Konstrukce aplikace v iOS

—

UIKit/Storyboard

IZA, Martin Hrubý, FIT VUT, 2021/22
Doba post-covidová, ovšem válečná… :(

Úvod

- Několik střípků z předchozí technologie:
  - Pro lepší pochopení konceptů SwiftUI.
  - SwiftUI aplikace smí využívat prvky UIKitu.
- Univerzální koncepty iOS aplikací.
- Budeme průběžně srovnávat se SwiftUI.
  - … a hledat kořeny… :)

Architektura aplikace

- UIApplication+delegate — API na jádro iOS.
  - ViewControllers — řízení ”jedné obrazovky” a řízení programu (Storyboard, Segue).
  - Základy Views a geometrie Views.
- M-V-C. M-VC.
  - V-C jsou platformově závislé (iOS/tvOS, macOS, watchOS).
  - Cocoa (AppKit, NSView…). CocoaTouch (UIKit, UIView…).
  - Model — DB, komunikace vně, komunikace uvnitř.

Start aplikace

- Instance *UIApplication, UIApplicationDelegate*.
  - didFinishLaunchingWithOptions — jako main() programu.
- Instanciace UIWindow / UISceneWindow.
  - Počáteční ViewController — rootViewController.
  - UIWindow začne být "*key* and *visible*".
- Startovní výpočty — vedlejší vlákna.
  - Co nejrychlejší start aplikace.
  - Obsah aplikace "dodělávat za běhu".

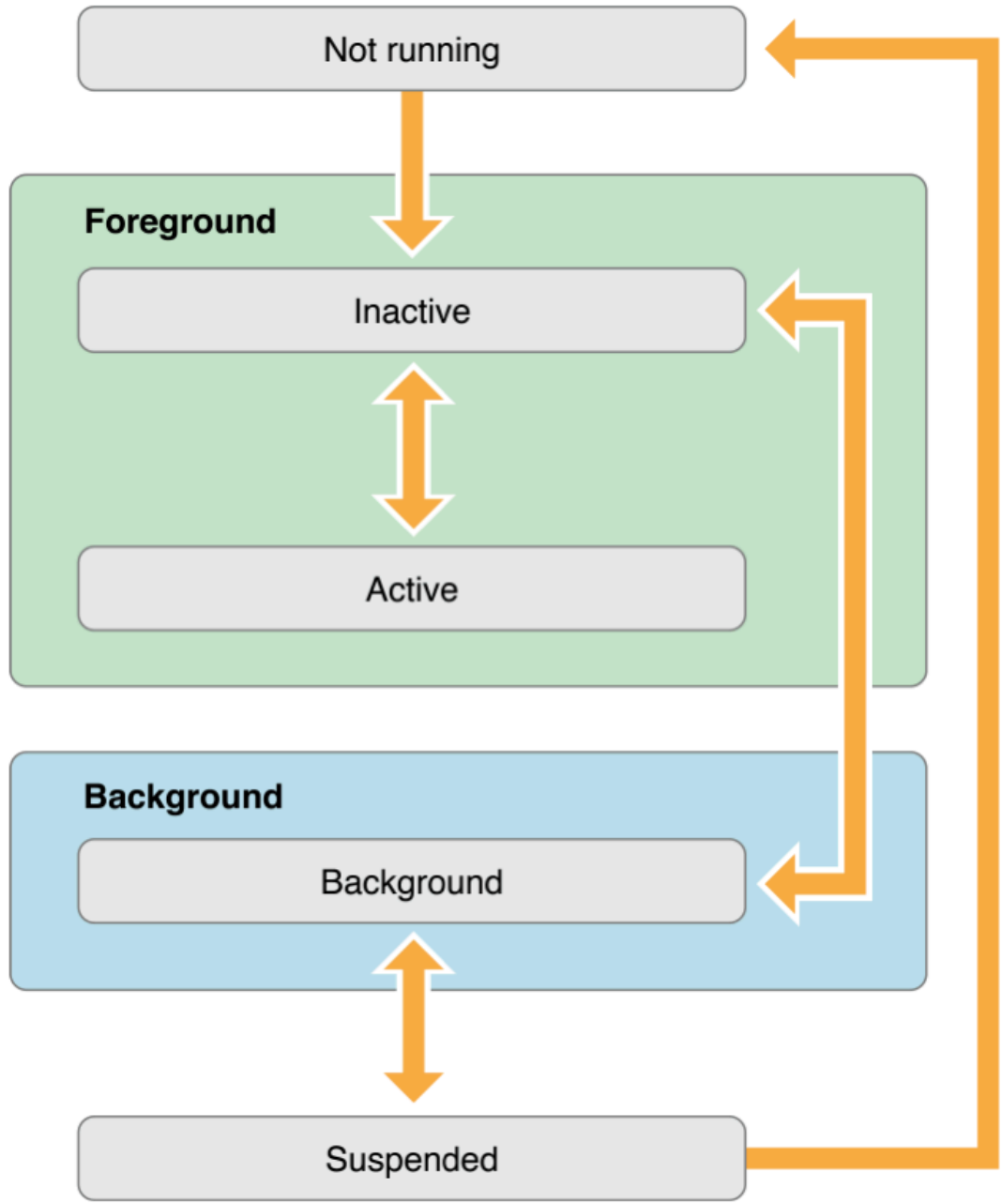
UIApplication

- Knihovní třída. Nepředpokládá se odvozování vlastní.
- Singleton (*UIApplication.shared*).
  - Rozhraní mezi OS a aplikací (události).
  - Ovládá základní podobu aplikace v systému (ikonka, horní lišta apod.).
  - Referencuje: *app-delegáta,* windows a keyWindow.

UIApplication delegate

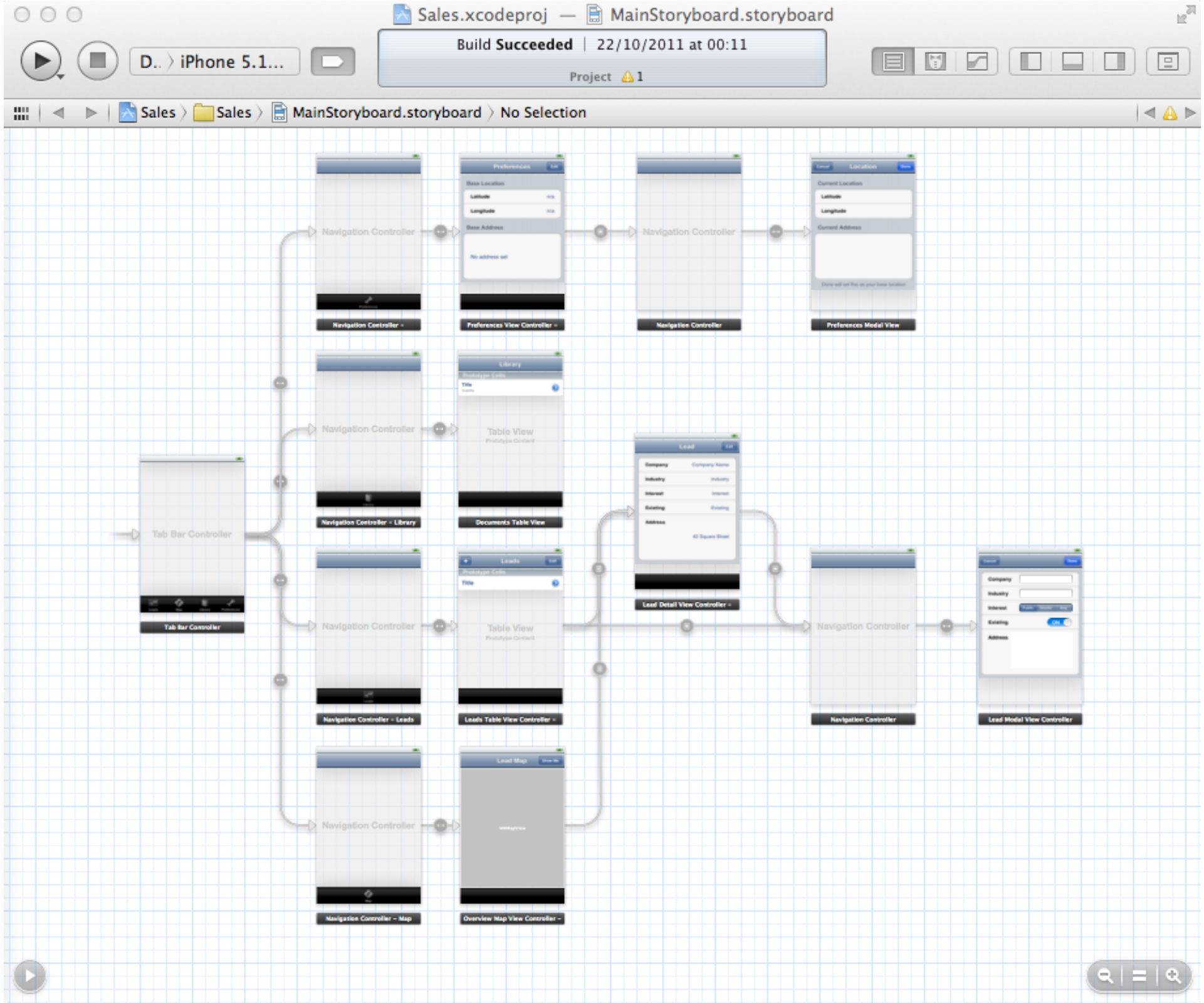
- Singleton. Může referencovat objekty globálního charakteru (typicky data, CoreData, apod.)
  - didFinishLaunchingWithOptions. Neblokovat příliš start aplikace (globální fronta).
  - dřív konstrukce UIWindow, rootVC apod. Dnes StoryBoard.
- Události změny stavu aplikace — popředí/pozadí, will/did,
  - … se přesunuly do UIWindowSceneDelegate

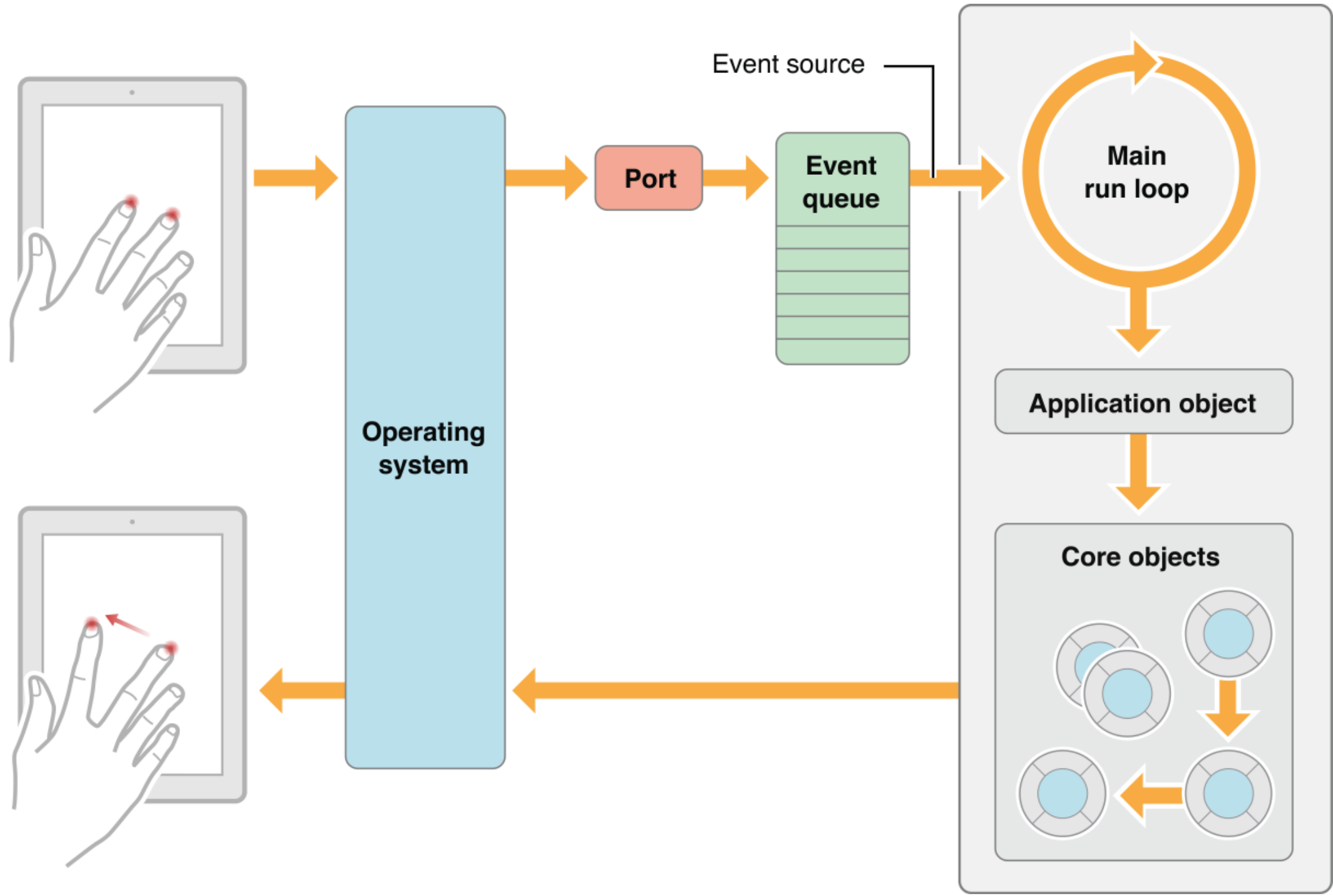
Stavy aplikace / scény



Stavy aplikace/scény

- *Not running* — nespuštěna nebo ukončena.
- Popředí (je viditelná):
  - *Active* — vykonává kód, přijímá události.
  - *Inactive* — nepřijímá události, běží. Typicky krátký moment.
- *Pozadí* (není viditelná): smí vykonávat kód.
- *Suspended* — je v paměti, neběží. Systém smí aplikaci kdykoli ukončit.
  - Aplikace je zodpovědná za uložení dat při přechodu do BG.


