

# Разработка приложений для iOS

## Лекция 3

# Views

**Глеб Тарасов**  
[gleb34@gmail.com](mailto:gleb34@gmail.com)

# Вспомним прошлое занятие

Чем отличаются строки  
"string1" и @"string2"

Чем отличаются строки  
"string1" и @"string2"

Char\*

C

NSString\*

Objective-C

**Как сохранить число 5 в массив?**

Как сохранить число 5 в массив?

NSNumber

@(5)

# Что значит минус перед определением

– `(void)deleteProfile;`

Что значит минус  
перед определением

– `(void)deleteProfile;`

*Метод объекта, а не класса*



# Как создать объект?

User \*user = ...

# Как создать объект?

```
User *user = ...
```

```
User *user = [[User alloc] init];
```

**Какой базовый класс  
для всех объектов в  
Objective-C?**

Какой базовый класс  
для всех объектов в  
Objective-C?

NSObject

Как вызвать метод у  
объекта `user` с  
параметрами «topic»,  
«text»

```
– (void)postCommentWithTopic:(NSString *)topic  
    andText:(NSString *)text;
```

# Как вызвать метод у объекта `user` с параметрами «topic», «text»

– `(void)postCommentWithTopic:(NSString *)topic  
andText:(NSString *)text;`

```
[user postCommentWithTopic:@"topic"  
andText:@"text"];
```

# Из чего состоит свойство?

```
@property(nonatomic) NSInteger age;
```

# Из чего состоит свойство?

```
@property(nonatomic) NSInteger age;
```

*getter, setter, внутренняя  
переменная*



# Ещё немного СИНТАКСИСА

# ПРОТОКОЛЫ

```
@protocol SendMessageProtocol
```

```
- (void)sendMessage:(NSString *)message;
```

```
@end
```

```
@interface User : NSObject<SendMessageProtocol>
```

```
@end
```

```
@implementation User
```

```
- (void)sendMessage:(NSString *)message
```

```
{
```

```
    // send message
```

```
}
```

```
@end
```

```
id<SendMessageProtocol> sender = [[User alloc] init];  
[sender sendMessage:@"message"];
```

# Модификаторы на свойствах

- strong
- weak

```
@interface Car : NSObject
```

```
@property(strong, nonatomic) NSArray *wheels;
```

```
@end
```

```
@interface Wheel : NSObject
```

```
@property(weak, nonatomic) Car *car;
```

```
@end
```

