

UIView a controllers

Martin Hrubý
Seminář iOS, FIT VUT v Brně

čerpáno z webu objc.io

Co nás zajímá

- **Model-View-Controller**
- **Konstrukce UI**
 - použití Views, tj. konkrétních prvků — konfigurace, delegace.
 - layout — constraints (umístění vůči okolí, velikost).
- **Princip fungování UIView**
 - Zcela pasivní UIView, aktivní — odvozené od UIControl.
- **Použití UIView s UIViewController**

Hierarchie tříd

- NSObject
 - UIResponder — je schopen reagovat na události.
 - UIView
 - UIControl — rozesílá události.
 - UIButton, UISlider, UIDatePicker
 - UIViewController

Vztah View+Controller

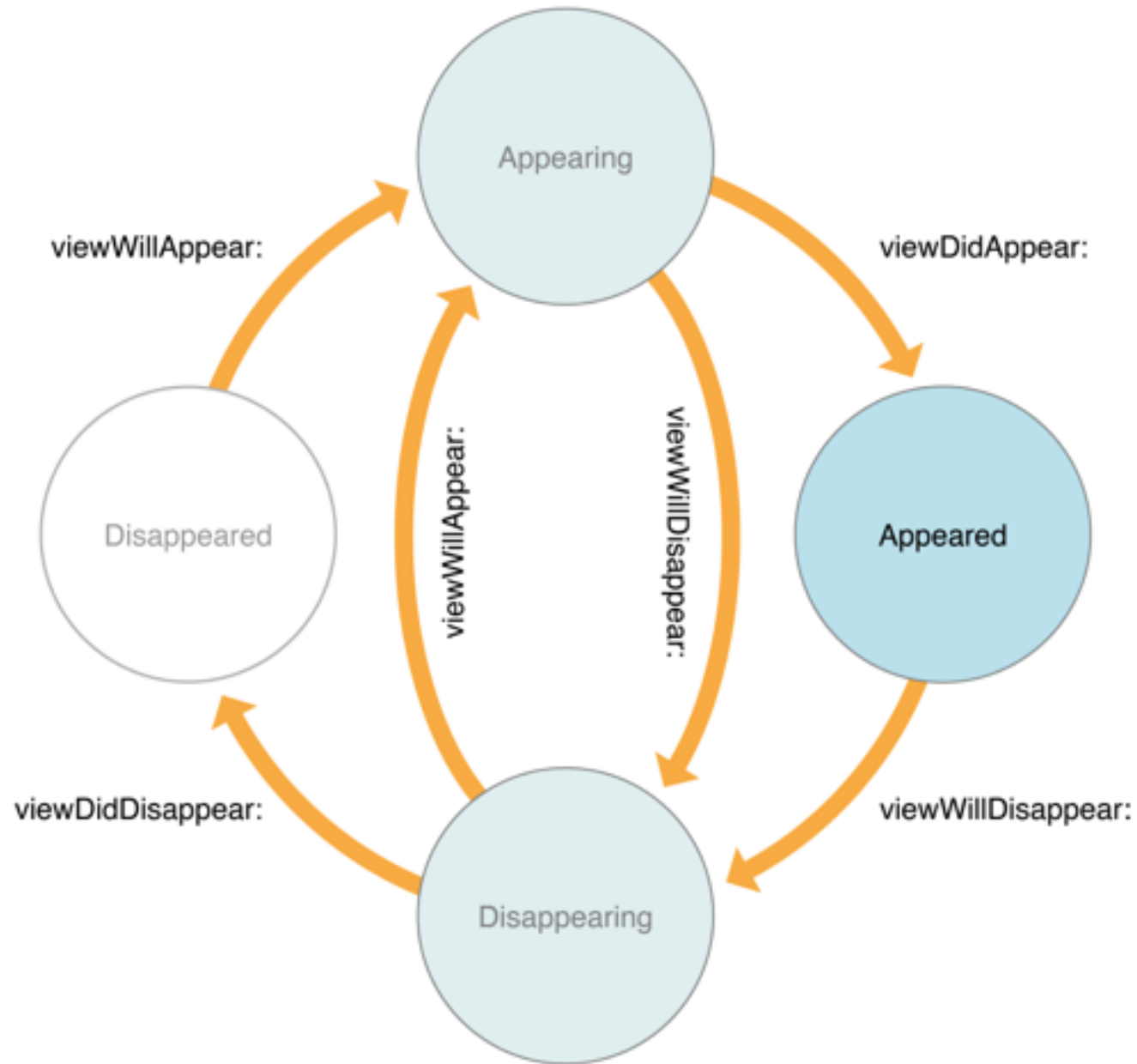
- **UIViewController, property view**
 - majitelem View je Controller, XIB editor
- **Controller modifikuje obsah a layout Views**
 - je to UIResponder (UIView, UIViewController, UIApplication)
 - firstResponder — adresát událostí bez explicitní adresy.
- **View posílá zprávy Controlleru**
 - delegate a protokol — proběh nějaké uživatelské aktivity.
 - události — diskrétní.