Sales.xcodeproj — MainStoryboard.storyboard
Build Succeeded | 22/10/2011 at 00:11
Project 1
D.. iPhone 5.1...
Sales Sales MainStoryboard.storyboard No Selection
Navigation Controller
Preferences
Base Location
Latitude
Longitude
Base Address
No address set
Navigation Controller
Location
Current Location
Latitude
Longitude
Current Address
Navigation Controller
Preferences View Controller
Navigation Controller
Preferences Modal View
Navigation Controller
Library
Table View
Prototype Content
Navigation Controller - Library
Documents Table View
Tab Bar Controller
Tab Bar Controller
Lead
Company
Company Name
Industry
Interest
Existing
Address
42 Square Street
Lead Detail View Controller
Navigation Controller
Navigation Controller
Leads
Table View
Prototype Content
Navigation Controller - Leads
Leads Table View Controller
Navigation Controller
Company
Industry
Interest
Existing
Address
Lead Modal View Controller
Navigation Controller
Load Map
Navigation Controller - Map
Overview Map View Controller


Event source
Main run loop
Port
Event queue
Operating system
Application object
Core objects

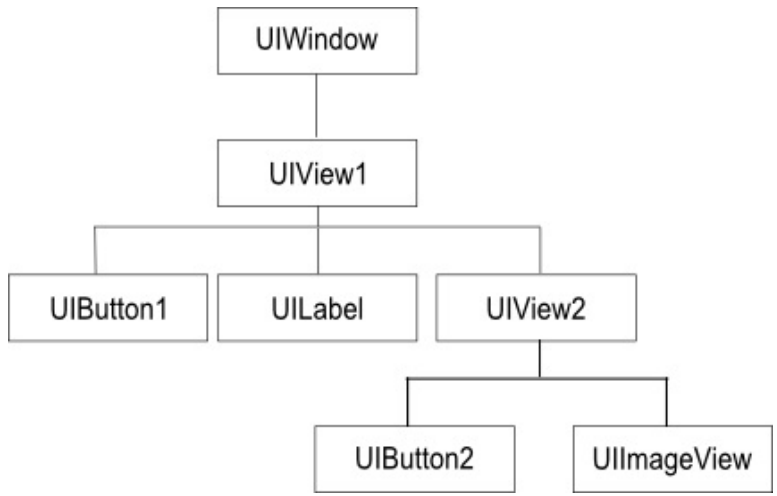
Koncepce Views

- UIView, UILabel, UISwitch, UITableViewCell,
- Superview, subviews — hierarchie.
- Smyslem je skládat UI z knihovních komponent (versus drawRect:). Zobrazovací postscript.
- Provádět změny do UI smí pouze *hlavní vlákno*.

```
class ViewController: UIViewController {

    @IBOutlet var textik : UILabel?

    func inicializuj() {
        textik?.text = "Napis hello";
    }
```

Topologie views

- View má: superview a subviews:
  - superview.addSubView(myView)
  - myView.removeFromSuperview()
- Viditelnost View na obrazovce. Dynamika.
- Předpokládáme stromovou topologii.



Geometrie View

- *Frame* — definuje umístění *view* v jeho *superview*.
- *Bounds* — definuje velikost *view* a jeho lokální souřadný systém.
- *Frame* a *bounds* nemusí mít stejné velikosti.
  - *CGRect*(x: y: width: height:). Origin. Size.
- Souřadnice (0, 0) — levý horní roh, (macOS).
- Pozor: jednotkou souřadnice NENÍ 1 pixel.
  - souřadný systém je veden v číslech typu CGFloat (float).

Vytvoření objektu view

- *var label = UILabel(frame: CGRect(x: 0, y: 0, width: 200, height: 21))*
  - Definuje frame,
  - nastaví *bounds* na *origin*=(0,0), *size*=(200,21).
- view.addSubView(label)

```
public struct CGRect {
    public var origin: CGPoint
    public var size: CGSize
    public init()
    public init(origin: CGPoint, size: CGSize)
```

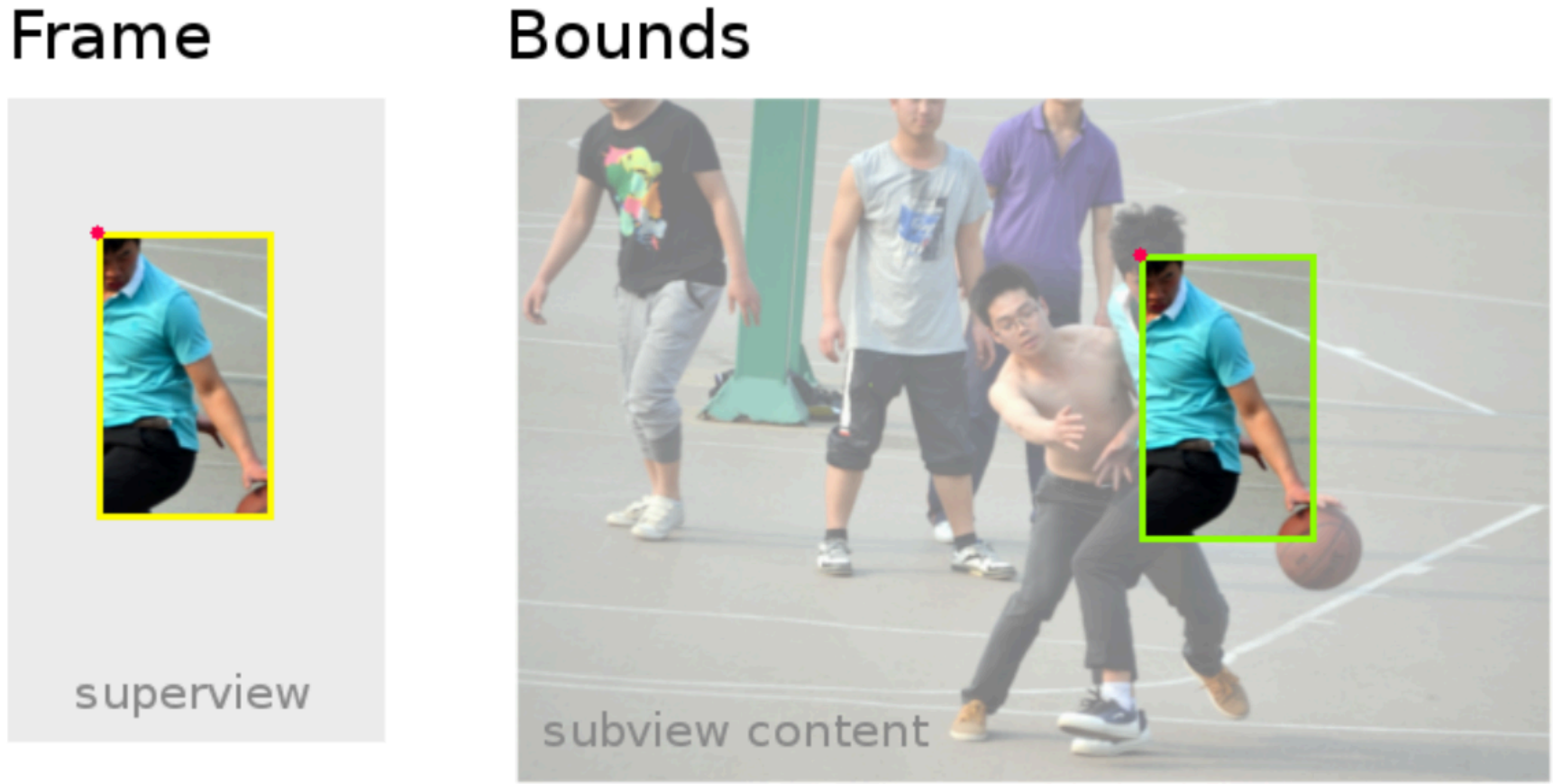
Proces rasterizace a kompozice

- Rasterizace (drawRect:), *bounds*((x,y), (w,h)).
  - Alokuje se prázdná bitmapa velikosti (w,h).
- Představa projekce rasterizace:
  - V abstraktním souřadném systému grafického objektu se objekt (myšleně) vykreslí s počátkem (0, 0).
  - Tento (myšlený) obraz se obtiskne do bitmapy (w, h), kterou přiložíme do bodu (x, y) obrazu.
  - Obecně *pouze část obrazu* se obtiskne do bitmapy (w, h).

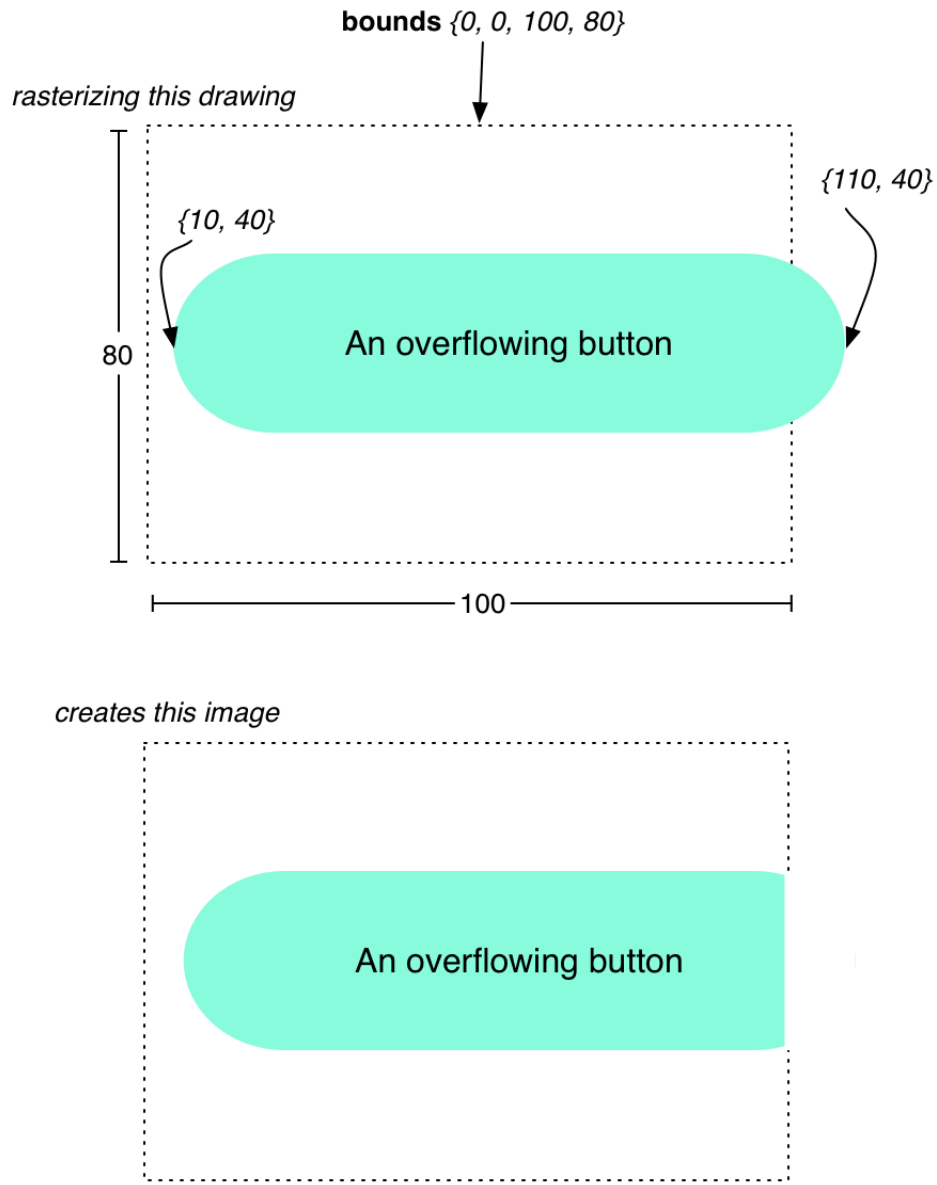
Kompozice

- Mějme superview *S*, *S.frame*, *S.bounds*,
- Představa: kompozice bitmapy *X* subview je pokračování rasterizace do myšleného obrazu.
  - Tj. do myšleného souř. sys. umístím na *X.frame.origin* bitmapu *X* (oříznutou *X.frame.size*).
  - Pro *S* mám bitmapu *S.bounds.size*, kterou obtisknu umístěnou do *S.bounds.origin*.
  - Efektivně *X* jde na *X.frame.origin - S.bounds.origin*.