Country

*strong*

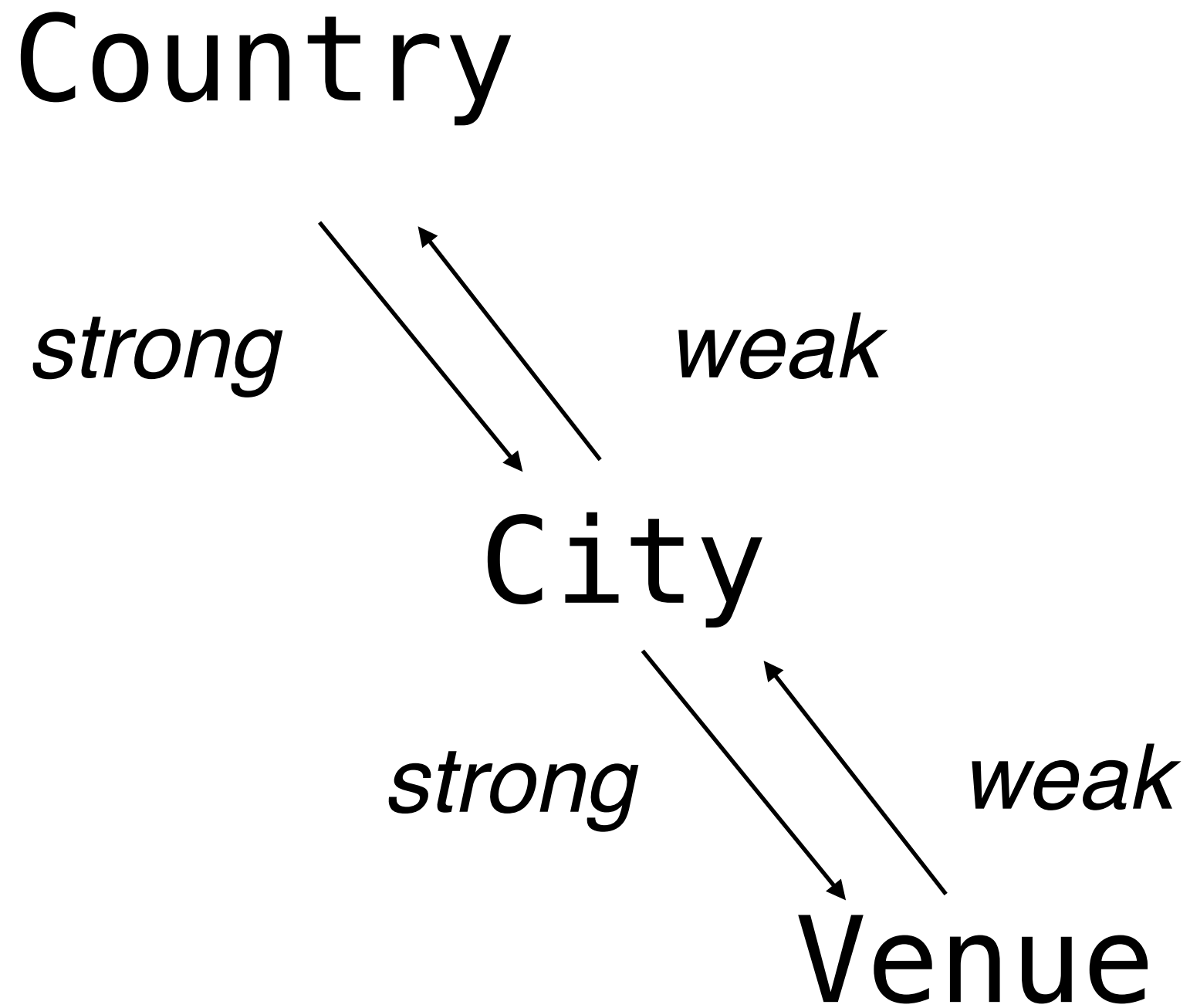
*weak*

City

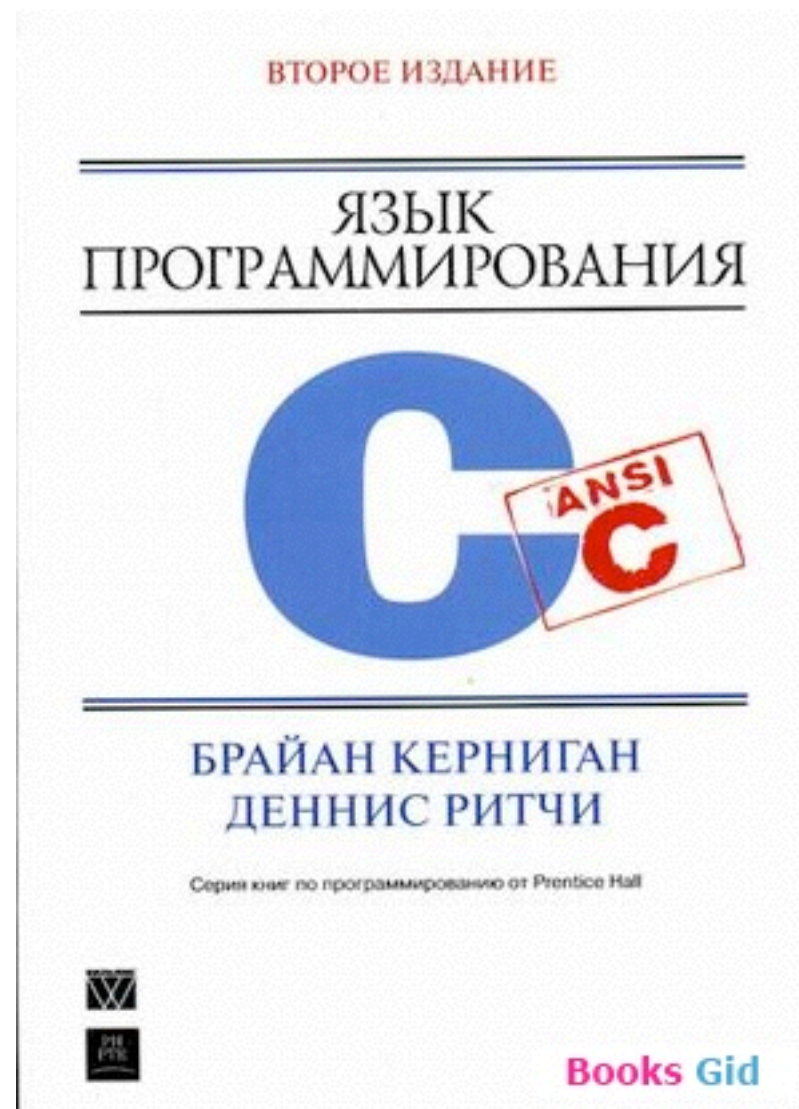
*strong*

*weak*

Venue



# Си



[http://ru.wikipedia.org/wiki/Си\\_\(язык\\_программирования\)](http://ru.wikipedia.org/wiki/Си_(язык_программирования))

# Структуры

```
CGPoint p;  
p.x = 10;  
p.y = 20;  
p = CGPointMake(10, 20);
```

```
CGSize s;  
s.width = 100;  
s.height = 100;  
s = CGSizeMake(100, 100);
```

```
CGRect r;  
r.size = s;  
r.origin = p;  
r = CGRectMake(10, 20, 100, 100);
```



# Структуры

```
typedef struct
{
    float x;
    float y;
} Location;
```

```
Location createLocation(float x, float y)
{
    Location l;
    l.x = x;
    l.y = y;
    return l;
}
```

```
int main()
{
    Location l = createLocation(1.5, 0.5);
    printf("location: {%g, %g}", l.x, l.y);
}
```

