

Блоки и МНОГОПОТОЧНОСТЬ

GCD, потоки, объекты синхронизации

Блоки

- Замыкания от Apple для C, Objective-C и C++.
- По применению похожи на анонимные функции или функции обратного вызова (callback), но гораздо шире.
- <http://goshdarnblocksyntax.com/>

- Можно хранить в переменной.

```
returnType(^blockName)(parameterTypes) =  
^returnType(parameters) {...};
```

- Можно хранить как свойство объекта.

```
@property (weak) returnType(^blockName)  
(parameterTypes);
```

- Корректно работают с ARC (weak/strong связи, подсчет ссылок).
- Можно использовать как замыкание, тогда в теле блока возможен захват окружающих его переменных.
- Можно вызвать как функцию.

Возможности блоков

```
// Объявление блока как переменной
double (^multiplyTwoValues)(double, double) =
// Литерал для блока такого же типа
(определение):
^double(double firstValue, double secondValue) {
    return firstValue * secondValue;
}
```

```
//Теперь блок можно вызвать с параметрами:
double result = multiplyTwoValues(4.0, 6.0);
```

Пример работы с блоками

```
@interface ImageDownloader : NSObject
@property (nonatomic, copy) void(^completionBlock)(UIImage *);
- (void)downloadImageNamed:(NSString *)imageName;
@end // Объявление некоего класса
```

// Использование:

```
NSString *imageName = @"defaultImage";
__block UIImage *image = [UIImage imageNamed:imageName];
if (image == nil) {
    ImageDownloader *downloader = [[ImageDownloader alloc] init];
    downloader.completionBlock = ^(UIImage *newImage) {
        image = newImage;
    };
    [downloader downloadImageNamed:imageName];
}
```

Пример блоков как замыканий с захватом переменных

Grand Central Dispatch (GCD)

- Реализует выполнение параллельных задач.
- Использует код внутри функций или блоков как код задачи.
- Задача может выполняться в потоке-“очереди” (системной или специально созданной).

Примитивы GCD

Dispatch Queues — объекты, поддерживающие очереди задач, и запускающие эти задачи в порядке очереди.

Dispatch Groups — объекты, позволяющие объединять задачи в группы для последующего объединения (joining).

Dispatch Semaphores — объекты, которые позволяют не более, чем определенному числу задач выполняться одновременно.

Dispatch Sources — объекты, которые позволяют регистрировать блоки или функции для их асинхронного выполнения при срабатывании определенного события.

```
void dispatch_async ( dispatch_queue_t queue, dispatch_block_t block );
```

Добавляет блок-задачу в пул очереди для асинхронного выполнения и сразу возвращает управление.

```
void dispatch_sync ( dispatch_queue_t queue, dispatch_block_t block );
```

Добавляет блок-задачу в пул очереди для синхронного выполнения и сразу возвращает управление.

```
void dispatch_after ( dispatch_time_t when, dispatch_queue_t queue,  
dispatch_block_t block );
```

Выполняет блок-задачу на определенной очереди спустя определенное время.

Распространенные функции GCD