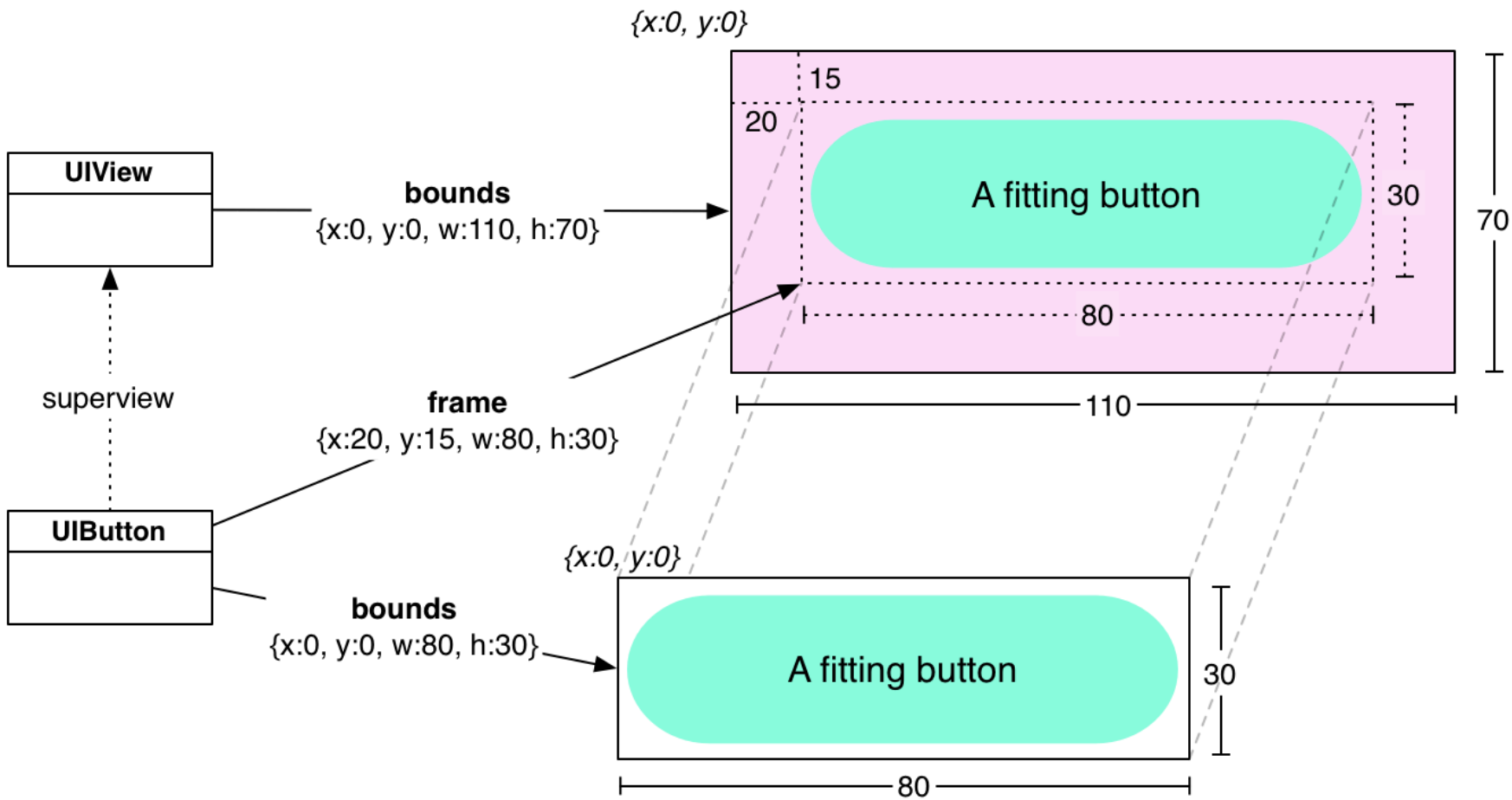
```
Frame
    origin = (40, 60)
    width = 80
    height = 130

Bounds
    origin = (280, 70)
    width = 80
    height = 130
```



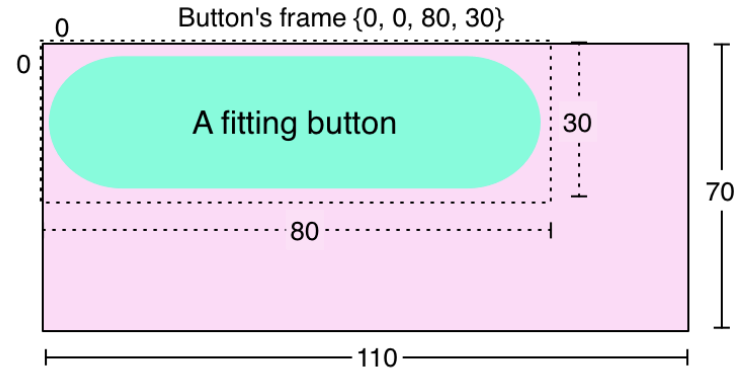
Rasterizace do bounds

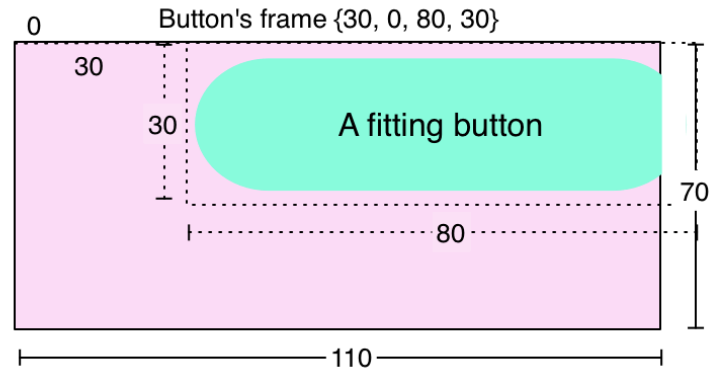


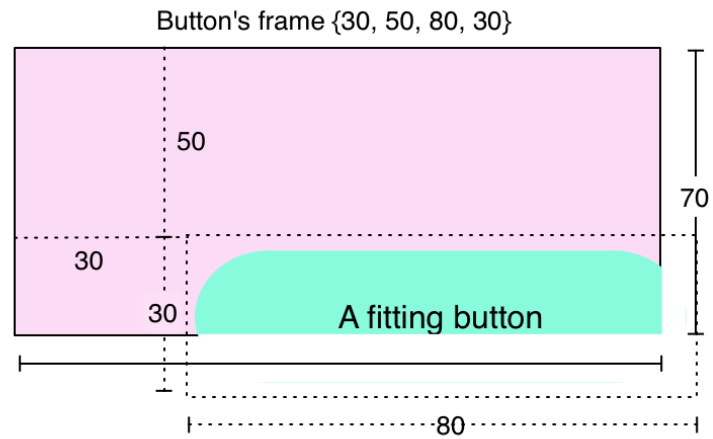
Kompozice view



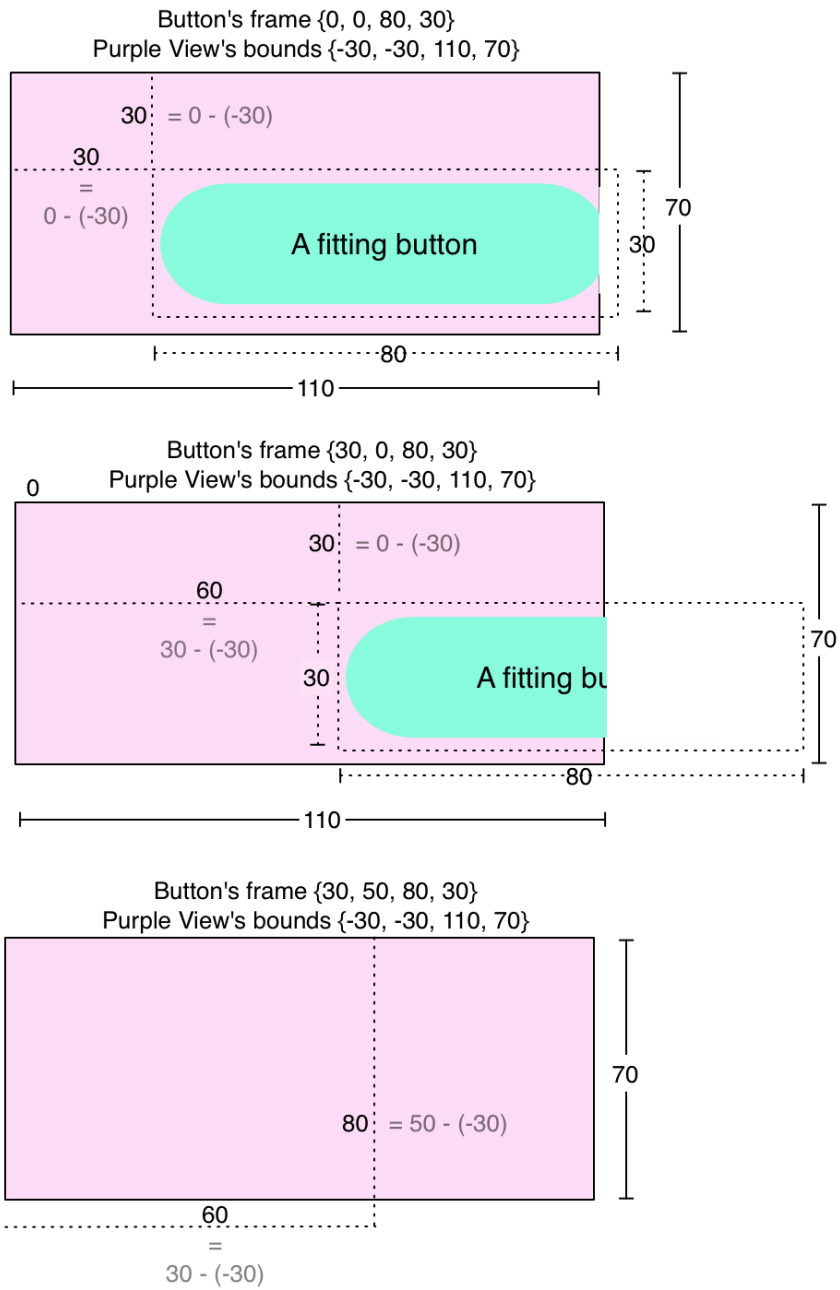
Kompozice view





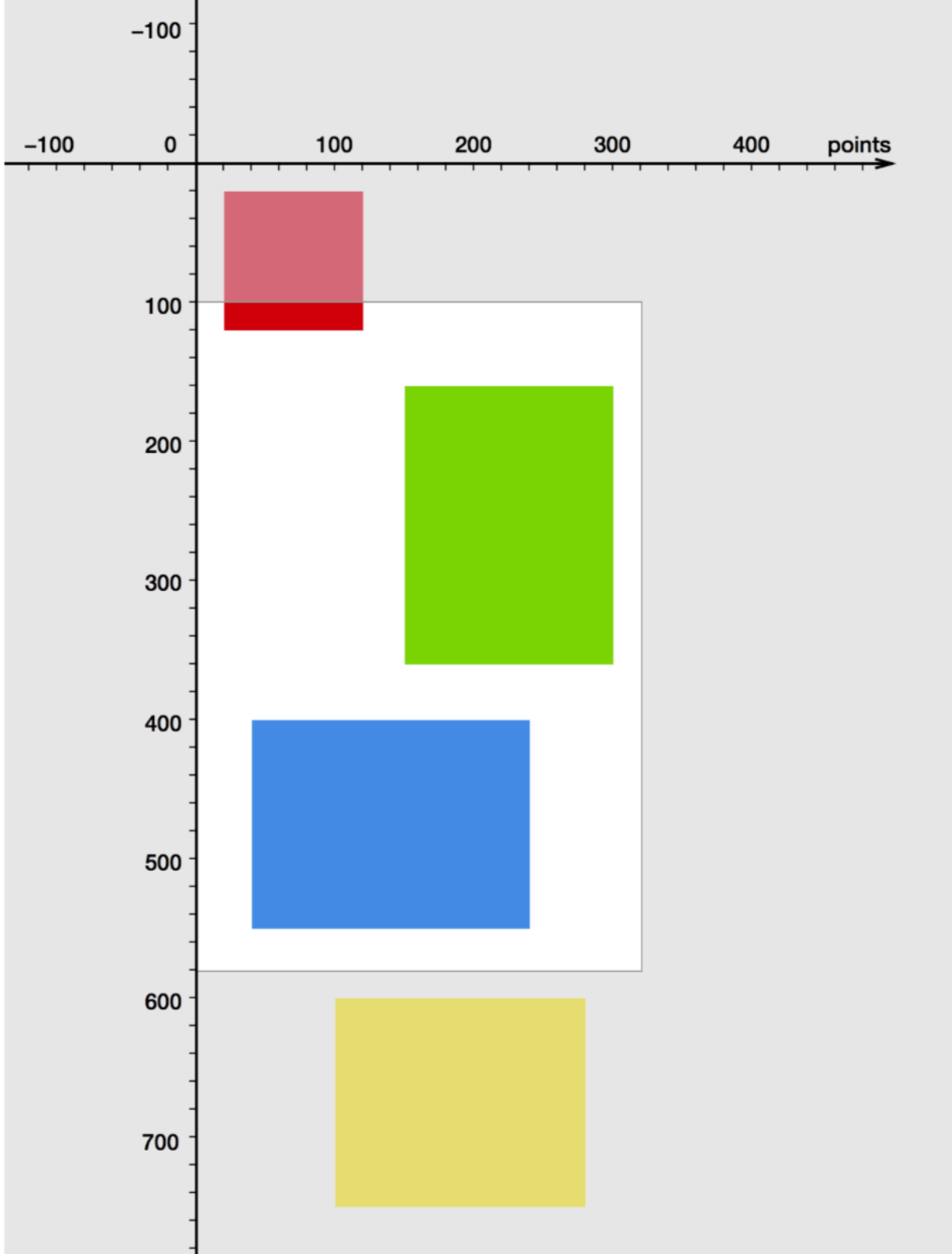


Kompozice, superview bounds



Jak pracuje *UIScrollView*

- Potřebujeme *View* s rozměrem přesahujícím rozměr obrazovky. Typicky *TableView*, *ImageView*, *CollectionView*, …
- ScrollView.ContentView a subviews:
  - různě umístěné
  - scrolling
  - dynamika


−100
−100
0
100
200
300
400
points
100
200
300
400
500
600
700

Komunikace Controller <-> View

- Controller (+View) vyjadřuje stav UI. (jako VM)
- Controller chce *změnit stav* svého view.
  - Tok událostí od Model (Controller) do View.
- View detekuje uživatelskou událost (gesto, klávesnice, rotace), volá controller.
  - Target-action. Objektu X pošli zprávu Y.
  - Typicky směřováno do Controlleru.