First Responder

- Objekty odvozené od UIResponder
- Responder chain — cesta UIResponderů od firstResponder.
- becomeFirstResponder, resignFirstResponder
- kdy se stane objekt FR — např. počátek editace v UITextField

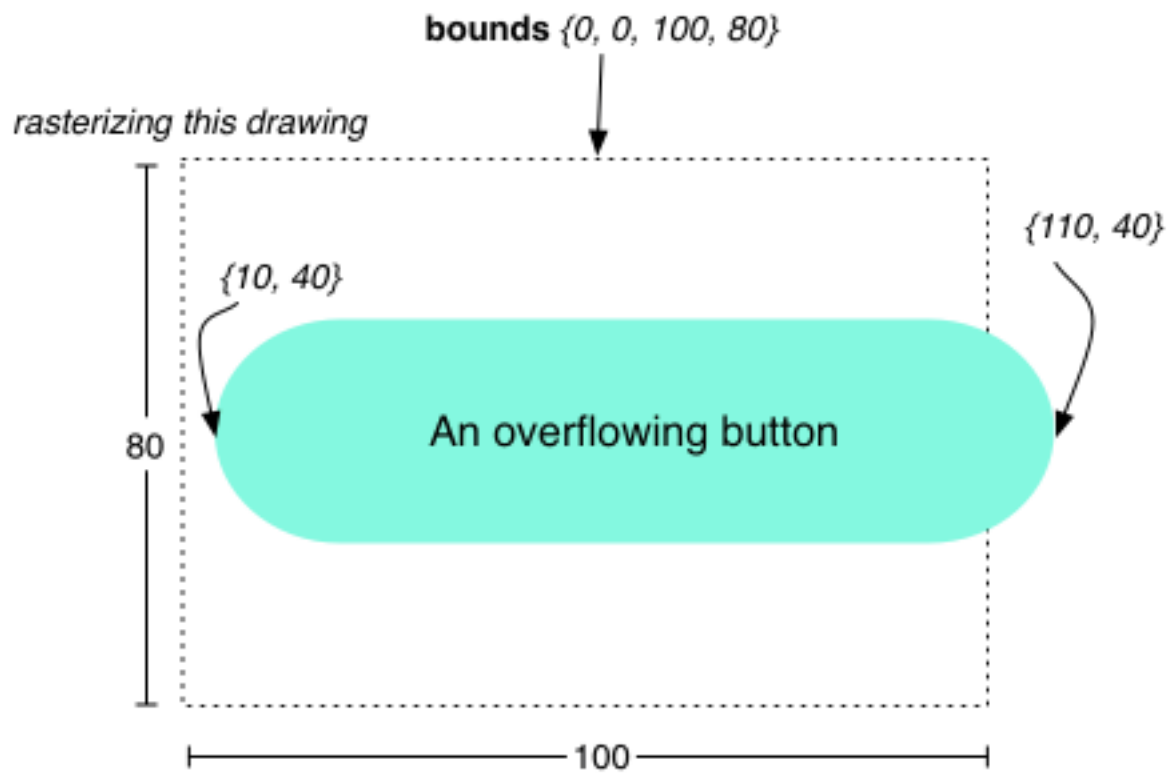
```
- (BOOL)textFieldShouldReturn:(UITextField *)textField {  
    // do whatever you have to do  
  
    [textField resignFirstResponder];  
    return YES;  
}
```

