Object Calisthenics

October 2016

Peter Kofler, ‘Code Cop’
@codecopkofler
www.code-cop.org

Peter Kofler

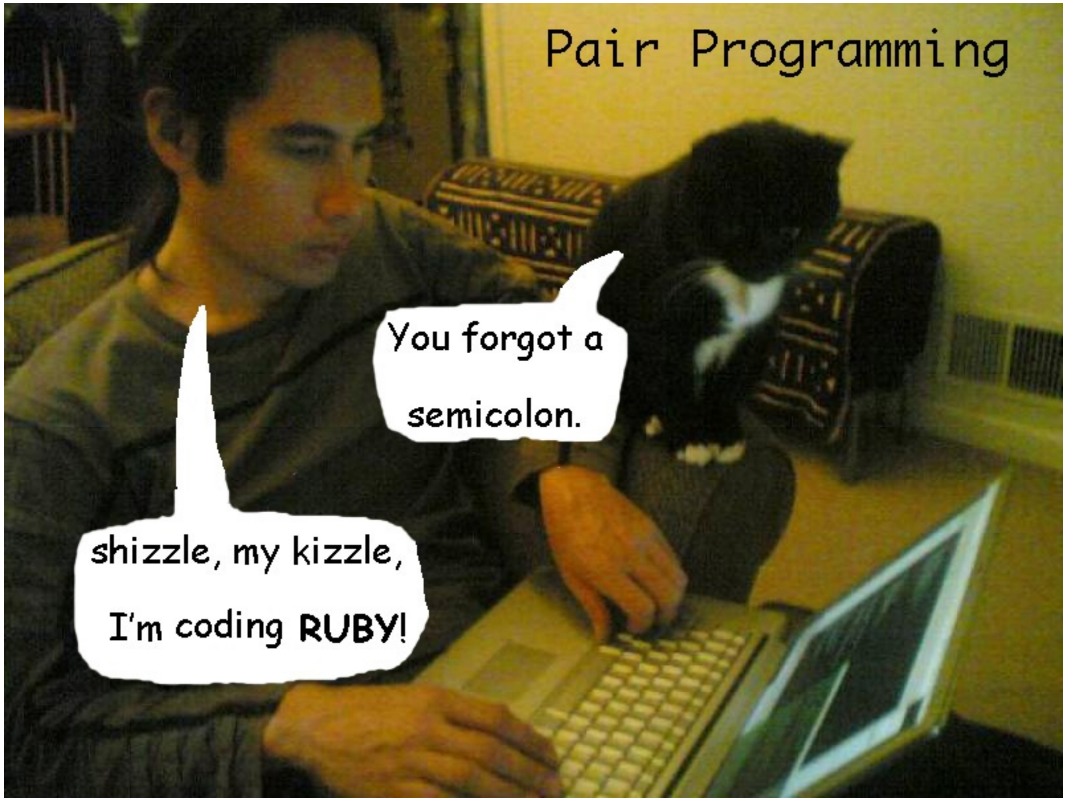
- Ph.D. (Appl. Math.)
- Professional Software Developer for 15+ years
- “fanatic about code quality”
- Freelance Code Mentor

I help development teams with

- Professionalism
- Quality and Productivity
- Continuous Improvement

Mentoring

- Pair Programming
- Programming Workshops
- Deliberate Practice, e.g. Coding Dojos



Developing Quality Software Developers

Agenda

- Concepts of OO
  - Encapsulation
  - Coupling and Cohesion

- Coding Exercise “Object Calisthenics”

Discussion

- What is Object Orientation?
- What are its basic features?

OOP is based on

- Abstraction
- Encapsulation
- Polymorphy
- Inheritance
- Coupling
- Cohesion

Abstraction

- "A class should capture **one and only one** key abstraction."
- Avoid Primitive Obsession.
- Single Responsibility Principle (SRP).

- Wrap All Primitives And Strings. (OC#3)
- First Class Collections. (OC#4)
- Keep All Entities Small. (OC#7)

Encapsulation

- “Objects are defined by what they **do**, not what they contain.“
- All data should be hidden within its class.
- Tell, Don't Ask.

