

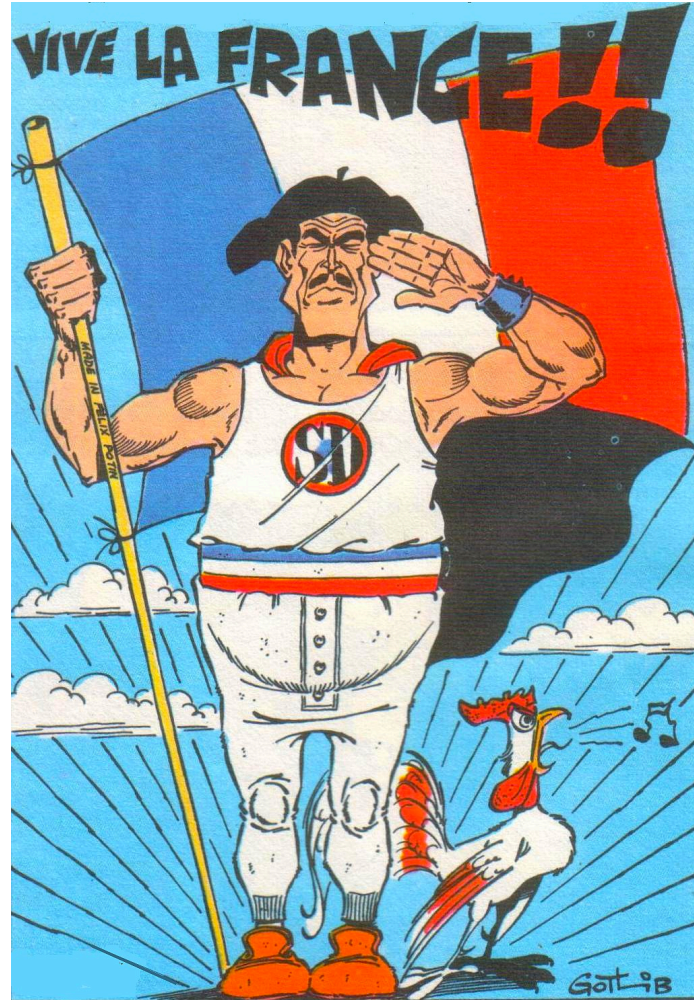
# РЕФАКТОРИНГ ЛЕГАСИ КОДА

@NICOLAS\_FRANKEL



Jpoint

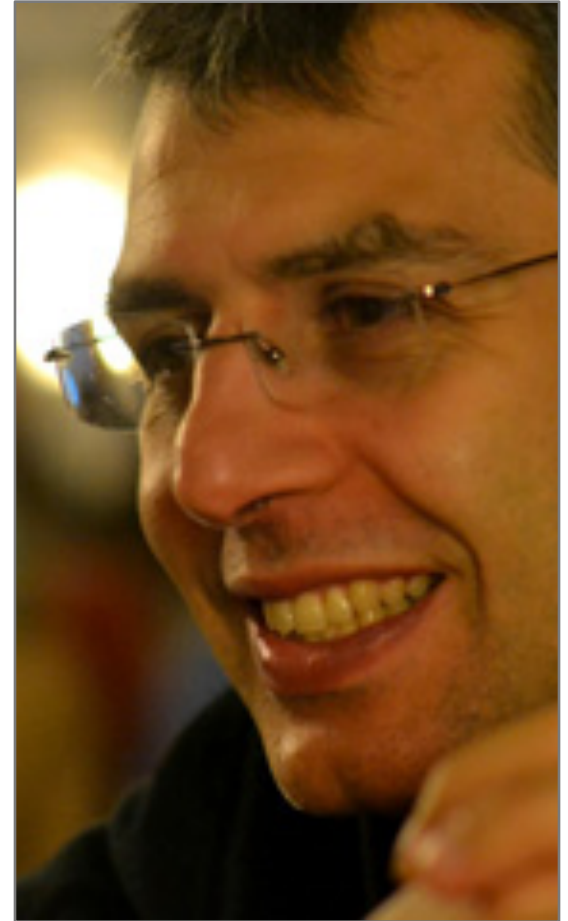
# ПРИВЕТ РЕБЯТА!



# ME, MYSELF AND I

## ➤ **Developer/Software - /Solution Architect**

- Java
- As consultant



# WRITE ONCE, READ MANY

**“Even if you don't intend anybody else to read your code, there's still a very good chance that somebody will have to stare at your code and figure out what it does: That person is probably going to be you, twelve months from now.”**

**MSDN blog**



# CODE IN REAL LIFE



# REFACTORING

**“process of restructuring existing computer code – changing the factoring – without changing its external behavior”**

**-- Wikipedia**



# THE BOY SCOUT RULE

**“Always leave the  
campground cleaner than you  
found it.”**



# OBJECT-ORIENTED PROGRAMMING

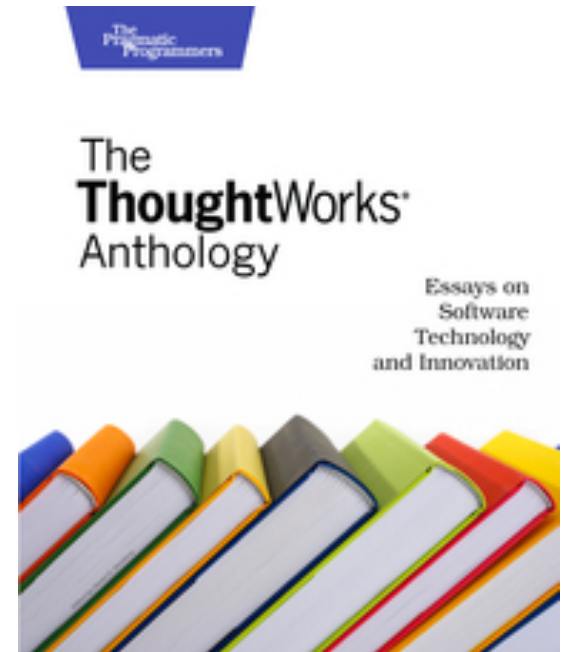
- Really?
- Feels like procedural
  - But in Java

```
window).scrollTop() > header1.css('padding-top') *  
(parseInt(header1.css('padding-top')) *  
header1.css('padding-top'))  
  
{  
header1.css('padding-top', '' +  
window).scrollTop() > header2.css('padding-top') *  
parseInt(header2.css('padding-top')) *  
header2.css('padding-top'))  
  
header2.css('padding-top', '' + he
```



# OBJECT CALISTHENICS

➤ Proposed by Jeff Bay



# OBJECT CALISTHENICS

1. Use only 1 level of indentation per method
2. Don't use `else`
3. Wrap all primitives and strings
4. Use only one `.` per line
5. Don't abbreviate
6. Keep all entities small
7. Don't use any classes with more than 2 instance variables
8. Use first-class collections
9. Don't use any getters/setters/properties



# OUR OWN LEGACY CODE

Rémi Sanlaville  
Johan Martinsson



Time for **DEMO**

# Q&A

 <http://blog.frankel.ch/>

 [@nicolas\\_frankel](https://twitter.com/nicolas_frankel)

 <http://frankel.in/>

<https://git.io/vwUcS>