Stavy View+C

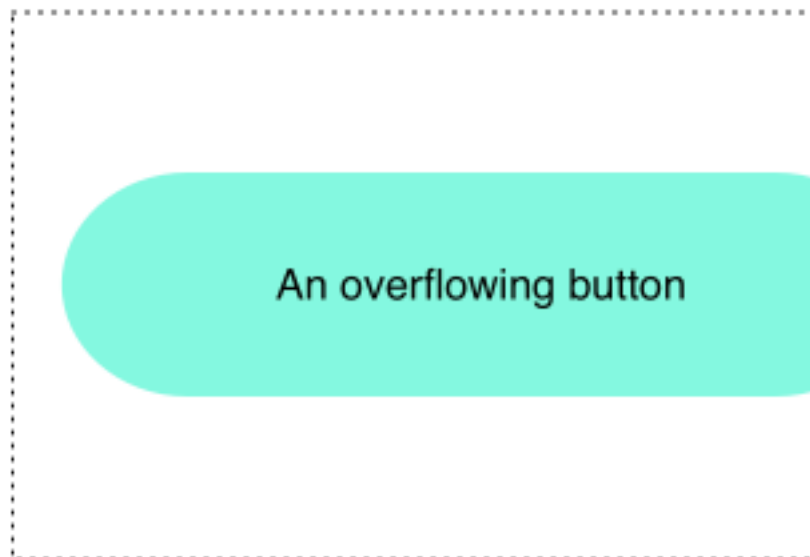


Strom Views

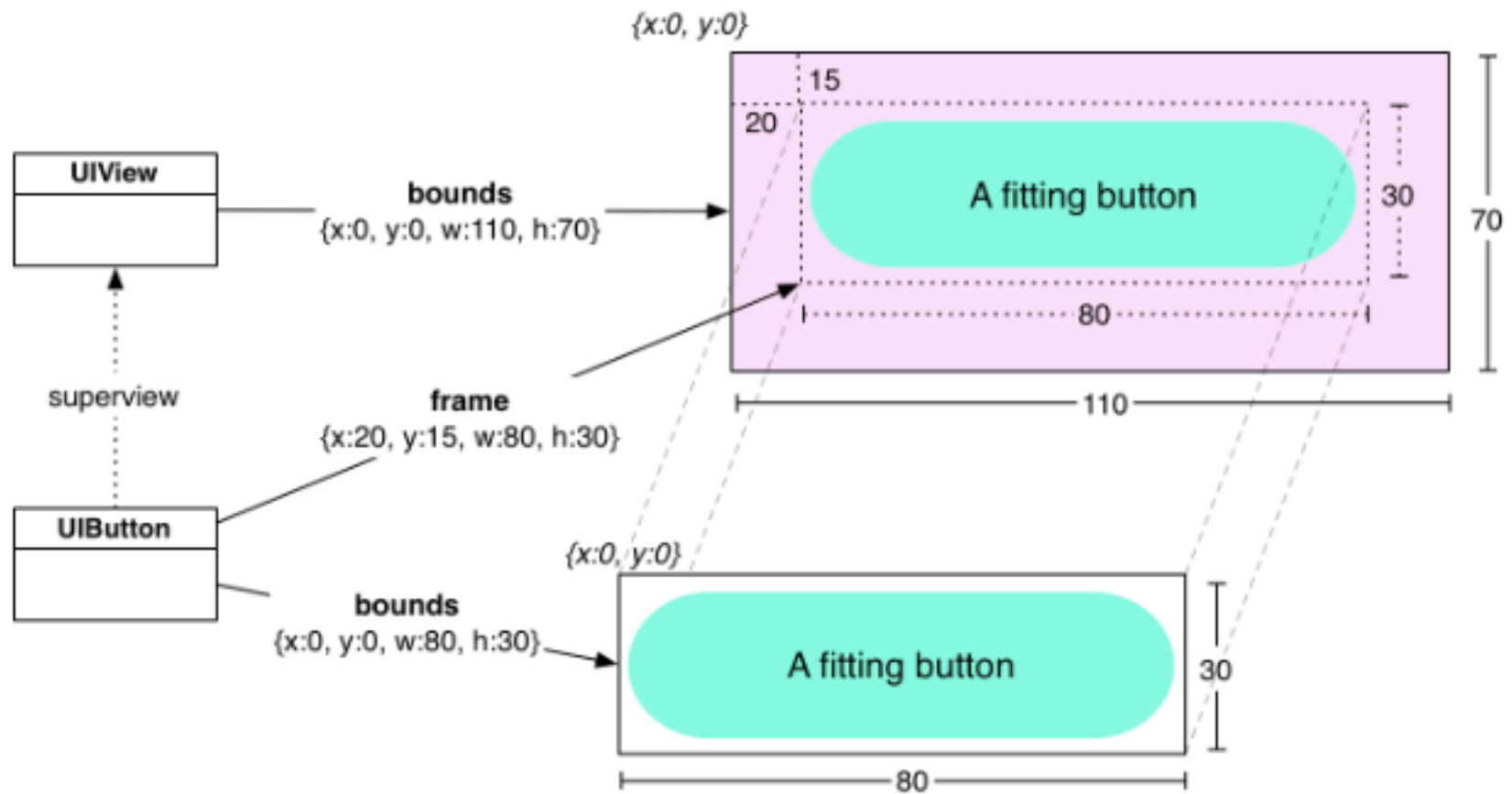
- Topologie view:
 - superview — [self removeFromSuperview];
 - subviews — [view addSubview: v2]
- Geometrie view:
 - CGRect — origin, size (float!). Nejsou pixely!
 - bounds — souřadný systém view.
 - frame — kde je view v souřadném systému **svého superview**.