- e.g.
- All fields must be private.
- No Getters/Setters/Properties. (OC#9)

Polymorphy

- Avoid `switch` statements (Type Code).
- “Explicit case on the type of an object (`instanceof`) is usually an error.”
- Open Closed Principle (OCP).

- e.g.
- Avoid `switch` statements.
- Don’t Use The `else` Keyword. (OC#2)

Inheritance

- „Inheritance should only be used to model a specialization hierarchy."
- "IS A ..." vs. „HAS A ..." means the subclass "IS **exactly the same** AS ..."
- Inheritance hierarchies should be shallow. (2-3)
- See Liskov Substitution Principle (LSP).

Composition over Inheritance

Low Coupling

- “Minimize the number of classes with which another class collaborates.“
- “Minimize the number of messages sent between a class and its collaborator.“
- Law of Demeter (“Only talk to friends”).
- See Interface Segregation Principle (ISP).

- e.g.
- One Dot (dereference) Per Line. (OC#5)

High Cohesion

- “Most of the methods defined on a class should be using most of the data members most of the time.“
- Related data and behaviour in one place.

- e.g.
- Don’t Abbreviate (long names). (OC#6)
- Keep All Entities Small. (OC#7)
- Max Two Instance Variables. (OC#8)

Jeff Bay's Object Calisthenics

1. Only One Level Of Indentation Per Method.
2. Don’t Use The `else` Keyword.
3. Wrap All Primitives And Strings.
4. First Class Collections.
5. One Dot/Arrow (dereference) Per Line.
6. Don’t Abbreviate (long names).
7. Keep All Entities Small. (50 LoC per class)
8. Not More Than Two Instance Variables.
9. No Getters/Setters/Properties.

Try it yourself

Coding Dojo Mindset

- Safe place outside work
- We are here to learn
- Need to slow down
- Focus on doing it right
- Collaborative Game

Rules

Pair Programming & TDD

- regular Pair Programming
    - do not talk for too long
    - do not interrupt the other
    - no “keyboard hugging“
- use TDD (or at least “sort of” TDD)
    - write a test before you write code
    - refactor mercilessly
    - no debugger

LCD Numbers

Assignment

- Create an LCD string representation of an integer value using a 4x7 grid of space, minus and pipe characters for each digit.
- Each digit is shown below (dot instead of space)

```
.--......--..--......--..--..--..--..--.
|..|...|...|...||..||...|......||..||..|
|..|...|...|...||..||...|......||..||..|
.........--..--..--..--..--......--..--.
|..|...||......|...|...||..|...||..|...|
|..|...||......|...|...||..|...||..|...|
.--......--..--......--..--......--..--.
```

- Bar size should be adjustable. The default value is 2.

“Calisthenics” Constraint

Jeff Bay's Object Calisthenics

1. Only One Level Of Indentation Per Method.
2. Don’t Use The `else` Keyword.
3. Wrap All Primitives And Strings.
4. First Class Collections.
5. One Dot/Arrow (dereference) Per Line.
6. Don’t Abbreviate (long names).
7. Keep All Entities Small. (50 LoC per class)
8. Not More Than Two Instance Variables.
9. No Getters/Setters/Properties.

Prepare

- Find a pair.
- Choose a programming language.
- Get the Java project or create new one.
- Implement LCD Numbers.

Don't Focus on
Getting it Done.
Focus on Doing
It Perfectly.

→Practice

Closing Circle

- What did you learn today?
- What surprised you today?
- What will you do differently in the future?

Peter Kofler

@codecopkofler

www.code-cop.org

CC Images

- Bruce http://www.flickr.com/photos/sherpas428/4350620602/
- pairing http://www.flickr.com/photos/dav/94735395/
- agenda http://www.flickr.com/photos/24293932@N00/2752221871/
- wants you http://www.flickr.com/photos/shutter/105497713/
- city http://www.flickr.com/photos/42311564@N00/2411037726/
- hands https://www.flickr.com/photos/ninahiironniemi/497993647/
- dojo http://www.flickr.com/photos/49715404@N00/3267627038/
- rule http://www.flickr.com/photos/phunk/4188827473
- LCD https://www.flickr.com/photos/ta3/3275907319/
- calisthenics https://www.flickr.com/photos/snickclunk/18522120656/