Stav objektů View

- Uvažujme *var o: UILabel* (property: text).
- V proceduře kontroleru chceme změnit obsah *UILabelu* (prováděno nějakým vláknem).
  - *o.text = newValue;*
  - *o.setNeedsDisplay()*
- Změny obsahu UI se provádí pouze vláknem zpracovávajícím *MainQueue*!

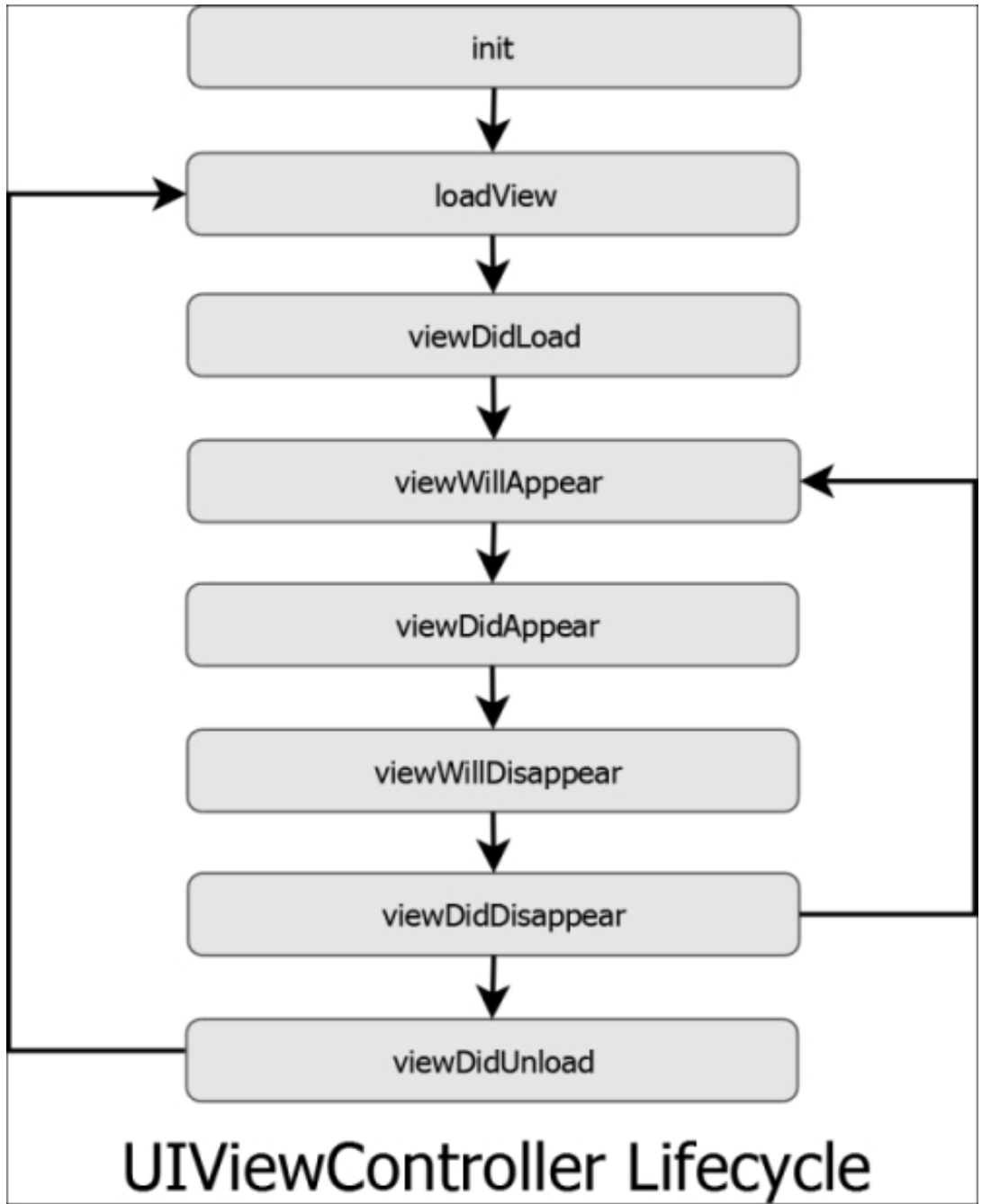
Stav objektů view

- Když uživatel změní stav objektu *view*, *view* posílá zprávu svému ”target” (target-action mechanism). Cílem je typicky controller.
- *UISwitch, UITextField, UIButton, …*
- Typicky:
  - func nejakaZmenaTlacitka(sender: UISwitch);
  - @IBAction
  - přechod na closures.

MVC Stav UI

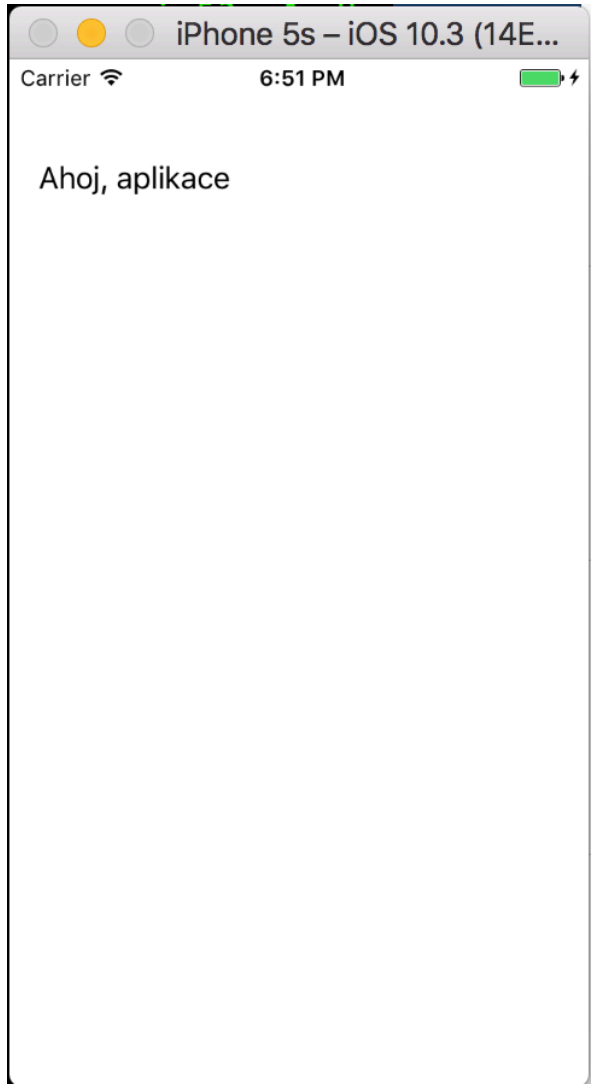
- VC je vlastníkem View. (následně subviews).
- UIWindow — key and visible. (UIResponder)
- ViewController — předávání řízení (toku událostí z iOS).
- Přechod od jednoho VC k druhému.
  - Předání toku událostí.
  - Změny v topologii Views.
- Životní cyklus ViewControllerů.

Životní cyklus ViewController



M-V-C, Základní demo

```
class MujModel {
    var obsah : String = "Ahoj, aplikace"
}

class ViewController: UIViewController {

    @IBOutlet var lei : UILabel?

    let mujmodel = MujModel()

    override func viewDidLoad() {
        super.viewDidLoad()

        //
        lei?.text = mujmodel.obsah
    }
}
```



Forma referencování Outlets

- Vlastníkem Outletu je View, ne VC.
  - (Co se stane, když outlet odstraním ze superView?) setHidden.
- @IBOutlet var textik : UILabel!
  - strong reference s garantovanou/předpokládanou hodnotou
- @IBOutlet var textik : UILabel?
  - je v XCode výchozí. Zřejmě reaguje na lazy loadView.
  - @IBOutlet weak var textik : UILabel?
- Totálně v rozporu s myšlenkou Swiftu.

Co se stane po lei?.text = ”…”

- Předpokládejme *mainThread*.
  - Jinak kritická chyba. Stále: jaké vlákno vykonává tento kód?
- Nastaví se datový obsah UILabel.text, setter.
- Nastaví se příznak ”needsDisplay”.
  - Propaguje se směrem k super-views až po UIWindow.
- V dalším cyklu RunLoop vyvolá překreslení UIWindow.
  - tj. **UILabel.setText neprovádí okamžitou změnu obrazovky!**

M-V-C, KVO verze

```
//
class MujModel : NSObject {
    dynamic var obsah : String = "Ahoj, aplikace"
}
//
class ViewController: UIViewController {
    @IBOutlet var lei : UILabel?
    let mujmodel = MujModel()
    //
    override func viewDidLoad() {
        super.viewDidLoad()

        lei?.text = mujmodel.obsah

        mujmodel.addObserver(self, forKeyPath: #keyPath(MujModel.obsah), options:
[.new, .old], context: nil);
    }
    //
    override func observeValue(forKeyPath keyPath: String?,
                               of object: Any?,
                               change: [NSKeyValueChangeKey : Any]?,
                               context: UnsafeMutableRawPointer?)
    {
        self.lei?.text = self.mujmodel.obsah
    }
}
```

M-V-C, didSet verze, obezřetně!

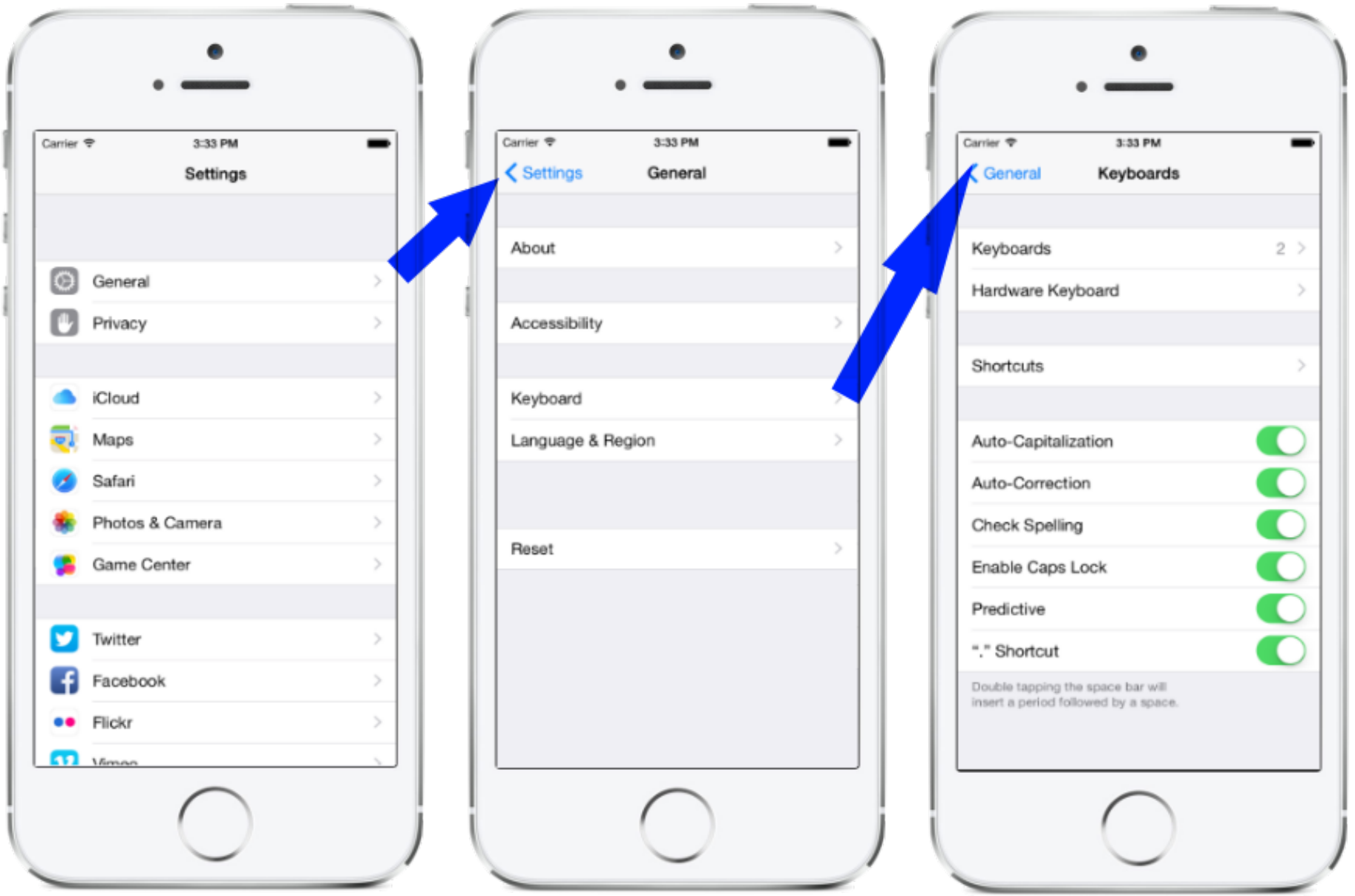
```
class MujModel {
    //
    weak var vc : ViewController? = nil

    var obsah : String = "Ahoj, aplikace" {
        didSet {
            // bacha na vlakno
            vc?.lei?.text = obsah;
        }
    }
}
```

```
class ViewController: UIViewController {
    //
    @IBOutlet var lei : UILabel?
    //
    let mujmodel = MujModel() ;
    //
    override func viewDidLoad() {
        super.viewDidLoad()
        //
        mujmodel.vc = self;
        //
        lei?.text = mujmodel.obsah
    }
```

