

Ať se z kódu nepráší!

Juraj Michálek

7. 5. 2012 - FI MUNI - Brno

<http://georgik.sinusgear.com>

<http://twitter.com/sinusgear>

Niečo o mne



Čo nás čaká

- * Testovanie v C
- * Kvalita kódu, open source a projekty
- * C vo svete Applu, Objective-C

* Když se dívám na svůj kód,
tak se musím pochválit.
Nevypadá věru zle,
zkrátka program jak má být.

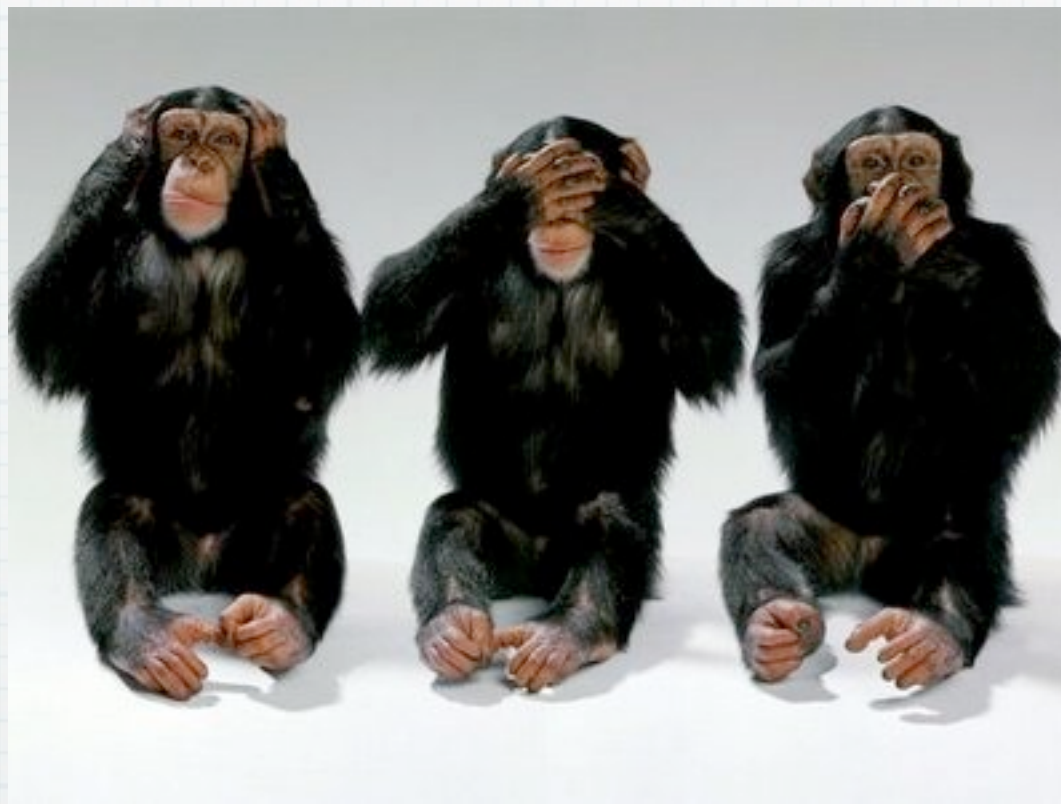
* Před naší, za naší
ať se z kódu nepráší! Hej!

Háá! Chybáá!

Testovanie

- * žiadne
- * manuálne
- * automatické

Žiadne testovanie



Ľudová múdrosť

*** Bez testů do vývoje nelez!**

Manuálne - black box

* Spustíme a aplikácia ...



Manuálne - debugger



* Když něco rozeberem, tak leda debuggerem

* <http://georgik.sinusgear.com/2010/09/19/zaznam-z-prednasky-kdyz-neco-rozeberem-tak-leda-debuggerem/>

Proč automatizovat?

* Som vývojář alebo debuggrista?