creates this image



Geometrie View a Superview



System kompozice Views

- View na požádání vytvoří bitmapu reprezentující jeho grafický obsah.
- Kompozice: tato bitmapa překryje bitmapu superview umístěna (top-left) na `view.frame.origin-superview.bounds.origin` a oříznuta `view.frame.size`:
 - umístí se na `view.frame.origin`
 - ovšem relativizováno počátkem souř. systému superview

initWithFrame: (CGRect) fr

- nastaví self.frame = fr
- nastaví self.bounds:
 - origin = (0,0)
 - size = fr.size

Vykreslování

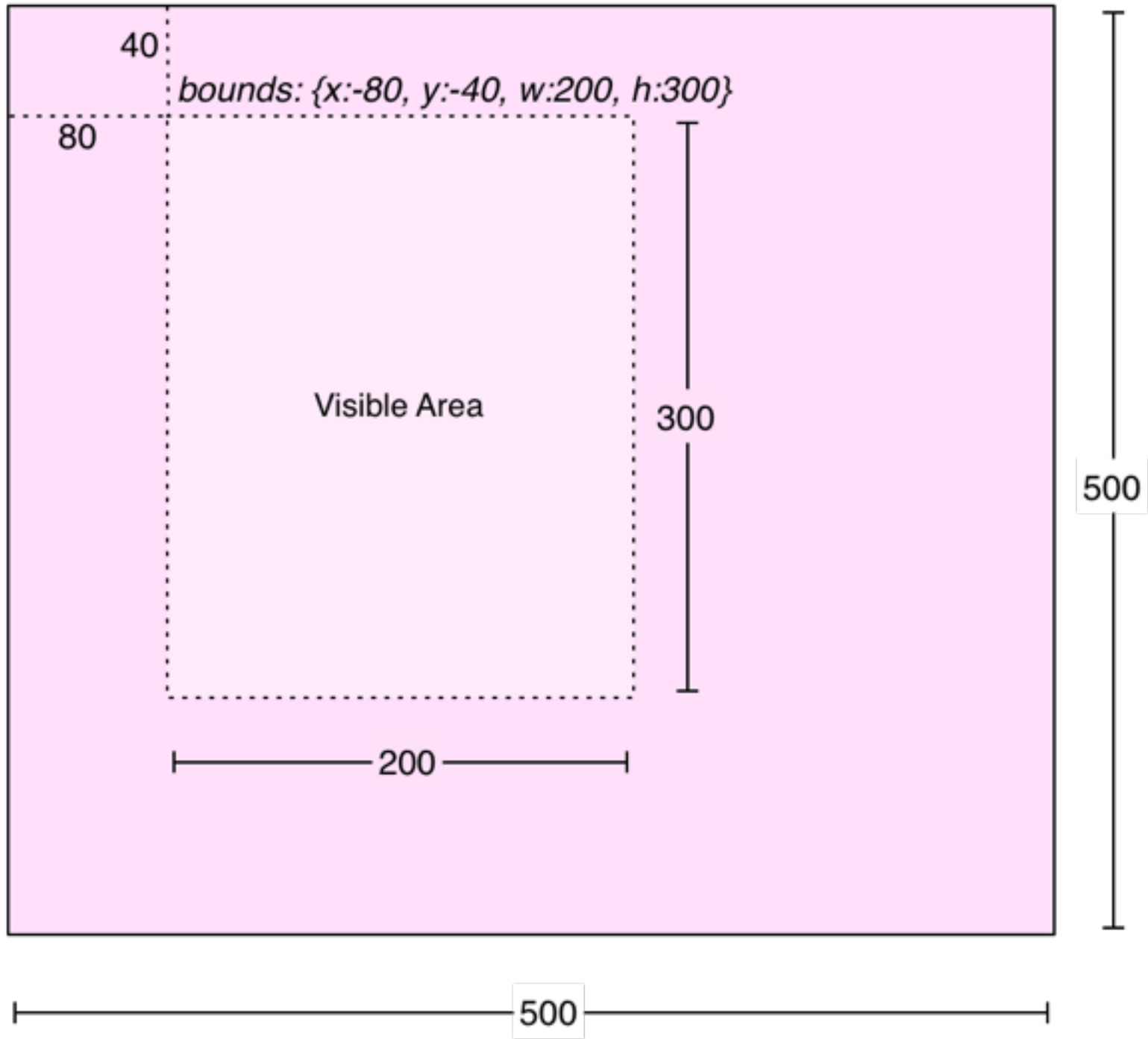
- `-(void) drawRect: (CGRect) inRect`
 - `[view setNeedsDisplay]`
- Vlastní odvozený `UIView`
 - Nepoužívat (minimalizovat) Core graphics — beztak je to pracné.
 - Preferováno budovat view z dostupných `UIView` (label, ...)