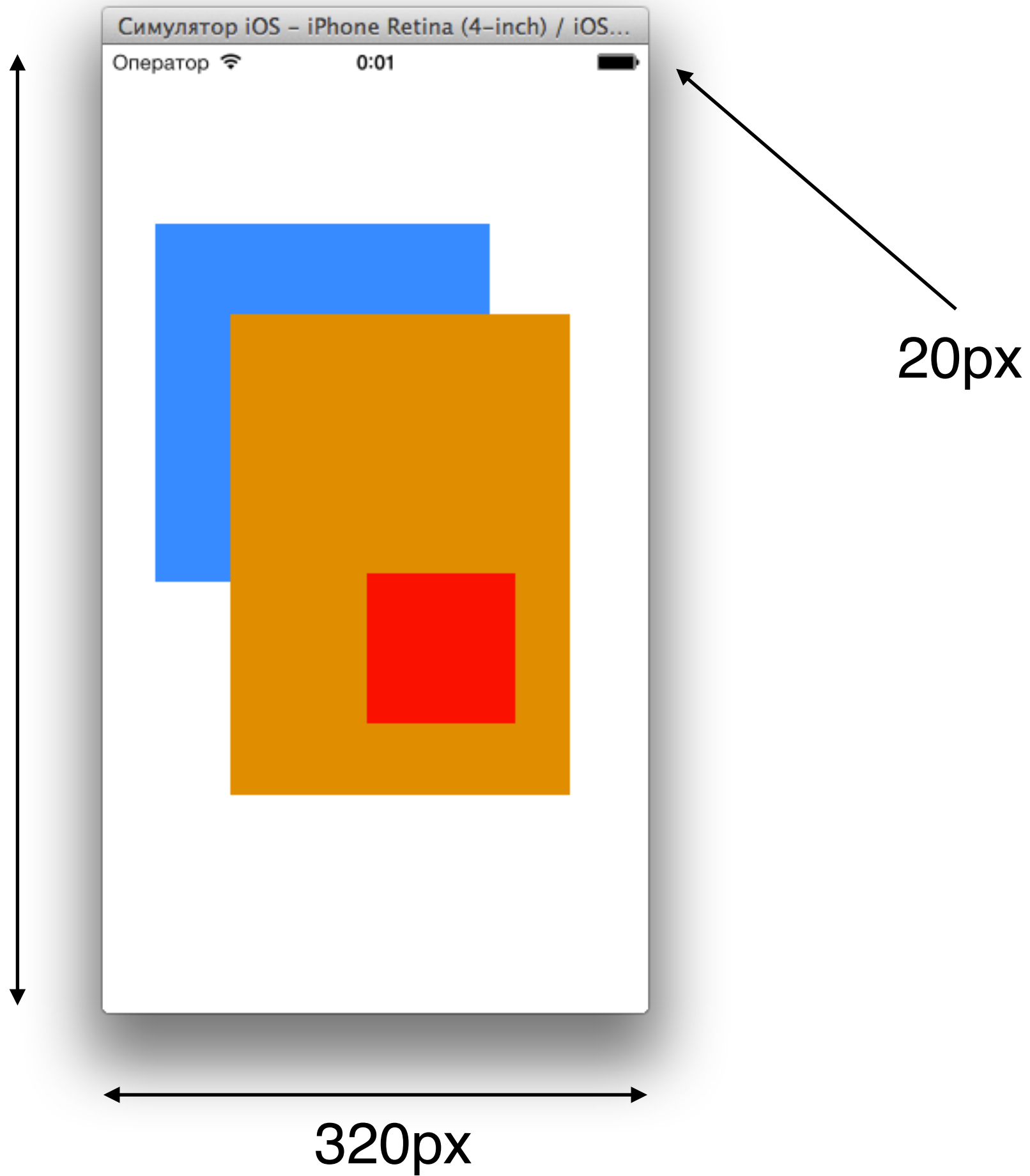
# Перечисления

```
typedef enum
{
    UIViewAnimationCurveEaseInOut,
    UIViewAnimationCurveEaseIn,
    UIViewAnimationCurveEaseOut,
    UIViewAnimationCurveLinear
} UIViewAnimationCurve;
```

```
UIViewAnimationCurve a = UIViewAnimationCurveLinear;
printf("%d", a);
```

**View**

iPhone 5: 568px  
iPhone 4: 480px



# UIView

```
@property(n nonatomic) CGRect frame;
```

```
@property(n nonatomic, readonly) UIView *superview;
```

```
@property(n nonatomic, readonly, copy) NSArray *subviews;
```

```
- (void)removeFromSuperview;
```

```
- (void)addSubview:(UIView *)view;
```

```
- (void)bringSubviewToFront:(UIView *)view;
```

