

Разработка приложений для OS X

Лекция I
Описание курса. Введение. XCode

Hexlet University, 2012

Модуль I

Здравствуйте!

Меня зовут Рахим Давлеткалиев, BSc

freetonik@hexlet.ru

<http://freetonik.com>

О курсе

- Язык программирования Objective C
- Среда разработки XCode
- Foundation Framework
- Cocoa API
- Основы разработки iOS-приложений (для iPhone и iPad)
- Основы хорошего дизайна приложений

Структура курса

- 15 лекций
- 5 коротких тестов
- 4 домашних задания
- Промежуточный экзамен в середине семестра (midterm)
- Финальный экзамен в конце семестра (final)

далее – Модуль II

Что такое Objective C?

- Объектно-ориентированный язык с системой сообщений в стиле Smalltalk
- Используется Apple в OS X и iOS

Hello, world!



далее – Модуль III

тип id

- хранение объекта любого типа (класса)

```
int height;  
float temperature;  
id graphicObject;
```

Файлы

- .h – header files
- .m – source files
- .mn – source files if C++ is present

Определение класса

Class
name

Parent class
name

```
@interface MyClass : NSObject
```

```
{
```

```
    int        count;
```

```
    id         data;
```

```
    NSString*  name;
```

```
}
```

```
- (id)initWithString:(NSString*)aName;
```

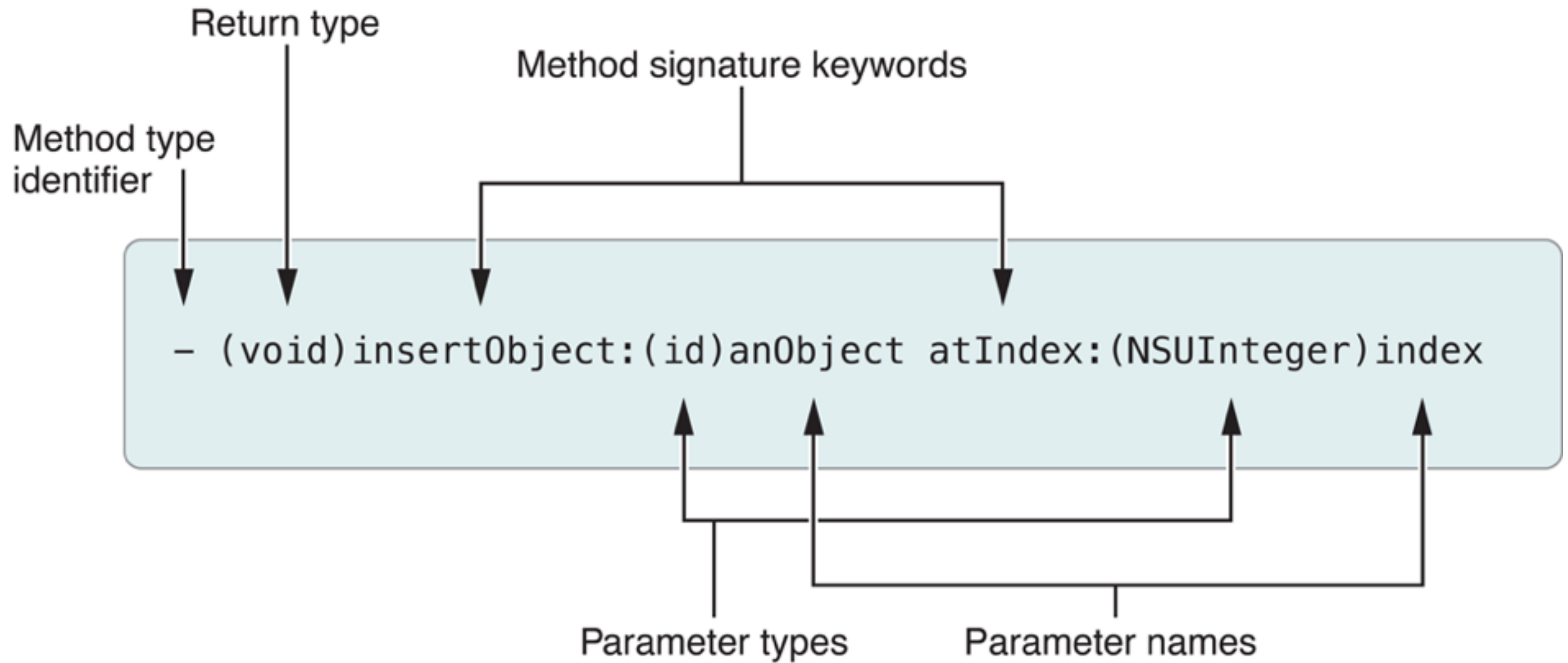
```
+ (MyClass*)createClassWithString:(NSString*)aName;
```

```
@end
```