UIScrollView

- UITableView, UITextView, UICollectionView.
 - obsahují další view (UITableViewCell). Nemá smysl přesouvat subviews.
 - `contentSize` — je-li menší než `bounds`, není scrolling
 - `contentOffset` — nelze přes (0,0)
- Posunutí zobrazované části je posunutím `bound.origin`

```
- (void)setContentOffset:(CGPoint)offset
{
    CGRect bounds = [self bounds];
    bounds.origin = offset;
    // zřejmě -bounds
    [self setBounds: bounds];
}
```

contentOffset: {x:80, y:40}
contentSize: {w:500, h:500}



Konstrukce ViewControllerů

- Single View.
- Navigation View.
 - push/pop.
- Tab View.
- Obvykle má každý UIView svůj vlastní VC.
 - VC + XIB
 - StoryBoard je zlo.

Dobrá praxe

- Pište ViewControllery + XIB.
- Seskládání do aplikace se může kdykoliv změnit:
 - navigationController
 - tabBarController
 - další

SingleView VC

- Třída MySVC + XIB.
- Nastavení IBOutlet (aspoň view)
 - propojení IBOutlet / IBAction

Vlastní Tab-View

- Více Views a jeden společný VC.
- Základem je SingleView Controller.
- V XIBu přidáme další UIView a ref. IBOutlet.
- Přepnutí Views.

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

    UIBarButtonItem *b1 = [[UIBarButtonItem alloc] initWithTitle: @"SW"
style:UIBarButtonItemStyleDone target: self action:
@selector(switchViews:)];

    self.navigationItem.rightBarButtonItem = b1;

    kt = 0;
    vs = @[v1, v2];

    v1.frame = self.view.frame;
    v2.frame = self.view.frame;
    [v1 setAutoresizingMask: UIViewAutoresizingFlexibleHeight |
UIViewAutoresizingFlexibleWidth];
    [v2 setAutoresizingMask: UIViewAutoresizingFlexibleHeight |
UIViewAutoresizingFlexibleWidth];

    [self.view addSubview: v1];
}
```

Přepnutí Views, animated=NO

```
-(void) switchViews: (id) sender
{
    UIView *oldv = vs[kt];
    kt = (kt == 0) ? 1 : 0;
    UIView *newv = vs[kt];

    [oldv removeFromSuperview];
    [self.view addSubview: newv];
}
```

animated=YES

```
[UIView transitionWithView:self.view
    duration:0.5
    options:UIViewAnimationTransitionFlipFromLeft
    animations:^(
        [oldv removeFromSuperview];
        [self.view addSubview: newv];
    )
    completion:nil
];
```

UITabBarController

- Odvozuje od UIViewController.
 - nepoužívá svůj .view
- Přidává tabBar.
- .viewController
- Delegate: customizeItems, didSelect:
- UIViewController má property tabBarItem.
 - obrázek a text

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame: [[UIScreen
 mainScreen] bounds]];
    self.window.backgroundColor = [UIColor whiteColor];

    LeftVC *vcLeft = [[LeftVC alloc] initWithNibName: @"FV1" bundle:nil];
    UIViewController *vcRight = [[UIViewController alloc]
 initWithNibName: @"FV2" bundle:nil];

    vcLeft.tabBarItem = [[UITabBarItem alloc] initWithTitle: @"Jeden"
 image:nil tag:0];
    vcRight.tabBarItem = [[UITabBarItem alloc] initWithTitle: @"Druhy"
 image:nil tag:0];

    UITabBarController *mainSplit = [[UITabBarController alloc] init];
    mainSplit.viewControllers = @[vcLeft, vcRight];

    self.window.rootViewController = mainSplit;
    [self.window makeKeyAndVisible];

    return YES;
}
```