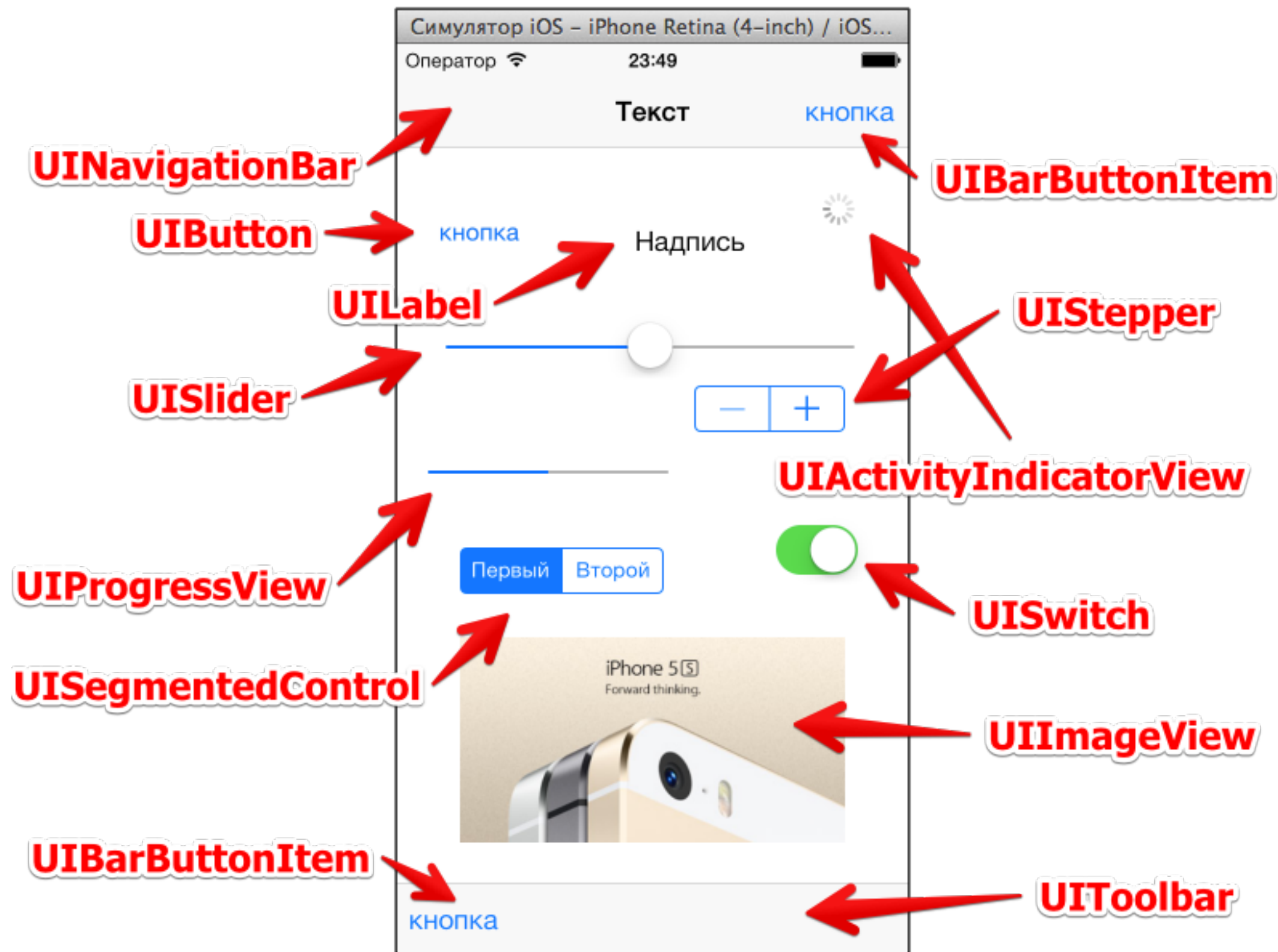
```
- (void)sendSubviewToBack:(UIView *)view;
```

```
@property(n nonatomic, copy) UIColor *backgroundColor;
```

```
@property(n nonatomic) CGFloat alpha;
```

```
@property(n nonatomic, getter=isHidden) BOOL hidden;
```

# Стандартные контролы




# AppDelegate.m

```
@implementation AppDelegate
```

```
- (BOOL)application:(UIApplication *)application  
    didFinishLaunchingWithOptions:(NSDictionary *)  
    launchOptions  
{  
    CGRect frame = [[UIScreen mainScreen] bounds];  
    self.window = [[UIWindow alloc] initWithFrame:frame];  
    self.window.backgroundColor = [UIColor whiteColor];  
  
    UIView *view2 = [[UIView alloc] init];  
    view2.frame = CGRectMake(30, 30, 200, 200);  
    view2.backgroundColor = [UIColor redColor];  
    [self.window addSubview:view2];  
  
    [self.window makeKeyAndVisible];  
  
    return YES;  
}
```

Симулятор iOS - iPhone Retina (4-inch) / iOS...

Оператор 

13:37





**Controller**

# UIViewController

```
@property(n nonatomic, retain) UIView *view;
```

```
- (void) viewDidLoad;
```

```
- (void) viewWillAppear: (BOOL) animated;
```

```
- (void) viewWillDisappear: (BOOL) animated;
```

# AppDelegate.m

```
#import "AppDelegate.h"
#import "ViewController.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    CGRect frame = [[UIScreen mainScreen] bounds];
    UIWindow *window = [[UIWindow alloc] initWithFrame:frame];
    ViewController *vc = [[ViewController alloc] init];
    window.rootViewController = vc;
    [window makeKeyAndVisible];
    return YES;
}
```

# ViewController.m

```
#import "ViewController.h"

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    UIView *view2 = [[UIView alloc] initWithFrame:
        CGRectMake(40, 40, 100, 100)];
    view2.backgroundColor = [UIColor blueColor];
    [self.view addSubview:view2];
}


@end
```

# UILabel

```
- (void) viewDidLoad
{
    [super viewDidLoad];

    UILabel *l = [[UILabel alloc] initWithFrame:
        CGRectMake(50, 50, 200, 200)];
    l.text = @"Текст";
    l.font = [UIFont systemFontOfSize:34];
    l.textColor = [UIColor redColor];
    [self.view addSubview:l];
}
```

Симулятор iOS – iPhone Retina (4-inch) / iOS...

Оператор 

0:23



Текст

**Демонстрация**

**События**



# target, action

```
- (void) viewDidLoad
{
    [super viewDidLoad];

    [self.button addTarget:self
                    action:@selector(buttonClicked)
                    forControlEvents:UIControlEventTouchUpInside];
}

- (void) buttonClicked
{
    NSLog(@"buttonClicked");
}
```

# delegate

```
@interface ViewController : UIViewController<UITextFieldDelegate>
- (void)viewDidLoad
{
    [super viewDidLoad];

    self.textField.delegate = self;
}
- (BOOL)textFieldShouldReturn:(UITextField *)textField
{
    NSLog(@"return pressed");
}
@end
```