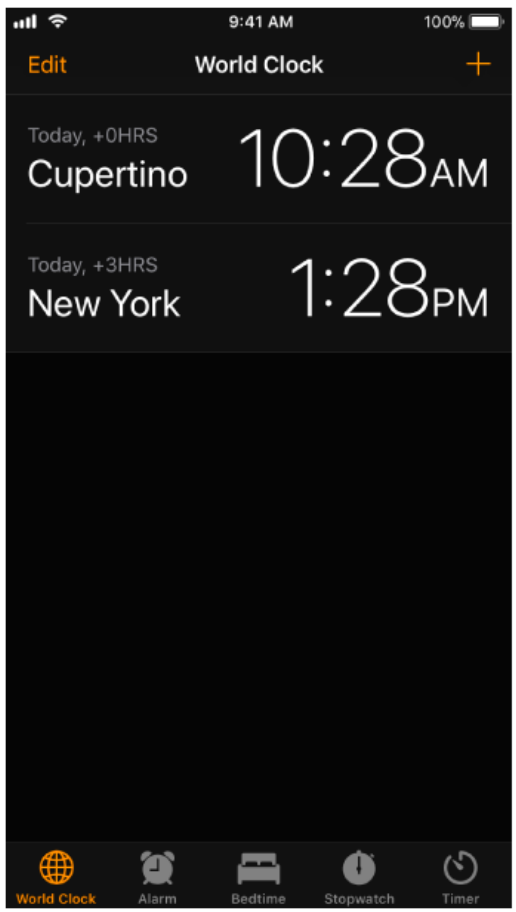
Multi-VC aplikace

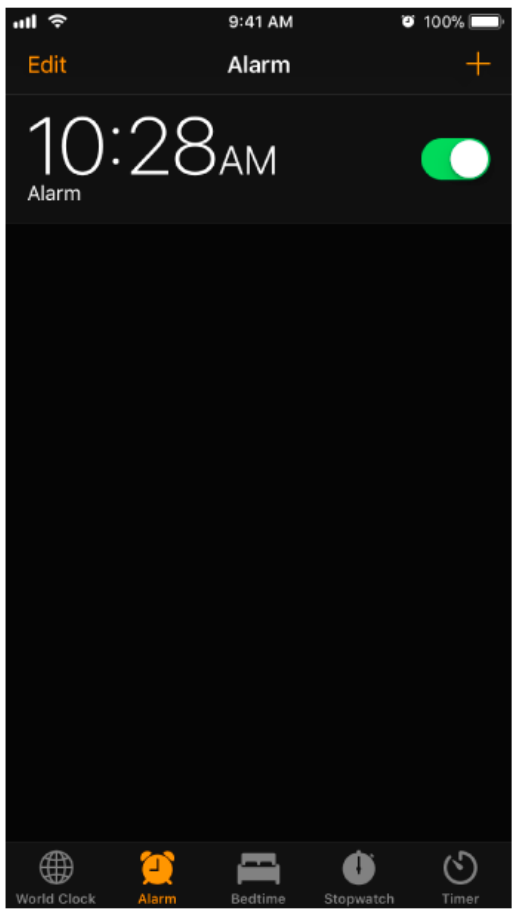
- Potřebujeme ”přepínat obrazovky”, tj. potlačit jeden VC a aktivovat druhý VC.
  - Změna v hierarchii views, přesměrování VC.
  - Abstraktní VC — pouze řídí další (vnořené) VC.
- ***UINavigationController,***
- *UITabBarController,*
- Styl *Master-Detail* — 2 VC spojené přes UISplitViewController.

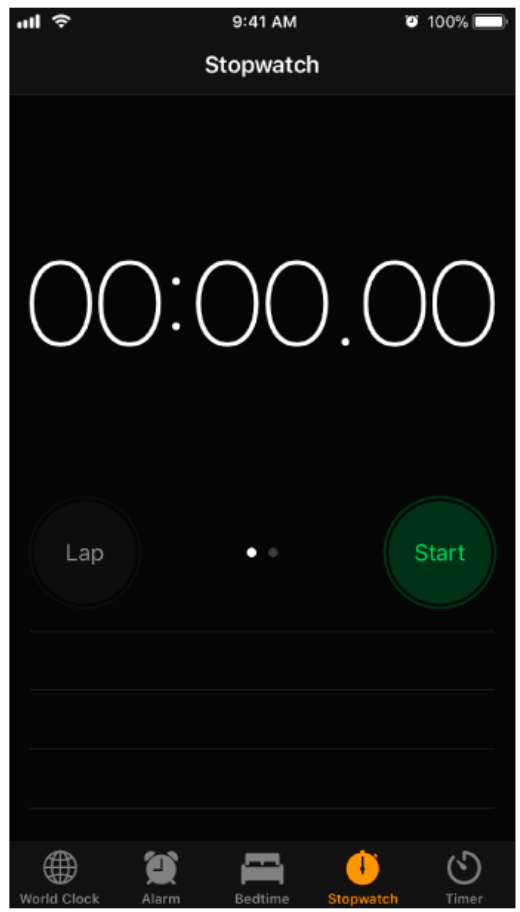
Navigation VC

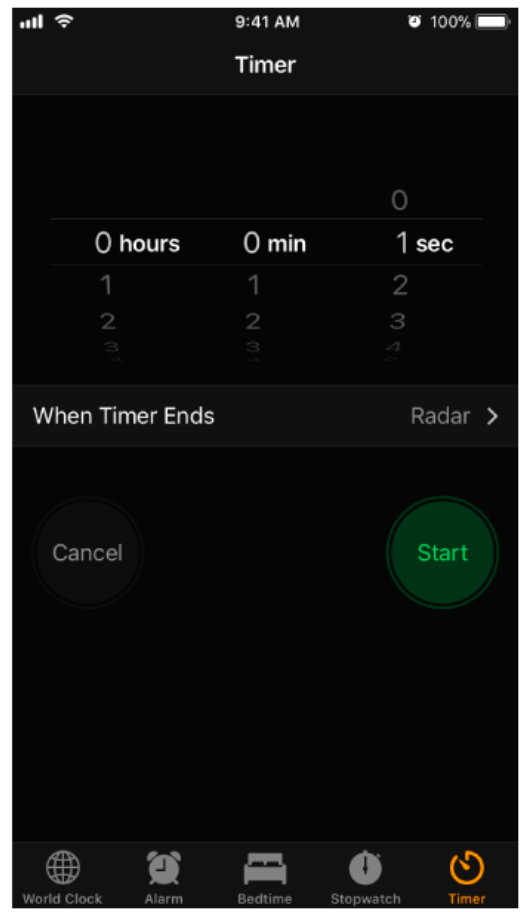


TabBar VC


9:41 AM
100%
Edit
World Clock
Today, +0HRS
Cupertino
10:28AM
Today, +3HRS
New York
1:28PM
World Clock
Alarm
Bedtime
Stopwatch
Timer

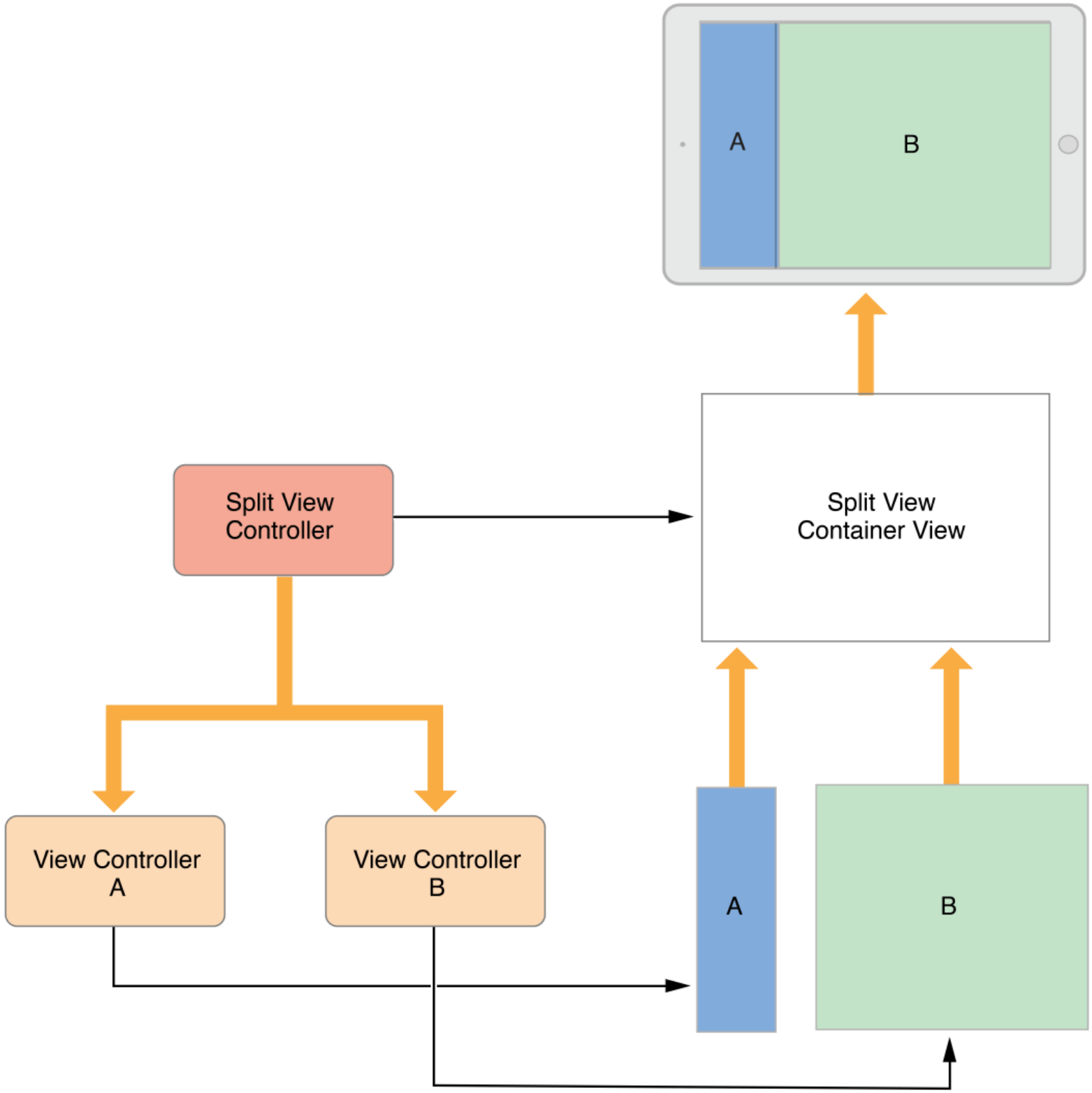

9:41 AM
100%
Edit
Alarm
10:28AM
Alarm
World Clock
Alarm
Bedtime
Stopwatch
Timer


9:41 AM
100%
Stopwatch
00:00.00
Lap
Start
World Clock
Alarm
Bedtime
Stopwatch
Timer


9:41 AM
100%
Timer
0 hours
0 min
1 sec
When Timer Ends
Radar
Cancel
Start
World Clock
Alarm
Bedtime
Stopwatch
Timer

Funkce Navigation-VC

- Tzv. ”container VC”. Má:
  - rootViewController — základní/počáteční VC.
  - viewControllers : [UIViewController]
  - contentView — View pro vnitřní obsah.
- *pushViewController*(o:VC, animated:Bool)
- *popViewController*(animated:Bool)
- Obdobně s TabBarVC.


A
B
Split View
Controller
Split View
Container View
View Controller
A
View Controller
B
A
B

Přechod na druhý VC

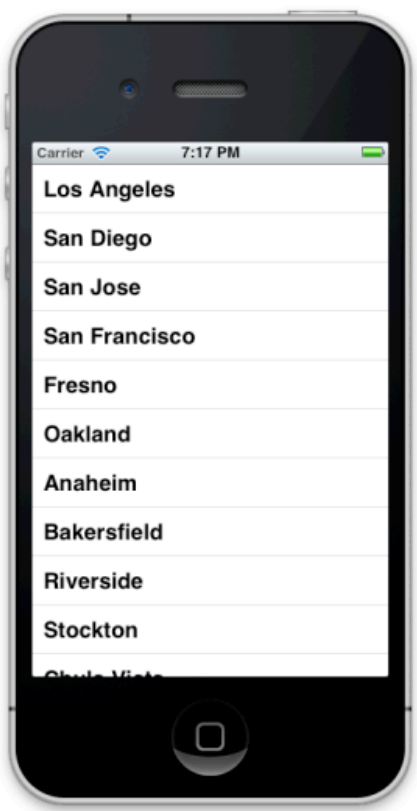
- Deaktivace původního vc : UIViewController:
  - vc.willMove(toParentViewController: nil)
  - vc.view.removeFromSuperview()
  - vc.removeFromParentViewController() — přeposílání zpráv.
  - (vc.viewWillDisappear, vc.viewDidDisappear).