Automatizované testy

- * Záchranná sieť pre vývojárov
- * Živá dokumentácia
- * Lepší dizajn
- * Viac času na vývoj

Typy testov

- * Unit testing - jednotkové testy
- * Integračné testy
- * Akceptačné testy
- * Regresné testy

Test Driven Devel.

- * Testy prvé

- * Potom kód

Unit Testing

- * izolácia
 - * setUp - príprava prostredia
 - * test - samotný test
 - * tearDown - upratanie prostredia

Príklad testu

- * `int a=41;`
- * `a++;`
- * `assert( a == 42, "Universe should be ok");`

Minunit

- * **Minimalist Unit Testing Framework for C**
- * <http://www.jera.com/techinfo/jtns/jtn002.html>

Minunit - code

```
/* file: minunit.h */
#define mu_assert(message, test) do { if (!(test)) return message; } while (0)
#define mu_run_test(test) do { char *message = test(); tests_run++; \
                                if (message) return message; } while (0)
extern int tests_run;
```

mu_runttest

* `mu_runttest( test_currency );`

* `mu_runttest( test_accounting );`

mu_assert

*** mu_assert("Should not be null", state == null);**

Spustenie

- * gcc *.c -o test
- * ./test
- * ALL TEST PASSED
- * Tests run: 2

gcc - c99

- * `for (int i=0; i<42; i++)`
- * error: 'for' loop initial declaration used outside C99 mode
- * zmena normy parametrom:
 - * `gcc -std=c99`

Check framework

- * <http://check.sourceforge.net/>

- * Inštalácia

 - * `./configure`

 - * `make`

 - * `make install`

Organizácia testov

- * Test Case

 - * testovaný prípad

- * Test Suite

 - * sada Test Case

- * Test Runner

Test Fixtures

- * `void setup();`

- * `void teardown();`

Kontrólne fcie

- * `fail_if ( employee->age != 43, "Problem");`

- * `fail_unless(employee->age == 43,
"Incorrect age.");`

Kompiláció

* `gcc -lcheck test.c -o test`

Spustenie testu

- * `./test`

- * Príklad:

- * Running suite(s): Employee testing

- * 0%: Checks: 1, Failures: 1, Errors: 0

- * `main.c:45:F:Core tests:test_age:0:`
Incorrect age.

Ďalšie frameworky

- * C/C++

- * CxxTest: <http://cxxtest.com/>

- * Boost: <http://boost.org>

Continuous Integration

Test Statistics Grid 							
Job	Success		Failed		Skipped		Total
	#	%	#	%	#	%	#
  as3-signals	518	100%	0	0%	0	0%	518
  as3corelib	0	0%	0	0%	0	0%	0
  fixel	0	0%	0	0%	0	0%	0
  mate-framework	0	0%	0	0%	0	0%	0
  parsley-core	0	0%	0	0%	0	0%	0
  picasaflashapi	20	100%	0	0%	0	0%	20
  pushbuttonengine	0	0%	0	0%	0	0%	0
  robotlegs-framework	218	100%	0	0%	0	0%	218
  smileapp	3	100%	0	0%	0	0%	3
  smileapp-flexmojos	3	100%	0	0%	0	0%	3
  swiz-framework	30	100%	0	0%	0	0%	30
Total	792	100%	0	0%	0	0%	792