# UITextFieldDelegate

```
@protocol UITextFieldDelegate <NSObject>
```

```
@optional
```

- (BOOL)textFieldShouldBeginEditing:(UITextField \*)textField;
- (void)textFieldDidBeginEditing:(UITextField \*)textField;
- (BOOL)textFieldShouldEndEditing:(UITextField \*)textField;
- (void)textFieldDidEndEditing:(UITextField \*)textField;
- (BOOL)textField:(UITextField \*)textField  
shouldChangeCharactersInRange:(NSRange)range replacementString:  
(NSString \*)string;
- (BOOL)textFieldShouldClear:(UITextField \*)textField;
- (BOOL)textFieldShouldReturn:(UITextField \*)textField;

```
@end
```

# Отрывок UIScrollViewDelegate

```
@protocol UIScrollViewDelegate
```

```
@optional
```

```
// any offset changes
```

```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView;
```

```
// any zoom scale changes
```

```
- (void)scrollViewDidZoom:(UIScrollView *)scrollView  
__OSX_AVAILABLE_STARTING(__MAC_NA, __IPHONE_3_2);
```

```
// called on start of dragging
```

```
// (may require some time and or distance to move)
```

```
- (void)scrollViewWillBeginDragging:(UIScrollView *)scrollView;
```

```
...
```

# notifications

*Событие случилось:*

```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center postNotificationName:@"UserChanged" object:nil];
```

*ЛОВИМ событие:*

```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center addObserver:self  
             selector:@selector(userChanged)  
             name:@"UserChanged"  
             object:nil];
```

```
- (void)userChanged  
{  
    NSLog(@"Событие произошло");  
}
```

```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center addObserver:self  
             selector:@selector(keyboardShown)  
             name:UIKeyboardDidShowNotification  
             object:nil];
```

```
- (void)keyboardShown  
{  
    NSLog(@"keyboardShown");  
}
```

# MVC

Controller

```
graph TD; Controller[Controller] --> View[View]; Controller --> Model[Model]; View --> Model;
```