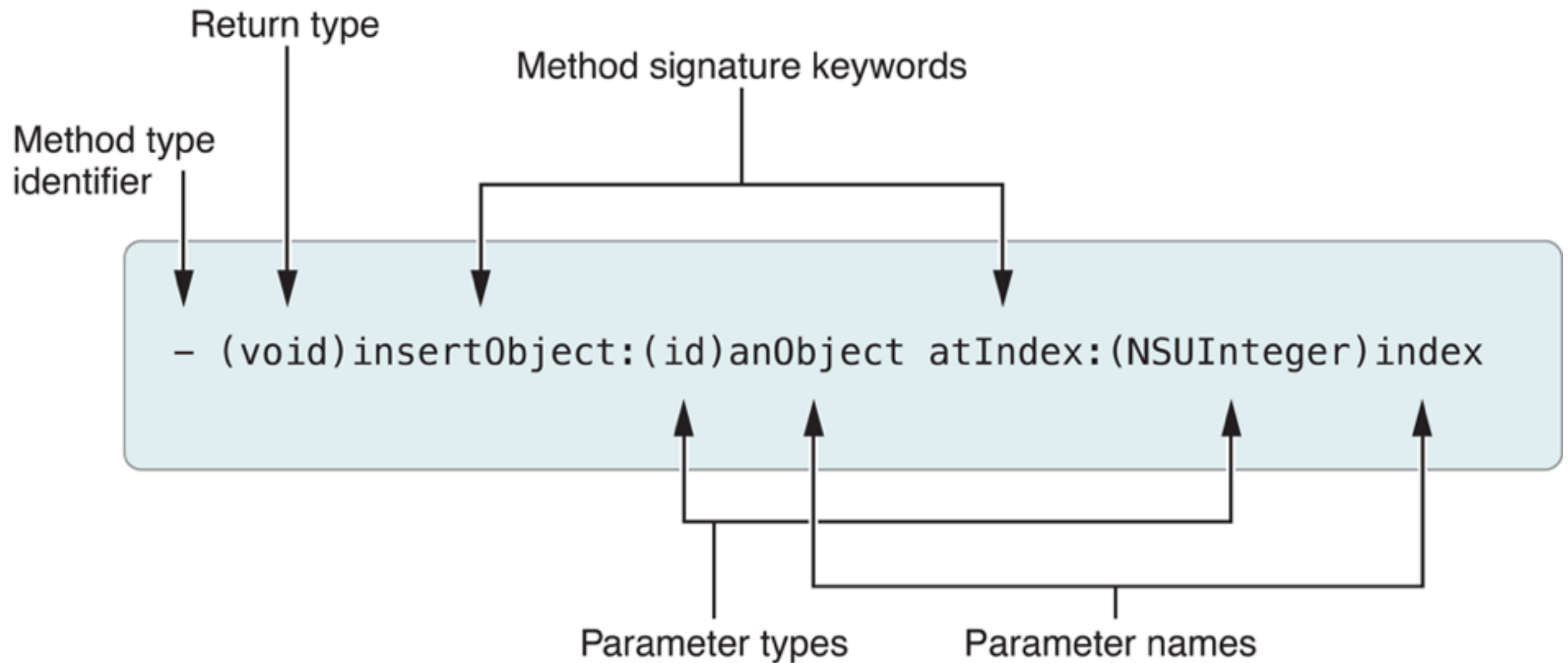
Member variable
declarations

Method
declarations

Методы



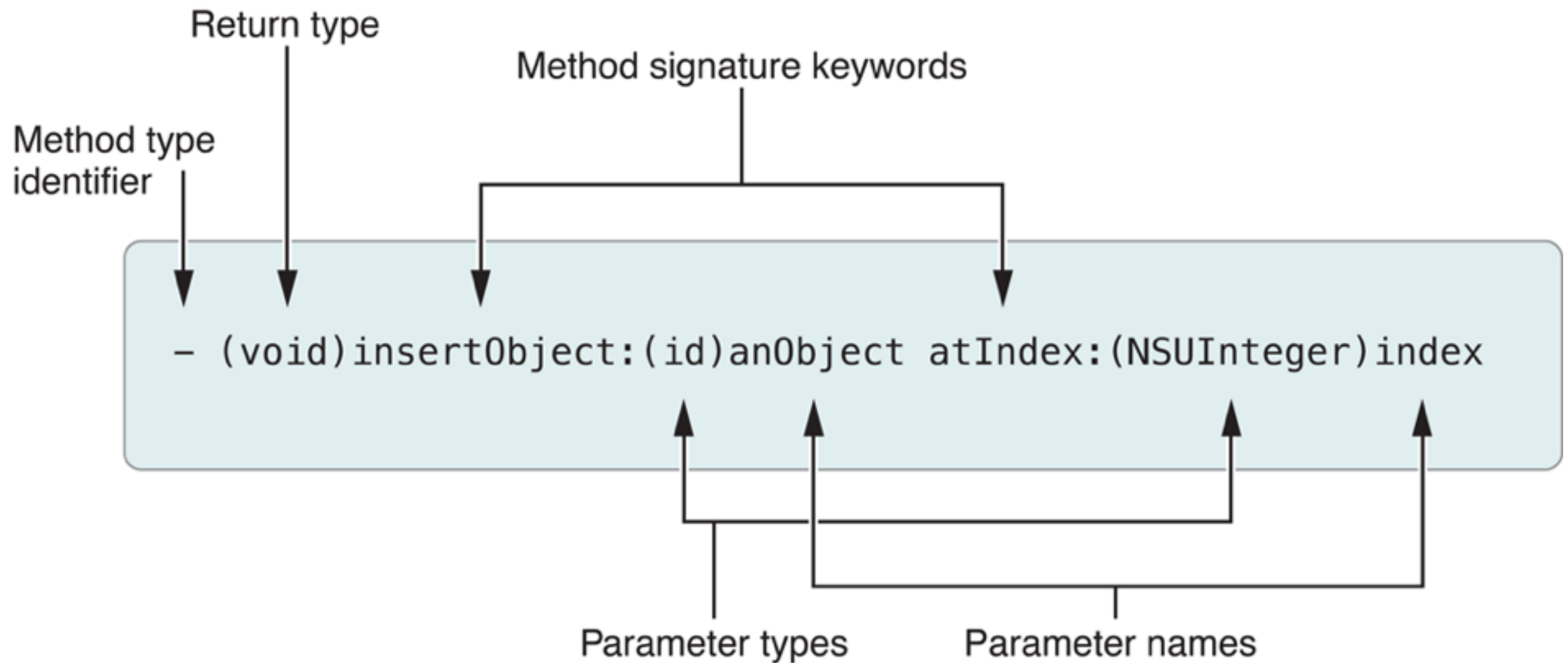
Методы



vs

```
void insertObjectAtIndex (object, index);
```

Методы



method's name is
insertObject:atIndex:

Методы

[Class method];

[Instance method];

[receiver message];

myCar = [Car new];

[myCar start];

пример

```
[myArray insertObject:anObject atIndex:0];
```

пример

```
[myArray insertObject:myObject atIndex:0];
```

ВЛОЖЕНИЯ

```
[[myAppObject theArray] insertObject:[myAppObject objectToInsert] atIndex:0];
```

```
[anArray insertObject:[myAppObject objectToInsert] atIndex:0];
```

```
[anArray insertObject:anObject atIndex:0];
```

dot notation

```
[myAppObject.theArray insertObject:myAppObject.objectToInsert atIndex:0];
```


Objective C vs C++

```
@interface Human : Animal {  
    // instance variables  
    int age;  
}  
+ classMethod1;  
+ (return_type)classMethod2;  
+ (return_type)classMethod3:(param1_type)param1_varName;  
  
- (return_type)instanceMethod1:(param1_type)param1_varName :(param2_type)param2_varName;  
- (return_type)instanceMethod2WithParameter :(param1_type)param1_varName otherParameter:  
  (param2_type)param2_varName;  
@end
```

```
class Human : public Animal {  
    protected:  
        // instance variables  
        int age;  
  
    public:  
        // Class (static) functions  
        static void * classMethod1();  
        static return_type classMethod2();  