* Jenkins

* <http://ci.sinusgear.com>

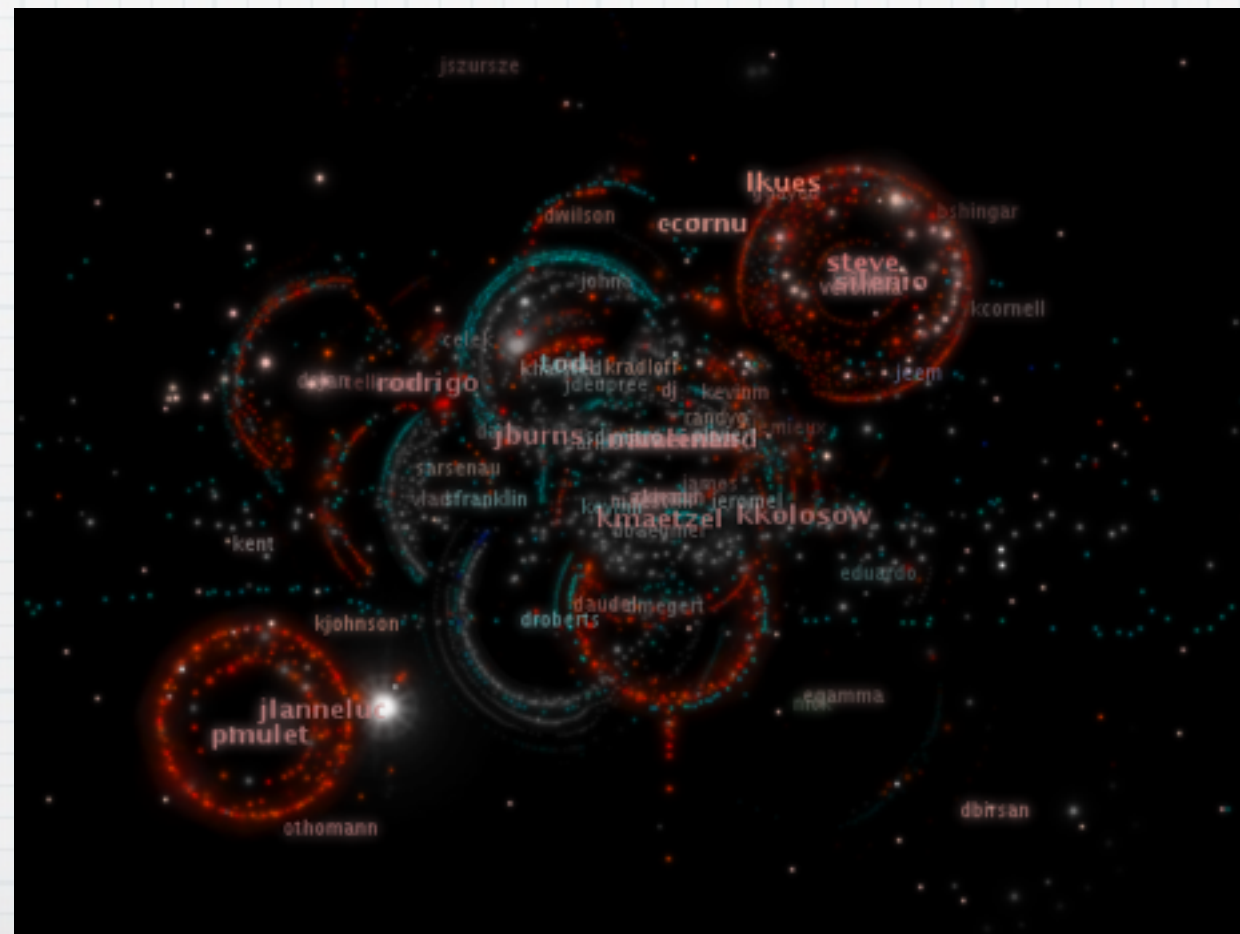
Open source

- * Projekty

- * Licencie

Code_swarm

- * Organic software visualization of project repositories
- * <http://code.google.com/p/codeswarm/>



Gource



* <http://code.google.com/p/gource/>

Ohloh.net

* hodnotenie open source projektov



Lynda.com

* Dobrý zdroj znalostí



se-radio.net

Software Engineering Radio

The Podcast for Professional Software Developers

Kvalita kódu

- * Kód sa viackrát číta, než sa píše.
- * Review kódu

Otvorenie kódu



- * GitHub.com
- * Social Coding
- * fork

Licencie

- * Free software
 - * GNU/GPL, LGPL
- * Open source
 - * BSD, MIT, Apache
 - * business friendly