View

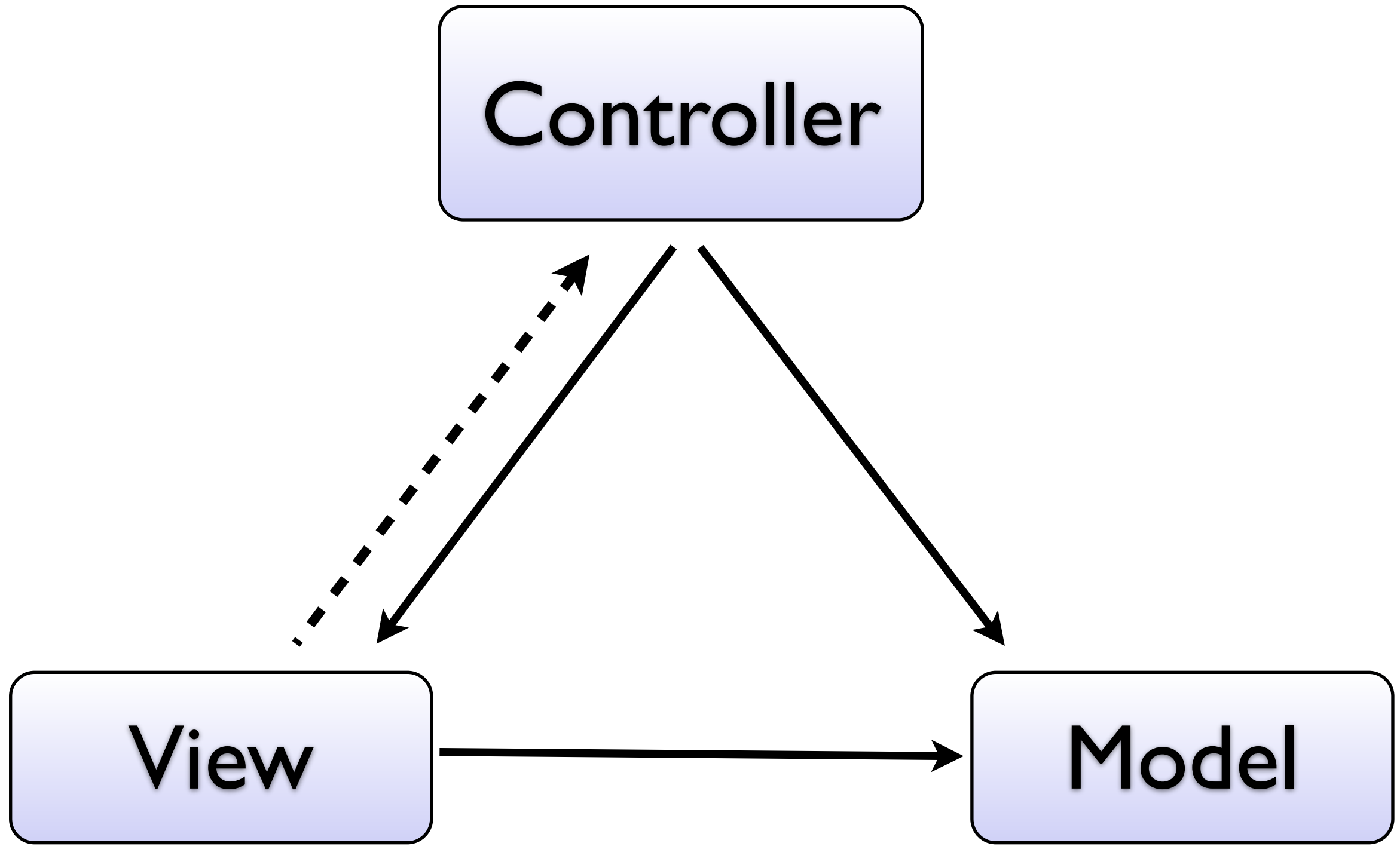
Model

# MVC

Controller

View

Model





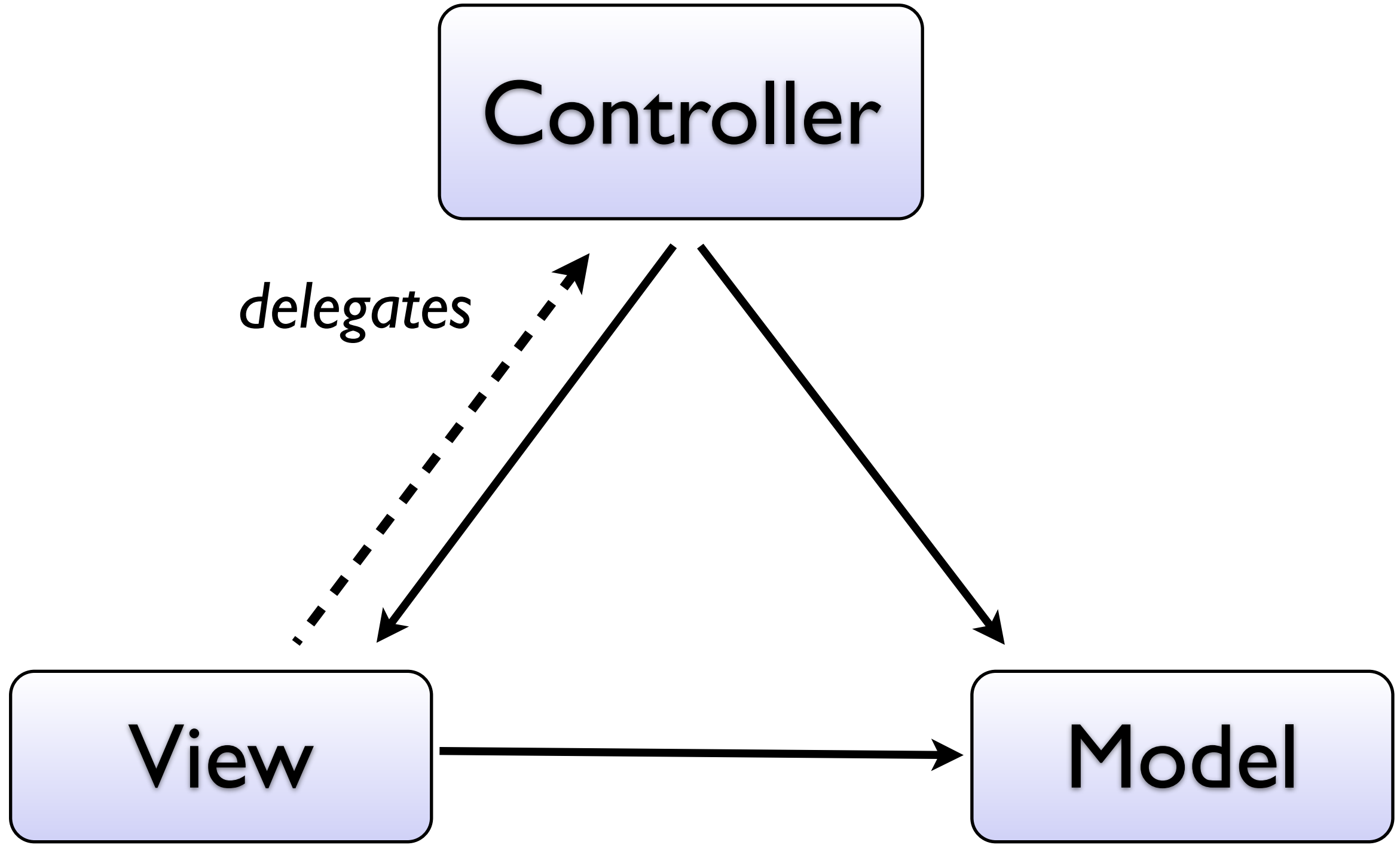
# MVC

Controller

*delegates*

View

Model



# MVC

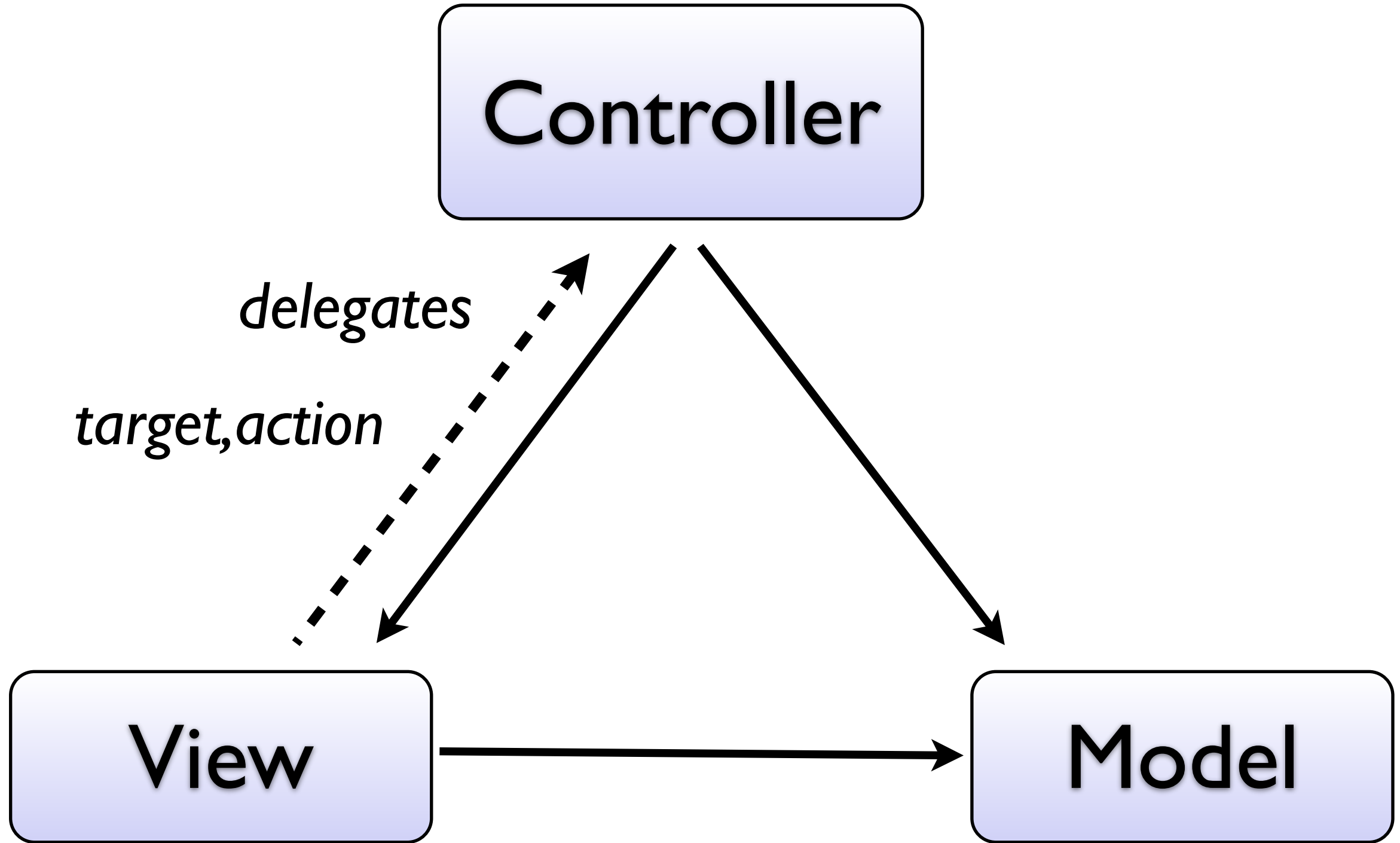
Controller

*delegates*

*target, action*

View

Model



# MVC

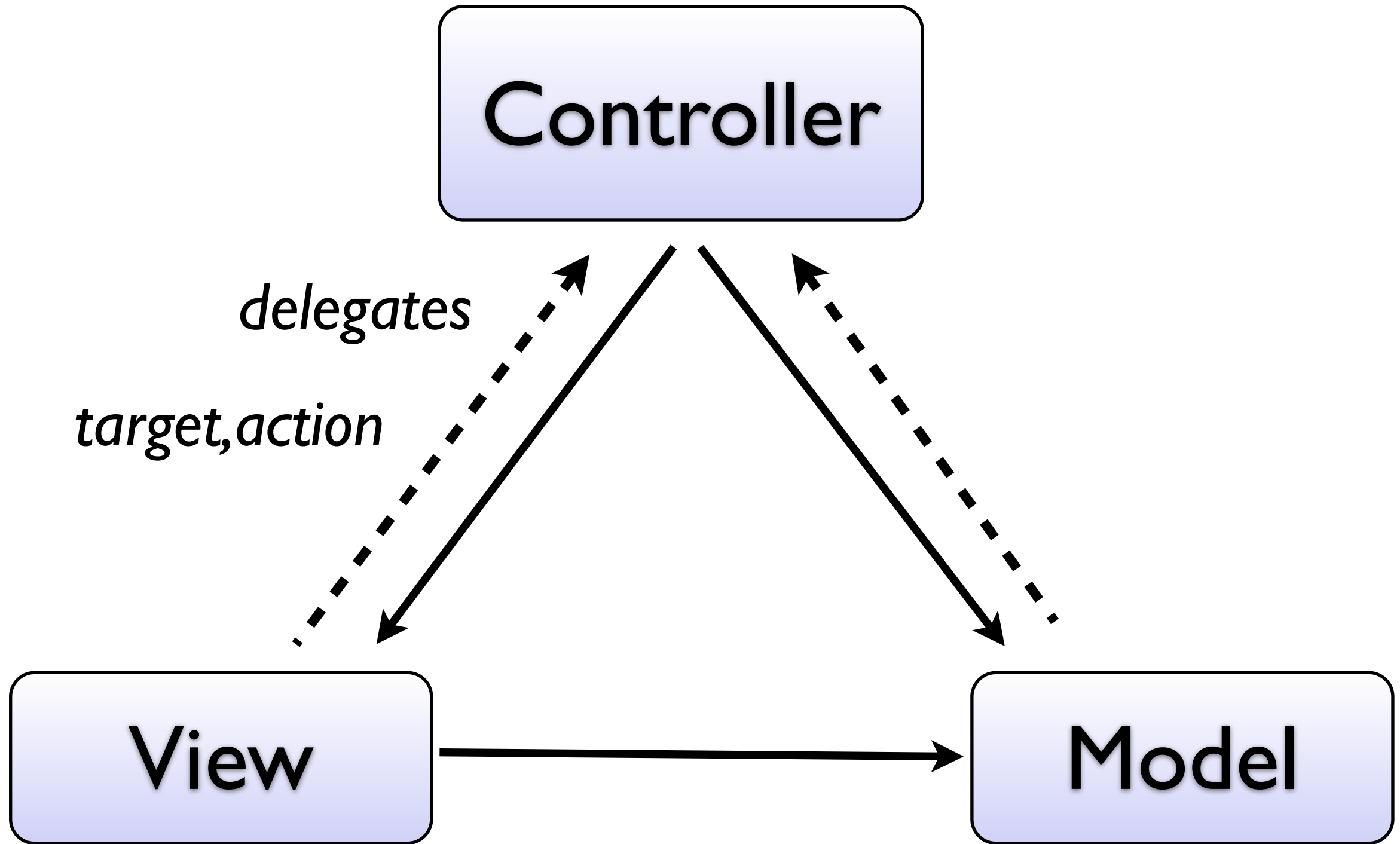
Controller

*delegates*

