSet định nghĩa một tập hợp các đối tượng duy nhất và là bất biến đổi.
- ❑ NSMutableSet định nghĩa một tập hợp các đối tượng duy nhất và là biến đổi.



1.2 Mục đích sử dụng trong ứng dụng

❑ Ta sử dụng **NSSet** khi cần tạo một mảng các đối tượng mà:

- Thứ tự không quan trọng
- Các đối tượng là duy nhất
- Có hiệu suất tốt cho việc tìm kiếm





1.3 Các phương thức thường sử dụng

❑ Tạo một đối tượng NSSet

```
NSSet *array = [[NSSet alloc] initWithObjects:@"t",@"3",@"h",  
nil];
```

```
NSArray *myArray = @[:@"t",@"3",@"h", nil];  
NSSet *array2 = [[NSSet alloc] initWithArray:myArray];
```

❑ Lấy ra số lượng phần tử có trong mảng

```
NSLog(@"Mảng có %ld phần tử",[array count]); //Mảng có 3 phần tử  
NSLog(@"Mảng có %ld phần tử",array.count); //Mảng có 3 phần tử
```



1.3 Các phương thức thường sử dụng

- ❑ **Truy xuất phần tử mảng:** Ta chỉ có thể truy xuất đối tượng bằng cách duyệt qua mảng và so sánh để lấy đối tượng cần thiết

```
NSSet *array = [[NSSet alloc] initWithObjects: @"t", @"3",  
@"h", nil];  
for (NSString *string in array) {  
    if ([string isEqualToString:@"t"]) {  
        NSLog(@"%@",string);  
    }  
}
```

- ❑ **Kiểm tra phần tử có trong mảng:**

```
[array containsObject:@"1"];
```



1.3 Các phương thức thường sử dụng

- ❑ Ngoài các phương thức trên, sau đây là các phương thức chỉ hỗ trợ riêng cho NSMutableSet

- ❑ Thao tác thêm phần tử cho mảng:

```
[mutableArray addObject:@"trung tâm tin học"]; //Thêm mới 1 phần tử  
[mutableArray addObjectsFromArray:myArray]; //Thêm mới nhiều phần tử từ một danh sách kiểu NSArray
```

- ❑ Thao tác xóa phần tử:

```
[mutableArray removeAllObjects]; //Xóa tất cả phần tử có trong mảng  
[mutableArray removeObject:@"trung tâm tin học"]; //Xóa những phần tử có giá trị bằng với đối tượng truyền vào
```



Nội dung

1. NSSet

2. NSDictionary

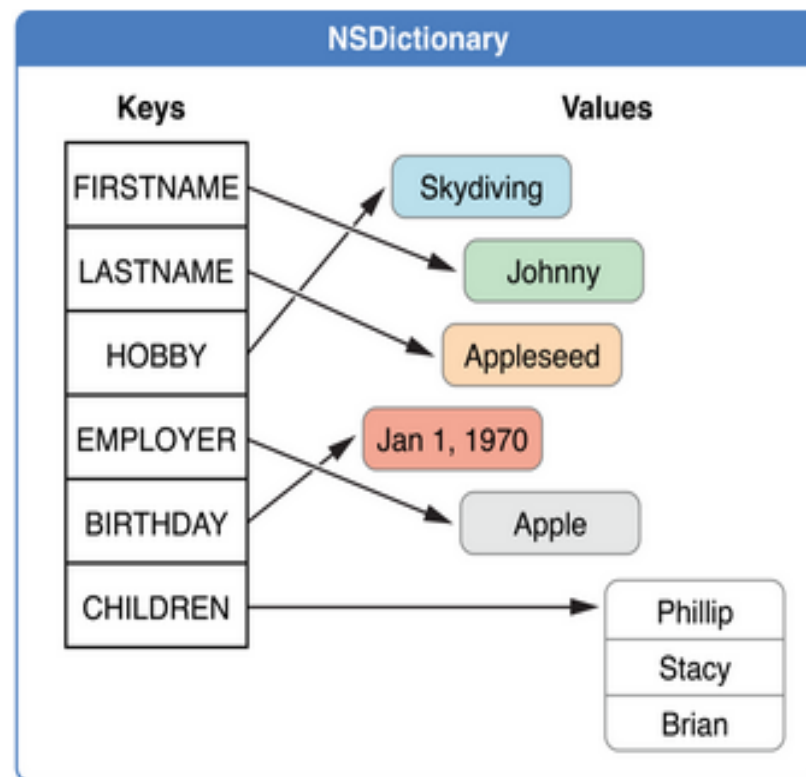
- NSDictionary là gì?
- Mục đích sử dụng trong ứng dụng
- Các phương thức thường sử dụng

3. NSEnumerator



2.1 NSDictionary là gì?

- ❑ Lớp NSDictionary là một tập hợp bất biến các đối tượng và quản lý các đối tượng dựa vào mối liên kết giữa khóa và giá trị.
- ❑ NSDictionary là tập hợp bất biến, còn NSMutableDictionary là tập hợp biến đổi các đối tượng.