Přechod na druhý VC

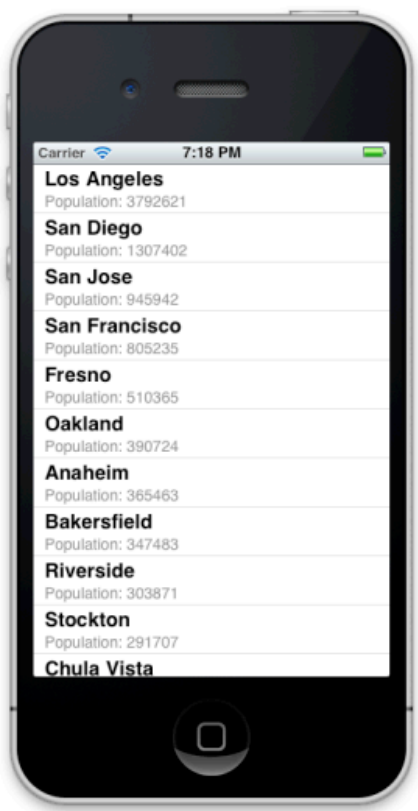
- Aktivace nového VC, nvc:
  - addChildViewController(nvc) — přeposílání zpráv.
  - view.addSubView(nvc.view)
  - nvc.view.frame = view.bounds
  - nvc.view.autoresizingMask = [.flexibleWidth, .flexibleHeight]
  - nvc.didMove(toParentViewController: self)
- Nav-VC a TabBarVC takto přepínají VC.

Table View Controller

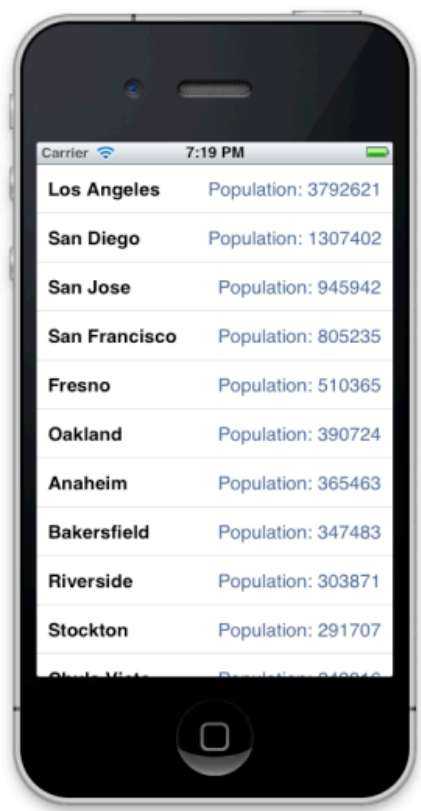
- Nejdůležitější prvek UI iOS.
- Seznam buněk (TableViewCell).



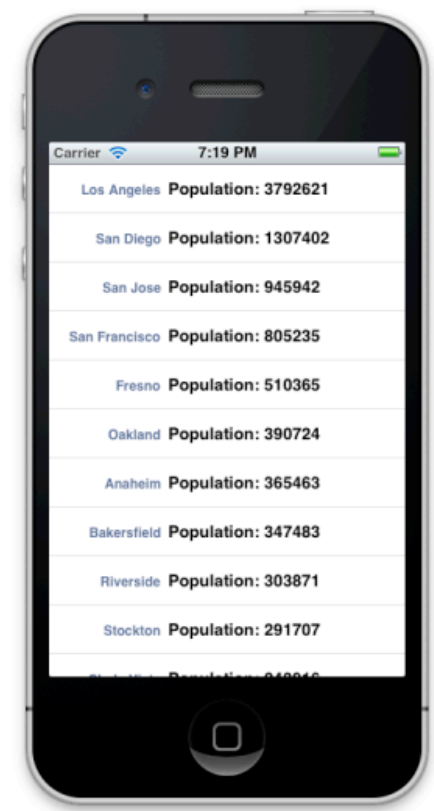
UITableViewCellStyleDefault



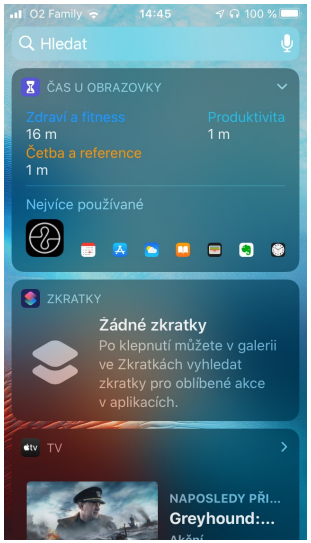
UITableViewCellStyleSubtitle

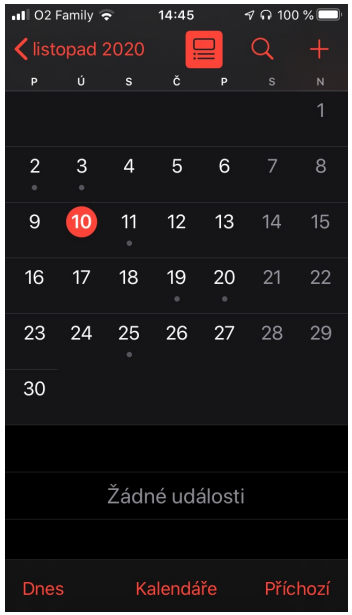


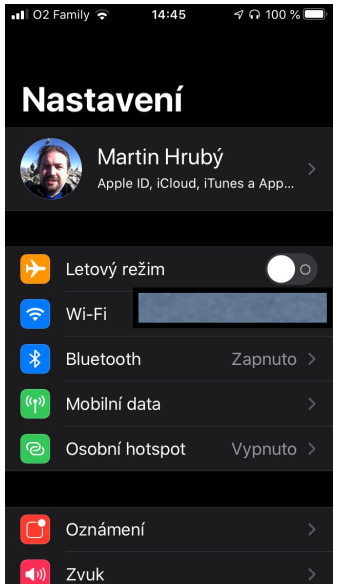
UITableViewCellStyleValue1

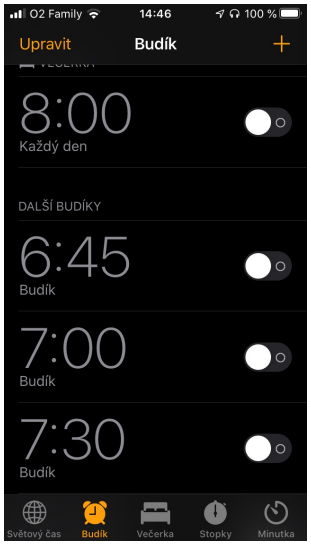


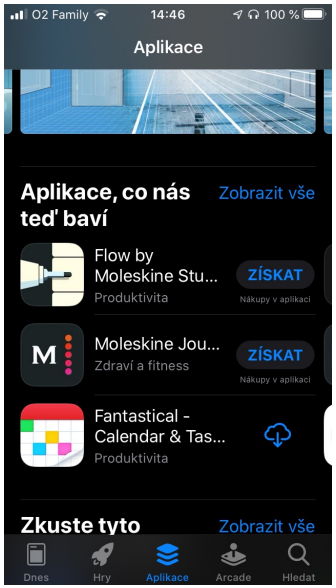
UITableViewCellStyleValue2

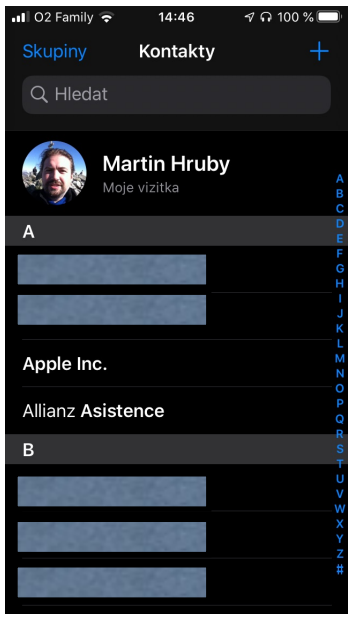

Hledat
ČAS U OBRAZOVKY
Zdraví a fitness
16 m
Produktivita
1 m
Četba a reference
1 m
Nejvíce používané
ZKRATKY
Žádné zkratky
Po klepnutí můžete v galerii ve Zkratkách vyhledat zkratky pro oblíbené akce v aplikacích.
TV
NAPOSLEDY PŘI...
Greyhound:...


listopad 2020
P Ú S Č P S N
Žádné události
Dnes
Kalendáře
Příchozí

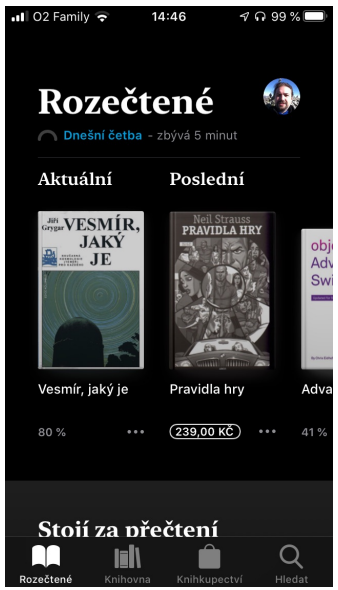

Nastavení
Martin Hrubý
Apple ID, iCloud, iTunes a App...
Letový režim
Wi-Fi
Bluetooth
Zapnuto
Mobilní data
Osobní hotspot
Vypnuto
Oznámení
Zvuk

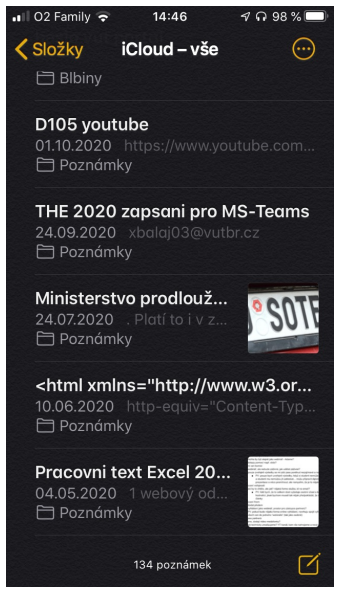

Upravit
Budík
8:00
Každý den
DALŠÍ BUDÍKY
6:45
Budík
7:00
Budík
7:30
Budík
Světový čas
Budík
Večerka
Stopky
Minutka

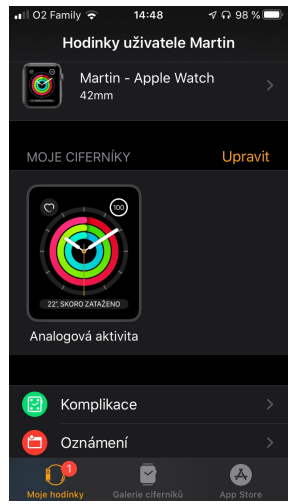

Aplikace
Aplikace, co nás teď baví
Zobrazit vše
Flow by Moleskine Stu...
ZÍSKAT
Produktivita
Moleskine Jou...
ZÍSKAT
Zdraví a fitness
Fantastical - Calendar & Tas...
Produktivita
Zkuste tyto
Zobrazit vše
Dnes
Hry
Aplikace
Arcade
Hledat

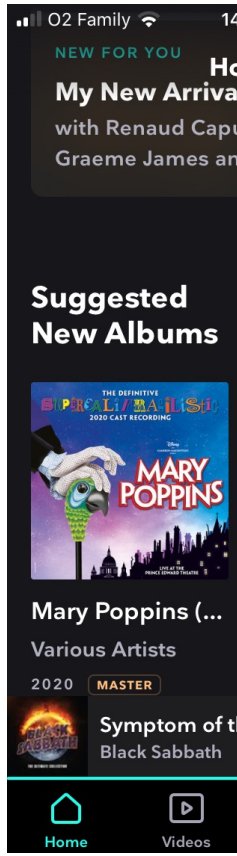

Skupiny
Kontakty
Hledat
Martin Hruby
Moje vizitka
A
Apple Inc.
Allianz Asistence
B


Brno
Oblačno
21 22 23 0 1 2 3 4
5° 5° 5° 5° 5° 4° 4° 4°
středa 6 3
čtvrtek 6 3
pátek 7 3
sobota 7 4
neděle 7 5
pondělí 8 4
úterý 8 4
středa 8 3
čtvrtek 7 3
Dnes: oblačno. Nejvyšší teplota vystoupá na 6°. V noci převážně


Rozečtené
Dnešní četba - zbývá 5 minut
Aktuální
Poslední
Vesmír, jaký je
Pravidla hry
80 %
239,00 Kč
41 %
Stojí za přečtení
Rozečtené
Knihovna
Knihkupectví
Hledat


Složky
iCloud – vše
Blbiny
D105 youtube
01.10.2020 https://www.youtube.com...
Poznámky
THE 2020 zapsani pro MS-Teams
24.09.2020 xbalaj03@vutbr.cz
Poznámky
Ministerstvo prodlouž...
24.07.2020 . Platí to i v z...
Poznámky
<html xmlns="http://www.w3.or...
10.06.2020 http-equiv="Content-Typ...
Poznámky
Pracovni text Excel 20...
04.05.2020 1 webový od...
Poznámky
134 poznámek