*target, action*

View

Model



# MVC

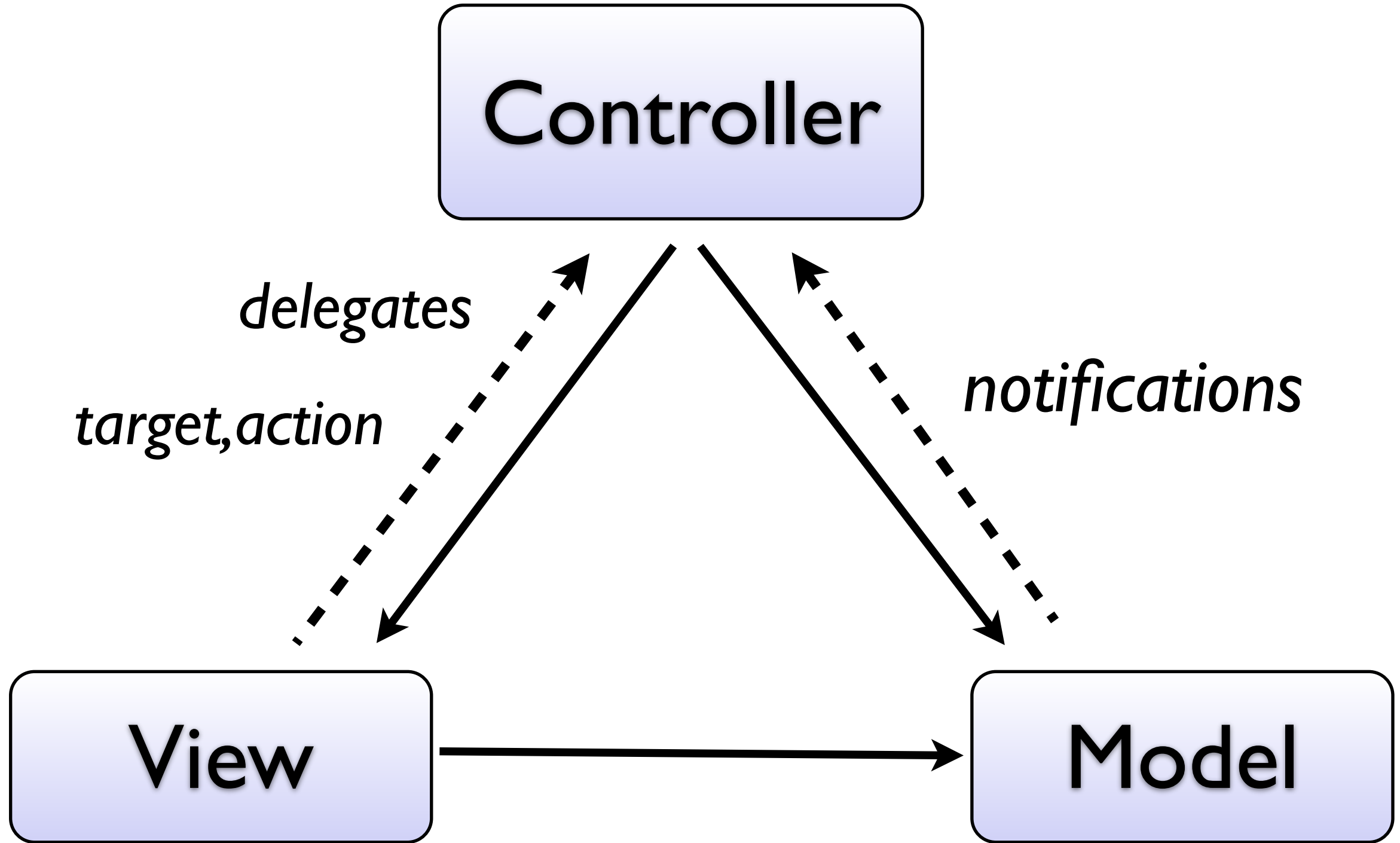
Controller

*delegates*  
*target, action*

*notifications*

View

Model



# Демонстрация

- как сохранять ссылки на views в контроллере
- как обрабатывать нажатия UIButton
- как работать с UITextField
- как отобразить картинку
- примеры других контролов

# Домашнее задание

- продумать из каких экранов будет состоять ваше приложение
- выбрать один из простых экранов, где ТОЛЬКО надписи, картинки, поля ввода
- реализовать его
- спросить у меня, если непонятно, как что-то из этого экрана реализовать

Всё!

*Глеб Тарасов*

*[gleb34@gmail.com](mailto:gleb34@gmail.com)*

*[twitter.com/pilot34](https://twitter.com/pilot34)*