UINavigationController

- Přinejmenším dodává vrchní bar pro tlačítka.
- Inicializuje se s rootControllerem.
 - každý VC má property navigationController
 - topViewController — aktuální top VC
 - pushViewController:animated:
 - varianty popVC, ...

iPad: Master-Detail VC

- Typický koncept pro iPad. Na iPhone lze převést na jeden UINavigationController + další VC (koncepte VC)
- 5 VC:
 - UISplitViewController — podobně jako TabBarController.
 - 2x
 - Navigation + VC

Master-Detail polohy

- Typická aplikace — systémová Mail.app
- Landscape — běžný Master+Detail.
- Landscape — Detail ve full Screen.
- Portrait — Detail, Master (potlačen, collapsed)
- Ovládá se rotací přístroje, tlačítka v NavBaru.
 - splitVC má delegate.

SplitVC konstrukce

```
LeftVC *vcLeft = [[LeftVC alloc] initWithNibName:@"FV1" bundle:nil];  
UIViewController *vcRight = [[UIViewController alloc] initWithNibName:  
@"FV2" bundle:nil];
```

```
UINavigationController *navLeft = [[UINavigationController alloc]  
initWithRootViewController: vcLeft];  
UINavigationController *navRight = [[UINavigationController alloc]  
initWithRootViewController: vcRight];
```

```
UISplitViewController *mainSplit = [[UISplitViewController alloc] init];  
mainSplit.viewControllers = @[navLeft, navRight];  
mainSplit.delegate = self;
```

```
navRight.topViewController.navigationItem.leftBarButtonItem =  
mainSplit.displayModeButtonItem;
```

...

Smysl Master-Detail

- Master — seznam datových záznamů ve zjednodušené podobě (vlastní UITableViewCell)
- Detail — editace záznamu, širší detail záznamu.
- Typicky je Master single-selection tabulka.
 - Pokud není v tabulce řádek označen, Detail by měl zobrazovat nějaký placeholder.
 - property detailObject, specifikace setteru

```
-(void) setDetailObject:(id)detailObject
{
    if (_detailObject != nil && detailObject == nil) {
        [detailView removeFromSuperview];
        [self.view addSubview: placeholderView];
    }

    if (_detailObject == nil && detailObject != nil) {
        [placeholderView removeFromSuperview];
        [self.view addSubview: detailView];
    }

    _detailObject = detailObject;
    if (_detailObject != nil) {
        // konfiguruj view (typicky tableView)
        [tableView reloadData];
    }
}

-(void) viewDidLoad
{
    [super viewDidLoad];
    [self.view addSubview: placeholderView];
}
```

iPad: UIPopoverController

- UIPopoverController — (není VC) nad zadaným VC.
- Inicializuje se zadaným VC (obsahuje nějaký View)
 - například i UINavigationController + root
 - ten by měl posílat zprávy mateřskému VC

Simple demo

```
-(UIPopoverController *) pop
{
    if (_pop == nil) {
        UIDatePicker *datePicker = [[UIDatePicker alloc] initWithFrame:
CGRectMake(0, 0, 200, 200)];
        datePicker.datePickerMode = UIDatePickerModeDateAndTime;

        UIViewController *vc = [[UIViewController alloc] init];
        vc.view = datePicker;

        _pop = [[UIPopoverController alloc]
initWithContentViewController: vc];
    }

    return _pop;
}

-(void) bbAkcce:(id)sender
{
    [self.pop presentPopoverFromBarButtonItem:popButton
permittedArrowDirections:UIPopoverArrowDirectionAny animated: YES];
}
```

Závěr

- Pište ViewControllery.
 - Nastavení IBOutletu je setValue:forKey.
 - VC může mít i různé XIB (varianta iPad, iPhone).
 - Má být reference na IBOutlet strong nebo weak?
 - je-li weak, pak je retainCount==1 !
- StoryBoard je zlo (každý na to jednou přijde).