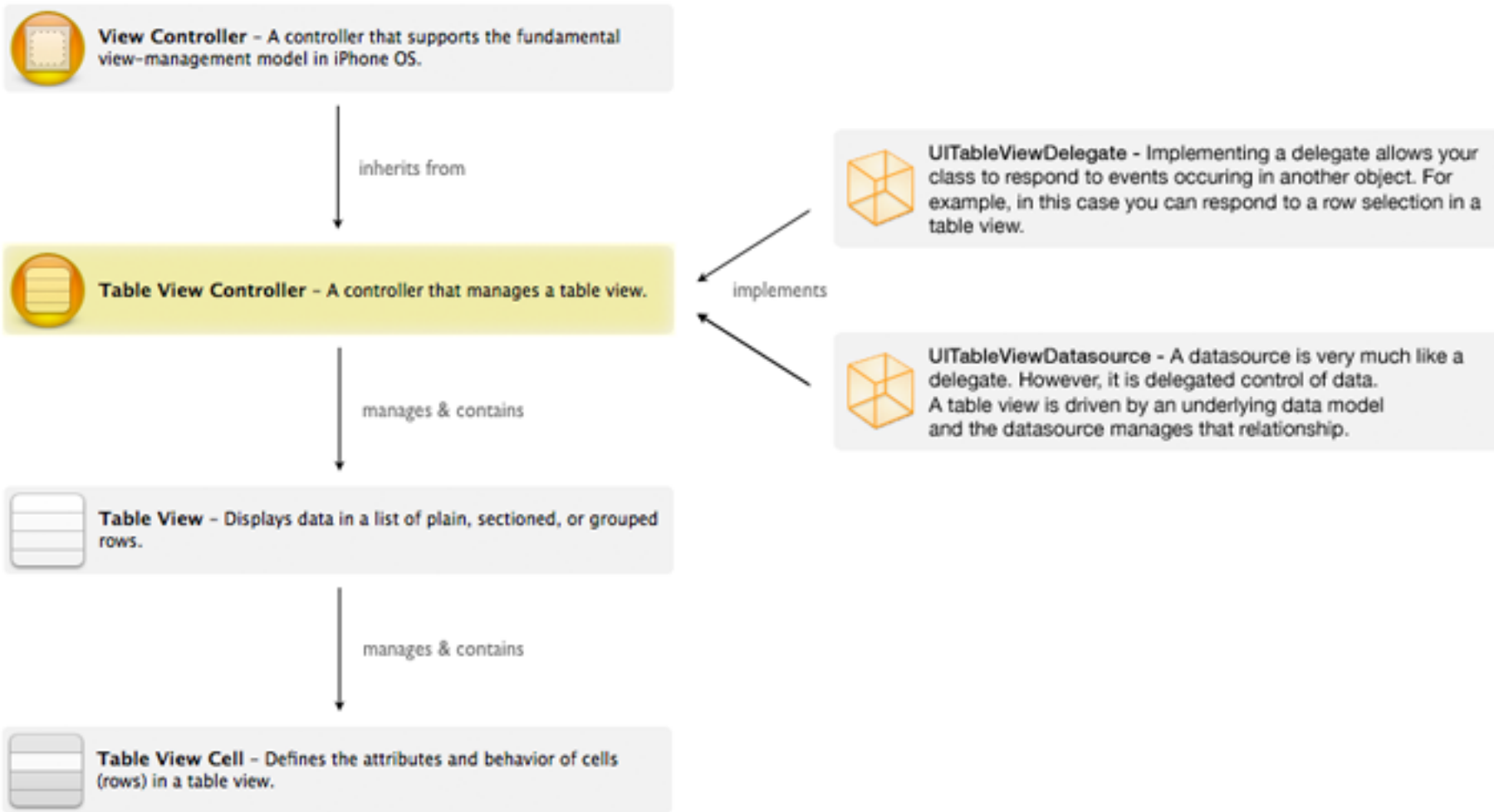

Hodinky uživatele Martin
Martin - Apple Watch
42mm
MOJE CIFERNÍKY
Upravit
Analogová aktivita
Komplikace
Oznámení
Moje hodinky
Galerie ciferníků
App Store


NEW FOR YOU
My New Arriva
with Renaud Capu
Graeme James an
Suggested New Albums
MARY POPPINS
Mary Poppins (...
Various Artists
2020
MASTER
Symptom of t
Black Sabbath
Home
Videos

Tabulka jako koncept

- *TableView* — *View*, dotazuje se přes *dataSource*/*delegate*.
  - *TableViewDataSource* — struktura tabulky, obsah.
  - *TableViewDelegate* — geometrie, události, povolení k …
  - *TableViewController* — VC implementující *dataSource* a *delegate*.
- Zapojení TableView v aplikačním VC:
  - Aplikační VC odvozuje od TVC, nahrazuje vybrané metody dataSource/delegate.
  - Aplikační VC má TV jako subView a tvoří dataSource. Co delegate?

UITableViewController

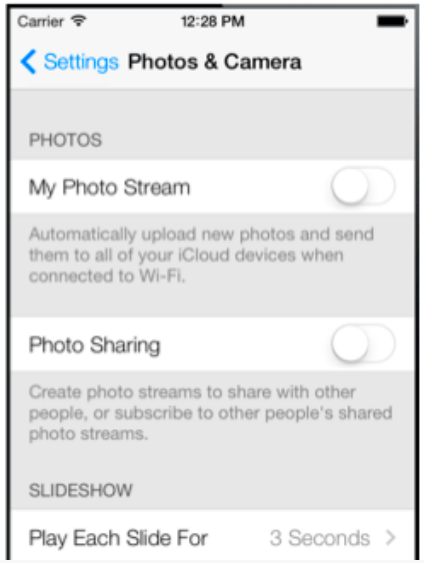


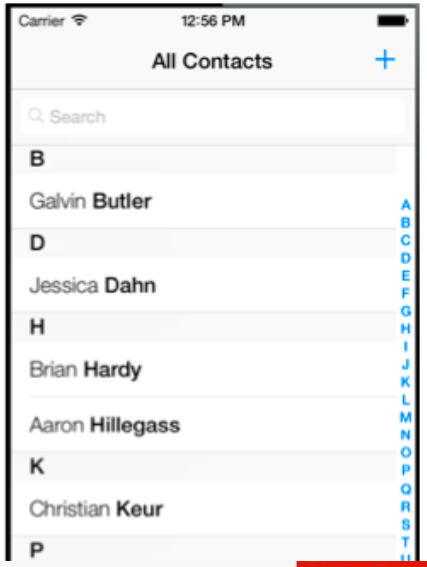
UITableViewController

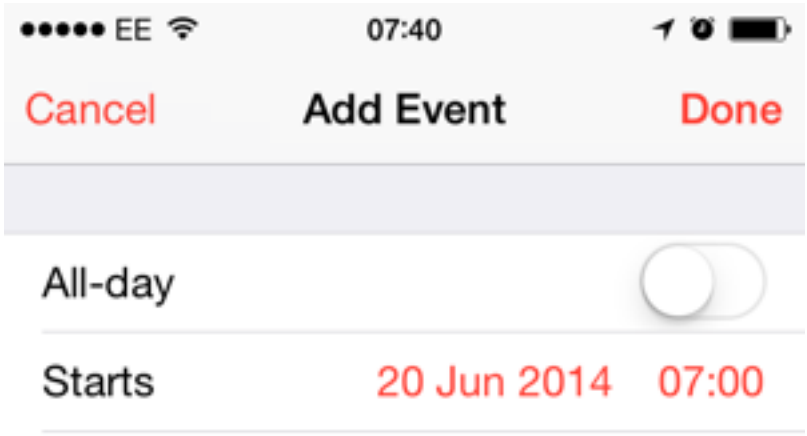
- Implementuje dataSource a delegate protokoly tabulky.
- Uživatelský VC dědí z TVC, přepisuje potřebné metody dataSource a delegate protokolů.

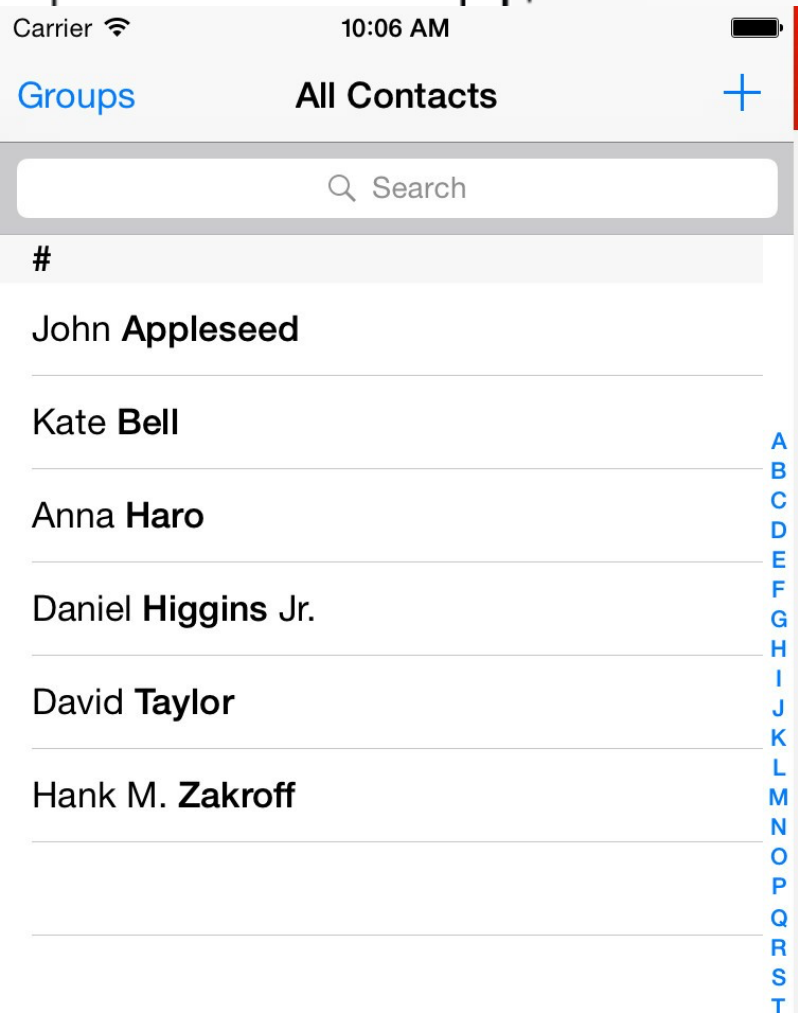
UITableView

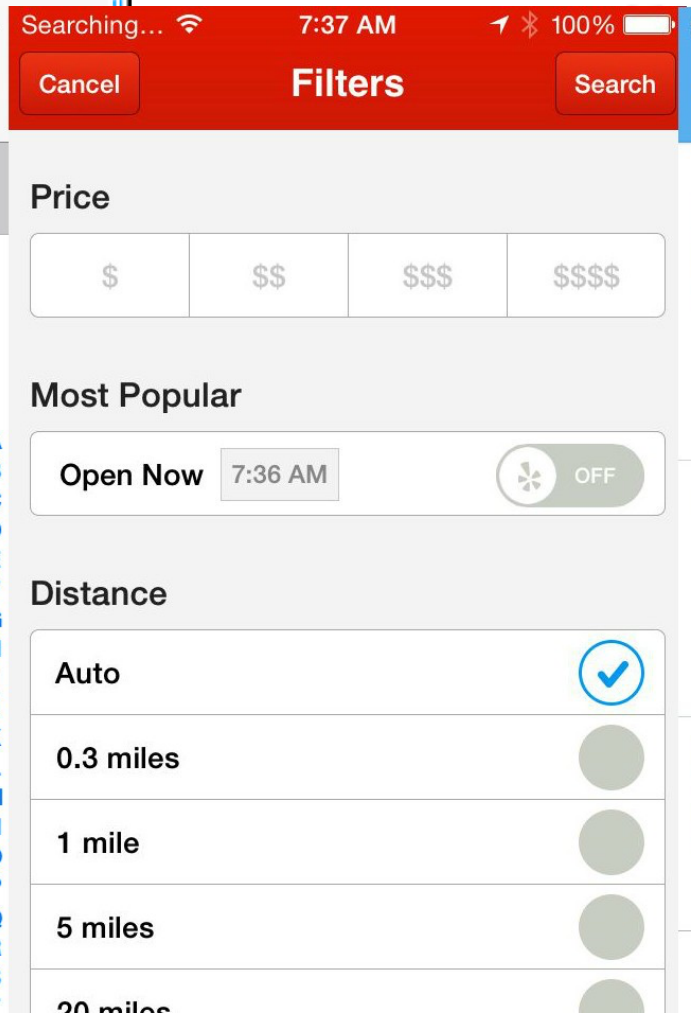
- Odvozen od UIScrollView (přesahuje rámec obrazovky).
- Provádí rozmístění (layout) buněk (Cell) ve skupinách.
- Obecný heterogenní dynamický soupis buněk vertikálně řazených.
- UITableViewCell — prototyp buňky tabulky

