

Разработка iOS приложений

Лекция P.S. 1.

P.S.

Singleton

Гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

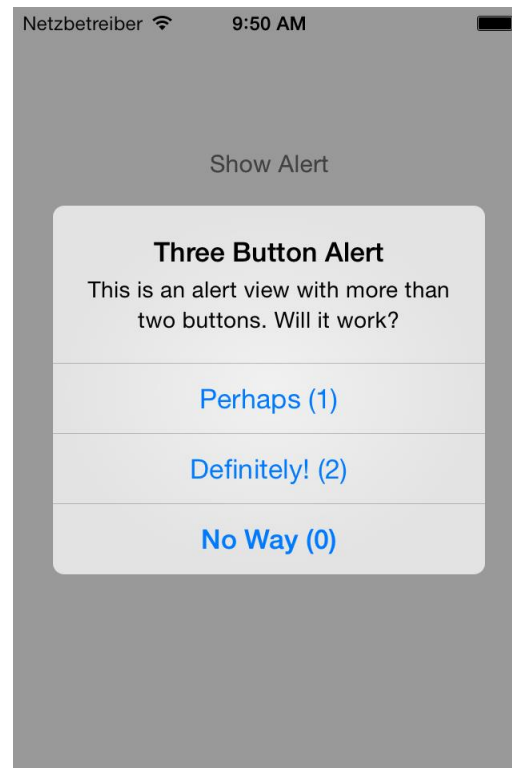
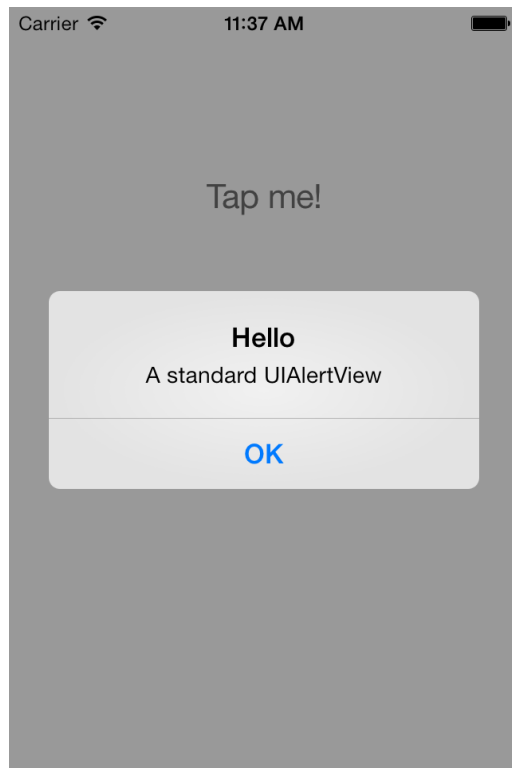
```
+( ClassName* ) instance
{
    static ClassName* sharedInstance = nil;

    static dispatch_once_t predicate;
    dispatch_once( &predicate, ^
    {
        sharedInstance = [ [ ClassName alloc ] init ];
    } );

    return sharedInstance;
}
```

Полезные вещи. UIAlertView

Глубокоуважаемый MessageBox



Полезные вещи. UIAlertView

Показать простой UIAlertView

```
UIAlertView* message = [ [ UIAlertView alloc ]
    initWithTitle: @"Hello World!"
    message: @"This is your first UIAlertView message."
    delegate: nil
    cancelButtonTitle: @"OK"
    otherButtonTitles: nil ] autorelease ];

[ message show ];
```

Полезные вещи. UIAlertView

Добавим кнопки ...

... при инициализации

```
UIAlertView* message = [ [ UIAlertView alloc ]
    initWithTitle: @"Hello World!"
    message: @"This is your first UIAlertView message."
    delegate: nil
    cancelButtonTitle: @"Button 1"
    otherButtonTitles: @"Button 2", @"Button 3", nil
] autorelease ];
[ message show ];
```

Полезные вещи. UIAlertView

Добавим кнопки ...

... после инициализации

```
UIAlertView* message = [ [ UIAlertView alloc ]
    initWithTitle: @"Hello World!"
    message: @"This is your first UIAlertView message."
    delegate: nil
    cancelButtonTitle: @"Button 1"
    otherButtonTitles: nil ] autorelease ];

[ message addButtonWithTitle: @"Button 2" ];
[ message addButtonWithTitle: @"Button 3" ];

[ message show ];
```

Полезные вещи. UIAlertView

Обработка нажатий кнопок:

Обозначим протокол

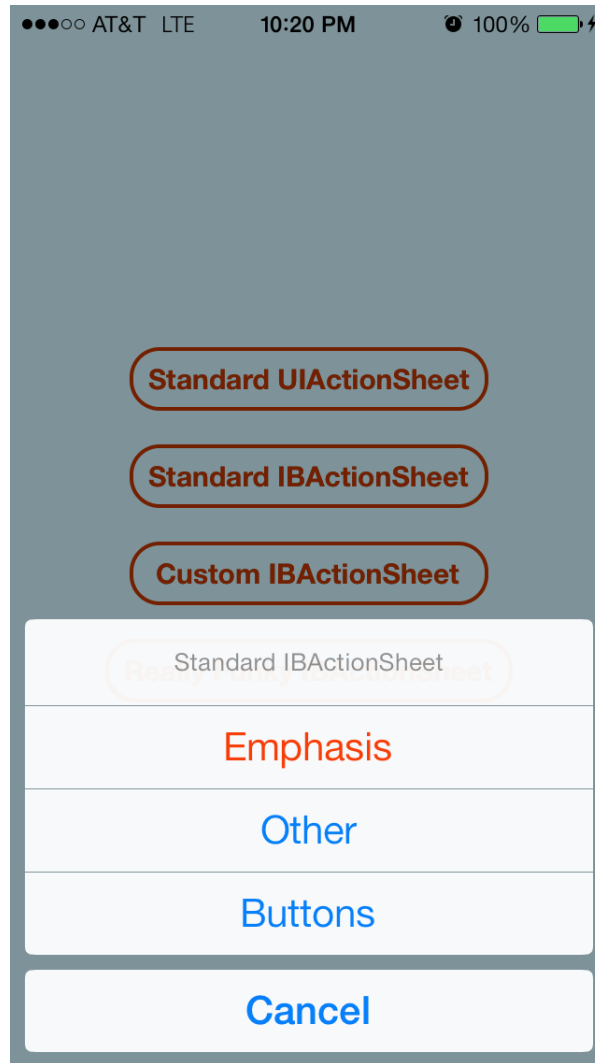
```
@interface ViewController :  
    UIViewController< UIAlertViewDelegate >
```

И реализуем функцию

```
- ( void ) alertView: ( UIAlertView* )alertView  
clickedButtonAtIndex: ( NSInteger )buttonIndex  
{  
    switch ( buttonIndex )  
    {  
        case 1: NSLog( @"Button 1 was selected." ); break;  
        case 2: NSLog( @"Button 2 was selected." ); break;  
        case 3: NSLog( @"Button 3 was selected." ); break;  
    }  
}
```

Полезные вещи. UIAlertController

Всплывающее меню



Полезные вещи. UIAlertController

Как показать такой как на картинке Action Sheet, очень просто

```
UIAlertSheet* popupQuery = [ [ UIAlertController alloc ]
    initWithTitle: @"Title"
    delegate: self
    cancelButtonTitle: @"Cancel"
    destructiveButtonTitle: @"Home"
    otherButtonTitles: @"Office", @"Car", nil
] autorelease ];

popupQuery.actionSheetStyle = UIAlertControllerStyleBlackOpaque;
[ popupQuery showInView: self.view ];
[ popupQuery release ];
```

Полезные вещи. UIAlertController

Обрабатываем нажатия

Обозначаем протокол

```
@interface ViewController :  
    UIViewController< UIAlertControllerDelegate >
```

И реализуем функцию

```
-( void ) actionSheet: ( UIAlertController* )actionSheet  
    clickedButtonAtIndex: ( NSInteger )buttonIndex  
{  
    switch ( buttonIndex )  
    {  
        case 0: break; // "Home" ( Destructive ) Button Clicked  
        case 1: break; // "Office" Button Clicked  
        case 2: break; // "Car" Button Clicked  
        case 3: break; // "Cancel" Button Clicked  
    }  
}
```

Полезные вещи. UIDatePicker

Если Вам надо выбрать дату или время

Cancel	Add Event	Done																					
Room 17																							
All-day	<input checked="" type="checkbox"/>																						
Starts	Fri, Sep 27, 2013																						
<table><tr><td>June</td><td>24</td><td>2010</td></tr><tr><td>July</td><td>25</td><td>2011</td></tr><tr><td>August</td><td>26</td><td>2012</td></tr><tr><td>September</td><td>27</td><td>2013</td></tr><tr><td>October</td><td>28</td><td>2014</td></tr><tr><td>November</td><td>29</td><td>2015</td></tr><tr><td>December</td><td>30</td><td>2016</td></tr></table>			June	24	2010	July	25	2011	August	26	2012	September	27	2013	October	28	2014	November	29	2015	December	30	2016
June	24	2010																					
July	25	2011																					
August	26	2012																					
September	27	2013																					
October	28	2014																					
November	29	2015																					
December	30	2016																					
Ends	Fri, Sep 27, 2013																						
Repeat	Never >																						

Полезные вещи. UIDatePicker

Показываем эту штуку

```
UIDatePicker* datePicker = [ [ [ UIDatePicker alloc ]
    initWithFrame: CGRectMake( 0, 250, 325, 300 ) ] autorelease ];

datePicker.datePickerMode = UIDatePickerModeDate;
datePicker.hidden = NO;
datePicker.date = [ NSDate date ];

[ datePicker addTarget: self
    action: @selector( dateChange: )
    forControlEvents: UIControlEventValueChanged ];

[ self.view addSubview: datePicker ];
```