Čítajte licencie

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

Čítajte licencie

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

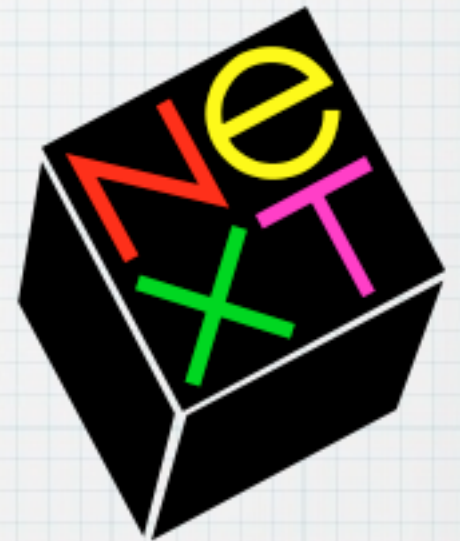
The Software shall be used for Good, not Evil.

Objective-C

- * programovací jazyk vycházející z C
- * objektové principy zo Smalltalku

Objective-C

- * 1983 - Brad Cox a Tom Love
 - * zakládají StepStone
- * 1988 - Steve Jobs - NeXT
 - * licencuje Objective-C od StepStone
 - * rozširuje GCC, vytvárá AppKit



Objective-C

- * 1996 - Apple ohlásil akvizíciu NeXT
 - * 429 mil. \$
- * Cocoa framework



Apple a C

- * Mac OS 10.7
- * Xcode 4
- * Command Line Tools for Xcode
- * <http://developer.apple.com>

Xcode 4



*** Nutné: Mac OS X 10.7**

Objective-C

- * Syntax: camelCase

- * Prefix: NS (od NextStep)

Hello FI MUNI!

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[])
{
    @autoreleasepool {
        NSLog(@"Hello, FI MUNI!");
    }
    return 0;
}
```

Command + R

NSString

- * NSString *who = @"FI MUNI";

- * NSLog(@"Hello %@!", who);

Boolean

* `BOOL isFIBest = YES;`

* `BOOL isCBoring = NO;`

Správa pamäte

- * Nové verzie Mac OS podporujú auto-garbage collection.
- * Nespoliehajte sa na to!
 - * Naučte sa manažovať pamäť.
 - * Nefunguje na mobilných zariadeniach.

Retain Counter

- * alokácia - zvýšenie počítadla
- * dealokácia - zníženie počítadla

Pravidlá

- * Ak objekt vlastníš, musíš ho uvoľniť.
- * Ak objekt nevlastníš, nesmieš ho uvoľniť.

Vlastnenie objektu

- * alloc, new, retain, copy
- * Každý ANRC musí mať release!

Alokáció - new

```
// Creating object  
NSDate *myDate = [NSDate new];  
  
// Release object  
[myDate release];  
|
```

Alokácia a inicializácia

```
// Call alloc an then init
NSDate *myDate = [NSDate alloc];
myDate = [myDate init];

// Common way
NSDate *myDate = [[NSDate alloc] init];
```

Preferovaný spôsob

Rozdelenie kódu

- * rozhranie (interface) - *.h

- * implementácia - *.m

Interface

```
@interface Employee : NSObject {  
    @private  
    NSString *name;  
    NSString *surname;  
    NSDate *age;  
}  
@end
```

Implementácia

```
@implementation Employee
```

```
- (void) myMethod {  
    NSLog(@"Hello");  
}
```

```
@end
```

Znak pred metódou

- * - (void) myMethod
- * znak mínus - metóda inštancie
- * znak plus - statická metóda

Prístup k zdrojom

- * NSURL

- * NSURL *myURL = [NSURL
fileURLWithPath:path];

- * file:///Users/georgik/file.txt

Prečo používať NSURL?

- * Pretože to tak Apple povedal.
- * rýchlejšie na iOS
- * zachytenie chyby v prípade zlého formátu
- * omnoho mocnejšie

Allegro

* podpora aj pre iPhone

Ať se z kódu nepráší!

Juraj Michálek

7. 5. 2012 - FI MUNI - Brno

<http://georgik.sinusgear.com>

<http://twitter.com/sinusgear>