

Object Calisthenics

- Exercise (why?)
- About Object Calisthenics
- Exercise

“Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away”

Antoine de Saint-Exupery

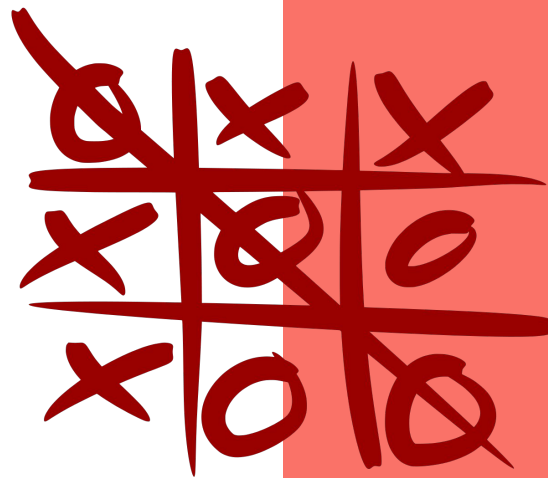
After spending a few sessions in test-driven development, we now turn our attention to design. Why do we need to consider it?

- **TDD is not enough.** TDD will have a positive side effect on design, but simply doing TDD will not take care of design by itself.
- **DRY (Do not Repeat Yourself) is not enough.** So far our refactoring efforts have been focused on removing duplication. This is crucial to write maintainable code, but it is not enough.
- **TDD will punish you if you don't understand design.** When we find ourselves writing very complex test code, we should consider changing the design of the implementation.

Tic Tac Toe

- **Rules**

- X always goes first
- Players alternate placing X's and O's on the board
- Players cannot play on a played position
- A player with 3 X's or 3 O's in a row (vertically, horizontally or diagonally) wins the game.
- If all 9 squares are filled and neither player achieves 3 in a row, the game is a draw.





The word ***calisthenics*** comes from the ancient Greek words *kalos* (κάλλος), which means "beauty" and *sthenos* (σθένος), meaning "strength".

Object Calisthenics rules

- Only one level of indentation per method
- Don't use the ELSE keyword
- Wrap all primitives and strings
- First class collections (wrap all collections)
- Only one dot per line ~~dog.Body.Tail.Wag()~~ => dog.ExpressHappiness()
- No abbreviations
- Keep all entities small
[10 files per package, 50 lines per class, 5 lines per method, 2 arguments per method]
- No classes with more than two instance variables
- No public getters/setters/properties



Tic Tac Toe kata

Rules:

- Players alternate placing X's and O's on a 3x3 board, with X starting first
- Players cannot play on already taken positions
- A player wins when it has three symbols in a row, either
 - Horizontally
 - Vertically
 - Diagonally
- If all squares are taken and there are not three symbols in a row, then the outcome is a draw

Caveat: try to make invalid state unrepresentable

Object Calisthenics



Game of Life kata

The universe of the Game of Life is an infinite two-dimensional grid. Any cell of the grid is in one of two possible states: dead or alive. Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent.

The universe evolves with time in steps.

At each step (or tick), the following transitions occur:

- Any live cell with fewer than two live neighbours dies, as if caused by under-population.
- Any live cell with two or three live neighbours should live on to the next generation.
- Any live cell with more than three live neighbours dies, as if by over-population.
- Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed.

The rules continue to be applied repeatedly to create further generations.