Полезные вещи. UIDatePicker

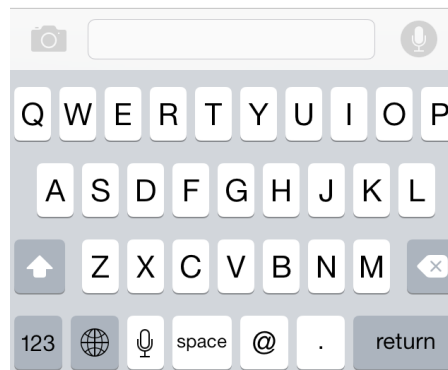
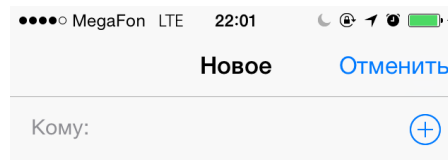
И Обрабатываем получение данных

```
-( void ) dateChange: ( id )sender
{
    NSDateFormatter* df = [ [ [ NSDateFormatter alloc ] init ]
autorelease ];

    df.dateStyle = NSDateFormatterMediumStyle;
    datelabel.text = [ NSString stringWithFormat: @"%@ ",
        [ df stringFromDate: datePicker.date ] ];
}
```

Спец. контроллеры. MFMessageComposeController

Иногда вам надо отправить SMS. Apple не позволяет отправлять SMS в тайне от пользователя, поэтому единственный способ – это MFMessageComposeViewController



MFMessageComposeViewController

Показываем контроллер

```
MFMessageComposeViewController* ctrl = [ [ [
MFMessageComposeViewController alloc ] init ] autorelease ];

ctrl.messageComposeDelegate = self;

ctrl.recipients = [ NSArray arrayWithObject: @"+7922922222" ];
ctrl.body = @"SMS body";

[ self presentViewController: ctrl animated: YES ];
```

MFMessageComposeViewController

И обрабатываем результат. Заявляем протокол ...

```
@interface ViewController : UIViewController <
MFMessageComposeViewControllerDelegate >
```

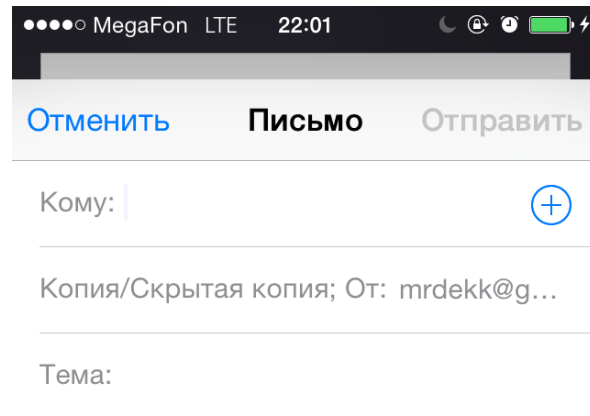
... и обрабатываем событие в селекторе:

```
-( void )messageComposeViewController: (
MFMessageComposeViewController* )controller
didFinishWithResult: ( MFMessageComposeResult )result
{
    switch ( result )
    {
        case MFMessageComposeResultCancelled: break;
        case MFMessageComposeResultSent: break;
        case MFMessageComposeResultFailed: break;
        default: break;
    }

    [ self dismissModalViewControllerAnimated: YES ];
}
```


Спец. контроллеры. MFMailComposeController

Аналогичная ситуация с отправкой почты - MFMailComposeViewController



MFMailComposeViewController

Показываем контроллер

```
MFMailComposeViewController* ctrl = [ [ MFMailComposeViewController alloc ] init ] autorelease ];

ctrl.mailComposeDelegate = self;

[ ctrl setSubject: @"Mail Subject" ];
[ ctrl setMessageBody: @"Mail Body" ];
[ ctrl setToRecipients: [ NSArray arrayWithObject:
@"null@example.com" ] ];

[ self presentModalViewController: ctrl animated: YES ];
```

MFMessageComposeViewController

И обрабатываем результат. Заявляем протокол ...

```
@interface ViewController : UIViewController <
MFMailComposeViewControllerDelegate >
```

... и обрабатываем событие в селекторе:

```
-( void )mailComposeViewController: (
MFMailComposeViewController* )controller
didFinishWithResult: ( MFMailComposeResult )result
{
    switch ( result )
    {
        case MFMailComposeResultCancelled: break;
        case MFMailComposeResultSaved: break;
        case MFMailComposeResultSent: break;
        case MFMailComposeResultFailed: break;
        default: break;
    }

    [ self dismissModalViewControllerAnimated: YES ];
}
```

Вопросы, пожелания, предложения?



Моя просьба

Пожалуйста, ответьте на пару вопросов по поводу курса:

<http://www.surveymonkey.com/s/L59NTCC>