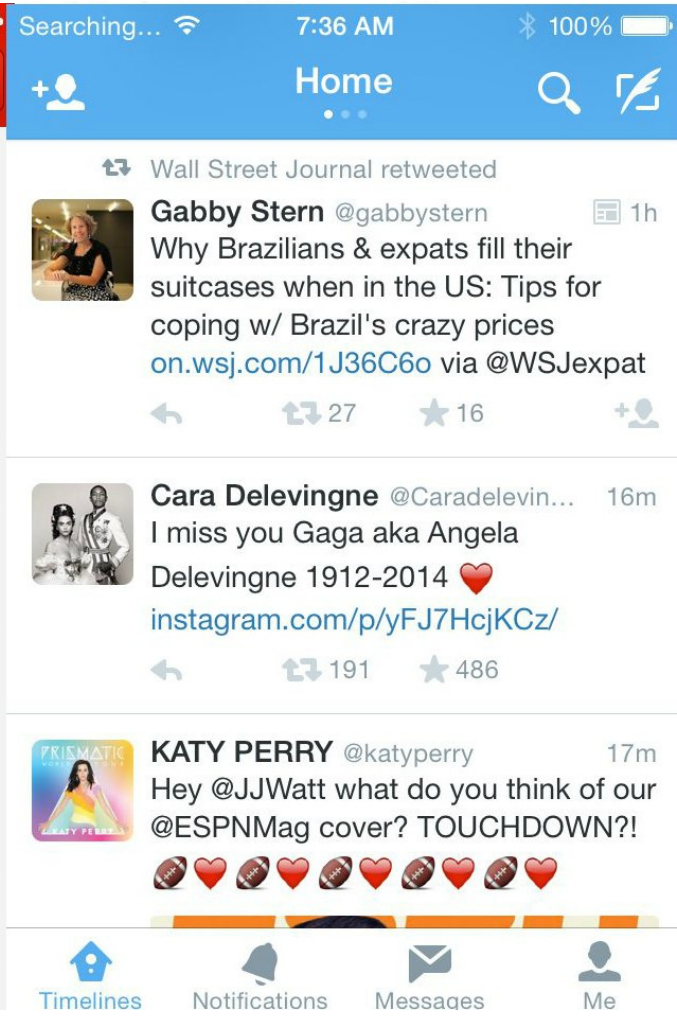

Carrier
12:28 PM
Settings
Photos & Camera
PHOTOS
My Photo Stream
Automatically upload new photos and send them to all of your iCloud devices when connected to Wi-Fi.
Photo Sharing
Create photo streams to share with other people, or subscribe to other people's shared photo streams.
SLIDESHOW
Play Each Slide For
3 Seconds


Carrier
12:56 PM
All Contacts
Search
B
Galvin Butler
D
Jessica Dahn
H
Brian Hardy
Aaron Hillegass
K
Christian Keur
P


EE
07:40
Cancel
Add Event
Done
All-day
Starts
20 Jun 2014
07:00

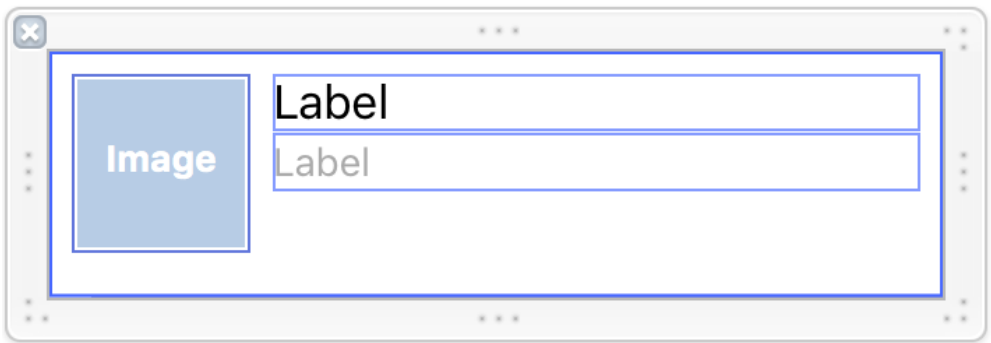

Carrier
10:06 AM
Groups
All Contacts
Search
#
John Appleseed
Kate Bell
Anna Haro
Daniel Higgins Jr.
David Taylor
Hank M. Zakroff

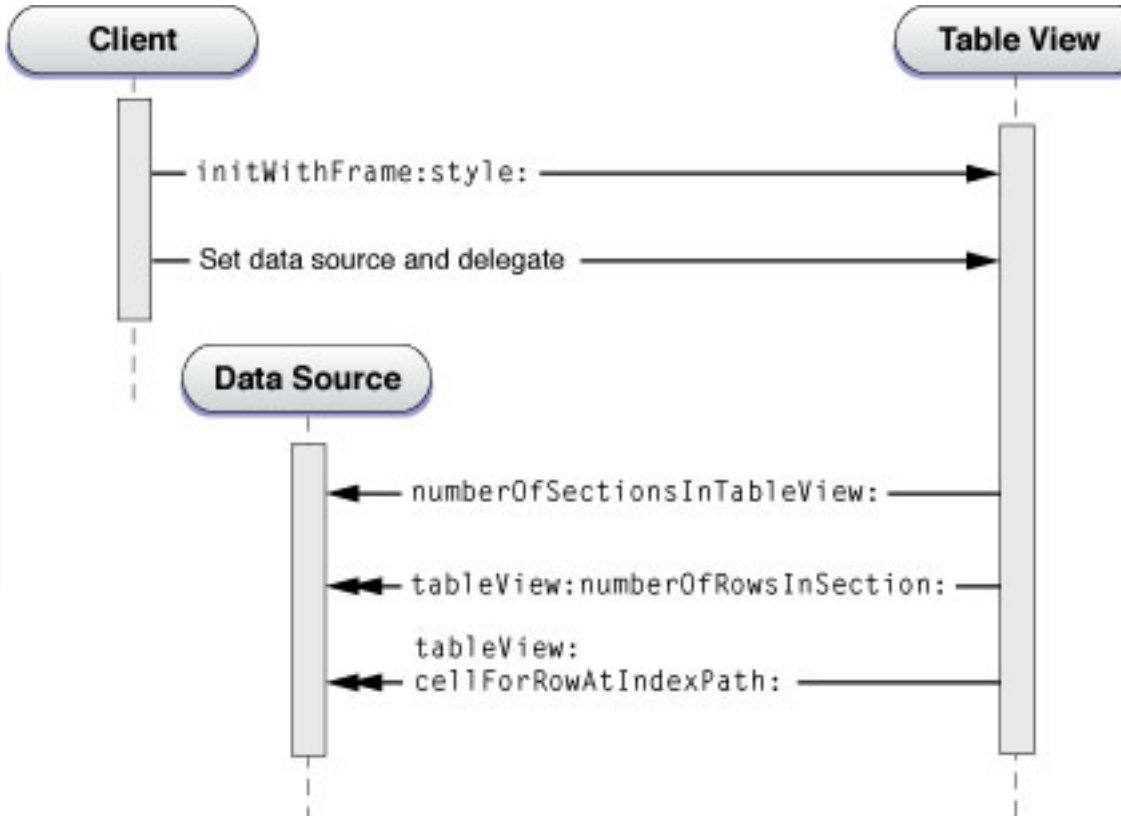

Searching...
7:37 AM
100%
Cancel
Filters
Search
Price
$
$$
$$$
$$$$
Most Popular
Open Now
7:36 AM
OFF
Distance
Auto
0.3 miles
1 mile
5 miles


Searching...
7:36 AM
100%
Home
Wall Street Journal retweeted
Gabby Stern @gabbystern 1h
Why Brazilians & expats fill their suitcases when in the US: Tips for coping w/ Brazil's crazy prices on.wsj.com/1J36C6o via @WSJexpat
27
16
Cara Delevingne @Caradelevin... 16m
I miss you Gaga aka Angela Delevingne 1912-2014
instagram.com/p/yFJ7HcjKCz/
191
486
KATY PERRY @katyperry 17m
Hey @JJWatt what do you think of our @ESPNMag cover? TOUCHDOWN?!
Timelines
Notifications
Messages
Me

Řízení — dataSource

- TVC zprostředkovává obsah (Model) do View.
- Model zde vystupuje jako ”dataSource” API:
  - počet sekcí, počet řádků v sekci.
  - sestavení TableViewCell pro zadanou souřadnici (IndexPath).

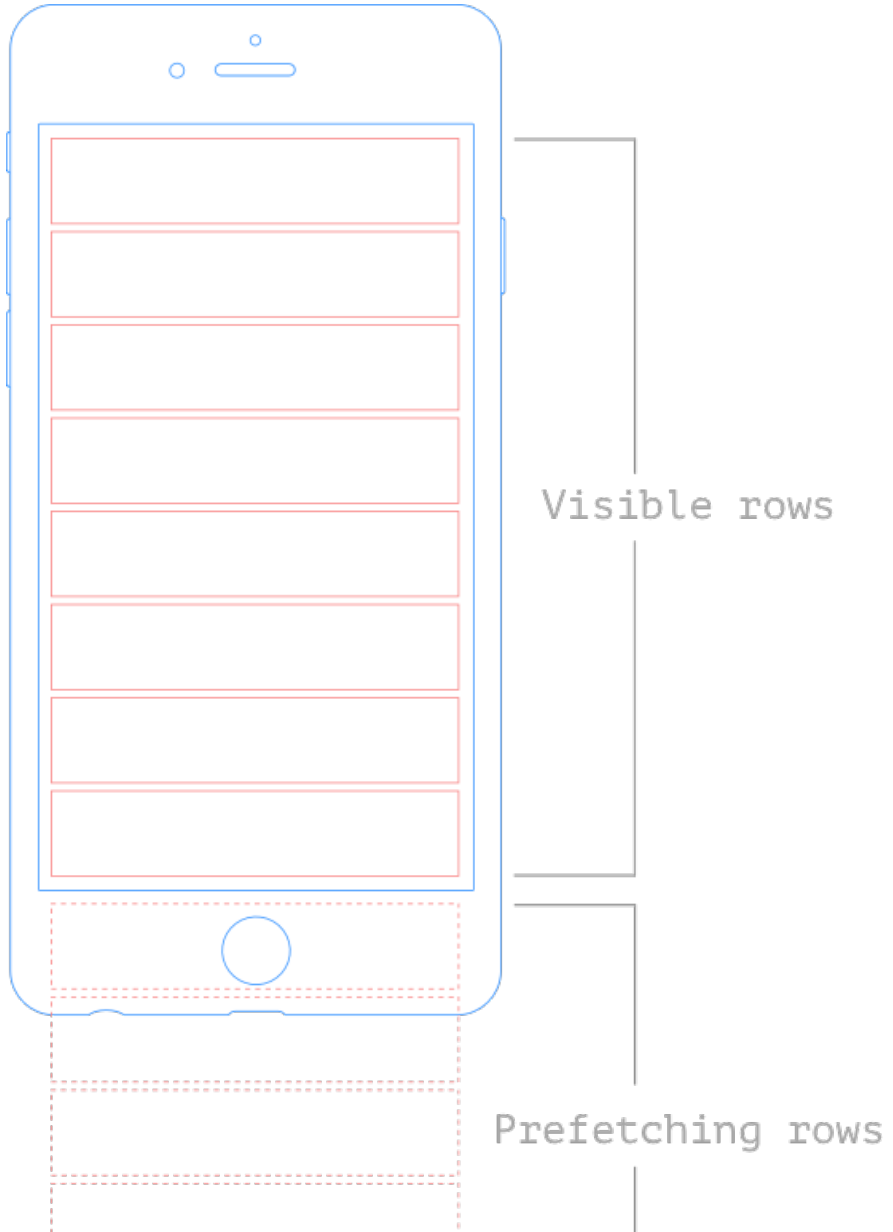




Řízení TV, dynamika

- TV zjistí datový rozsah (sekce, řádky).
- Předpokládá se, že viditelná je pouze část buněk.
  - Ty jsou alokovány a inicializovány.
  - Při posuvu tabulkou se dynamicky volá dataSource na doplňování obsahu buněk.
  - Znovupoužití objektů buněk (pool). Identifikátory buněk.

Dynamika přístupu na dataSource



Demo

```
class SimpleTab: UITableViewController {

    var seznam = ["Jeden", "Druhy", "Treti"];

    override func tableView(_ tableView: UITableView,
                numberOfRowsInSection section: Int) -> Int
    {
        return seznam.count
    }

    override func tableView(_ tableView: UITableView,
                cellForRowAt indexPath: IndexPath) -> UITableViewCell
    {
        let cell = tableView.dequeueReusableCell(withIdentifier: "cell",
for: indexPath)

        cell.textLabel?.text = seznam[indexPath.row]

        return cell
    }
}
```

Dynamika, posuv v Table

- Chod aplikace musí být hladký.
- Inicializace buněk může brzdit chod tabulky (rychlý posuv).
- Při náročnější inicializaci se přechází do bočních vláken (GCD).

```
override func tableView(_ tableView: UITableView,
                        cellForRowAt indexPath: IndexPath) ->
UITableViewCell
{
    let cell = tableView.dequeueReusableCell(withIdentifier: "tabik", for:
indexPath)

    cell?.imageView?.image = placeHolder;
    // do vedlejsiho/globalniho vlakna naplanuju...
    DispatchQueue.global().async {
        // sestaveni obsahu pro bunku tabulky
        if let loaded = myModel.load(cosi) {
            // pokud mame obsah, jdeme zpatky do mainThread
            DispatchQueue.main.async {
                // pokud na indexPath stale existuje bunka
                if let ncell = tableView.cellForRow(at: indexPath) {
                    // aktualizuji
                    ncell.imageView?.image = loaded;
                }
            }
        }
    }
    //
    return cell;
}
```

Závěr

- Exkurze do světa UIKit/StoryBoard.
- Některé koncepty ve SwiftUI přetrvávají.
- Příště:
  - Konstrukce typických částí v aplikaci SwiftUI.
  - ObservableObjects.
  - Úvod do Combine.