        static return_type classMethod3(param1_type param1_varName);  
  
        // Instance (member) functions  
        return_type instanceMethod1 (param1_type param1_varName, param2_type param2_varName);  
        return_type instanceMethod2WithParameter (param1_type param1_varName, param2_type  
param2_varName=default);  
};
```

@implementation

```
@implementation NewClassName {  
    memberDeclarations;  
}
```

```
methodDefinitions;  
@end
```

FractionsClass (пример): @interface

```
#import <Foundation/Foundation.h>
```

```
@interface Fraction: NSObject
```

```
- (void) print;  
- (void) setNumerator: (int) n;  
- (void) setDenominator: (int) d;
```

```
@end
```

FractionsClass (пример): @implementation

```
@implementation Fraction {
    int numerator;
    int denominator;
}

-(void) print {
    NSLog ("%i/%i", numerator, denominator);
}

-(void) setNumerator: (int) n {
    numerator = n;
}

-(void) setDenominator: (int) d {
    denominator = d;
}
@end
```

FractionsClass example: program code

```
int main (int argc, char * argv[]) {
    @autoreleasepool {
        Fraction *myFraction;

        // Create an instance of a Fraction
        myFraction = [Fraction alloc];
        myFraction = [myFraction init];

        // Set fraction to 1/3
        [myFraction setNumerator: 1];
        [myFraction setDenominator: 3];

        // Display the fraction using the print method
        NSLog (@"The value of myFraction is:");
        [myFraction print];
    }

    return 0;
}
```