2.2 Mục đích sử dụng trong ứng dụng

- ❑ Ta sử dụng NSDictionary với những dữ liệu tương tự từ điển.
- ❑ Trong một số trường hợp việc truy xuất đến một phần tử kiểu NSDictionary nhanh hơn so với NSArray.

Key	Type	Value
▼ Root	Dictionary	(1 item)
▼ Routes	Dictionary	(2 items)
▼ Houston	Dictionary	(3 items)
▼ Location 1	Dictionary	(3 items)
name	String	Fresno
type	String	Well
▼ coordinate	Dictionary	(2 items)
latitude	String	29.5386
longitude	String	-95.319099
► Location 2	Dictionary	(3 items)
► Location 3	Dictionary	(3 items)
► San Antonio	Dictionary	(1 item)



2.3 Các phương thức thường sử dụng

❑ Khởi tạo mảng

```
NSDictionary *dic = [[NSDictionary alloc]
initWithObjectsAndKeys: @"giá trị 1", @"khóa 1", @"giá trị 2",
@"khóa 2", nil];
NSDictionary *dic2 = [NSDictionary
dictionaryWithContentsOfFile: filePath];
```

❑ Lấy ra số lượng phần tử có trong mảng

```
NSLog(@"Mảng có %ld phần tử", [dictionary count]);
NSLog(@"Mảng có %ld phần tử", dictionary.count);
```

❑ Truy xuất phần tử mảng:

```
NSLog(@"Phần tử đầu tiên có giá trị là %@",
[dictionary objectForKey: @"khóa 2"]); // giá trị 2
NSLog(@"Phần tử thứ 2 có giá trị là %@", [dictionary
valueForKey: @"khóa 1"]); // giá trị 1
```



2.3 Các phương thức thường sử dụng

❑ Các phương thức chỉ hỗ trợ riêng cho NSMutableDictionary

- Thao tác thêm phần tử cho mảng:

```
[mutableDictionary setValue:@"giá trị 3" forKey:@"khóa 3"];  
[mutableDictionary setObject:@"giá trị 3" forKey:@"khóa  
3"];
```

- Thao tác xóa phần tử:

```
[mutableDictionary removeObjectForKey:@"khóa"];  
[mutableDictionary removeAllObjects];
```



Nội dung

1. NSSet

2. NSDictionary

3. NSEnumerator

- NSEnumerator là gì?
- Các phương thức thường sử dụng



3.1 NSEnumerator là gì?

- ❑ Lớp NSEnumerator là một lớp trừu tượng, mà lớp con liệt kê các tập hợp của các đối tượng khác như mảng.
- ❑ Tất cả phương thức khởi tạo được định nghĩa trong các lớp tập hợp như NSArray, NSSet và NSDictionary.
- ❑ Ta không thể sử dụng lại đối tượng NSEnumerator khi đã dùng rồi.



3.2 Các phương thức thường sử dụng

❑ Tạo NSEnumerator từ NSArray

```
NSArray *aArray = [NSArray arrayWithObjects: @"John", @"Bob",  
@"Jane", nil];  
NSEnumerator *enumerator = [aArray objectEnumerator];
```

❑ Lấy phần tử trong NSEnumerator

```
id obj;  
while(obj = [enumerator nextObject])  
    NSLog(@"%@", obj);
```

❑ Lấy tất cả phần tử trong NSEnumerator

```
NSLog(@"%@", [enumerator allObjects]);
```



Nội dung

1. NSSet

2. NSDictionary

3. NSEnumerator

4. NSHashTable

- NSHashTable là gì?
- Các phương thức thường sử dụng



4.1 NSHashTable là gì?

- ❑ Lớp NSHashTable được mô hình hóa sau NSMutableSet nhưng cung cấp các lựa chọn khác nhau, đặc biệt là có hỗ trợ tham chiếu yếu.
- ❑ Các phần tử có thể được sao chép ở đầu vào hoặc sử dụng con trỏ định danh cho việc so sánh và băm.
- ❑ Nó có thể chứa con trỏ tùy (các phần tử không cần phải là các đối tượng).



4.2 Các phương thức thường sử dụng

❑ Khởi tạo NSDictionary

```
NSDictionary *hashTable = [NSDictionary dictionaryWithOptions:  
NSDictionaryWeakMemory];
```

❑ Thêm một phần tử

```
[hashTable addObject:@"foo"];
```

❑ Xóa một phần tử

```
[hashTable removeObject:@"bar"];
```

❑ Lấy các phần tử trong mảng

```
NSLog(@"Members: %@", [hashTable allObjects]);
```

Thảo luận

