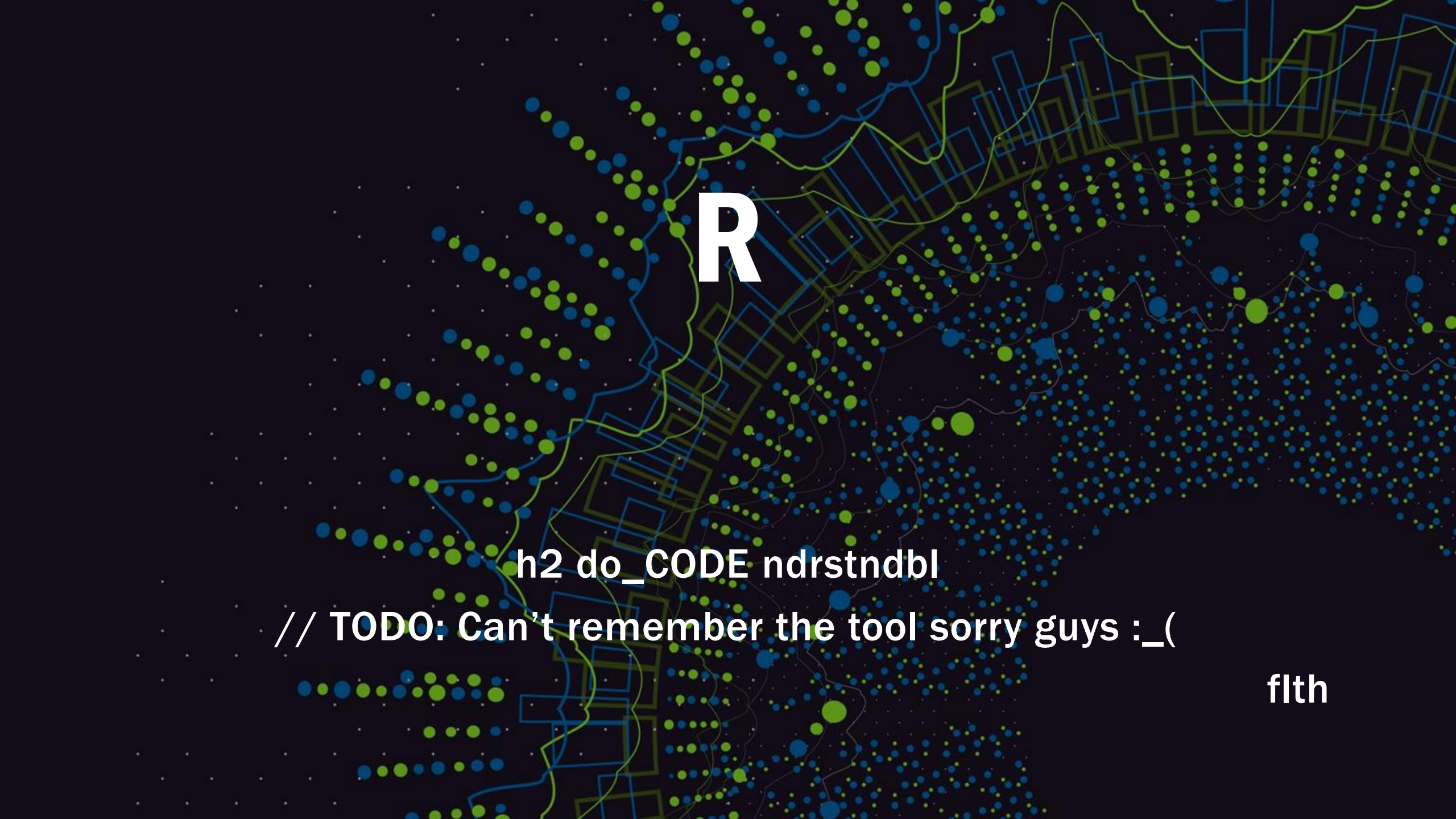
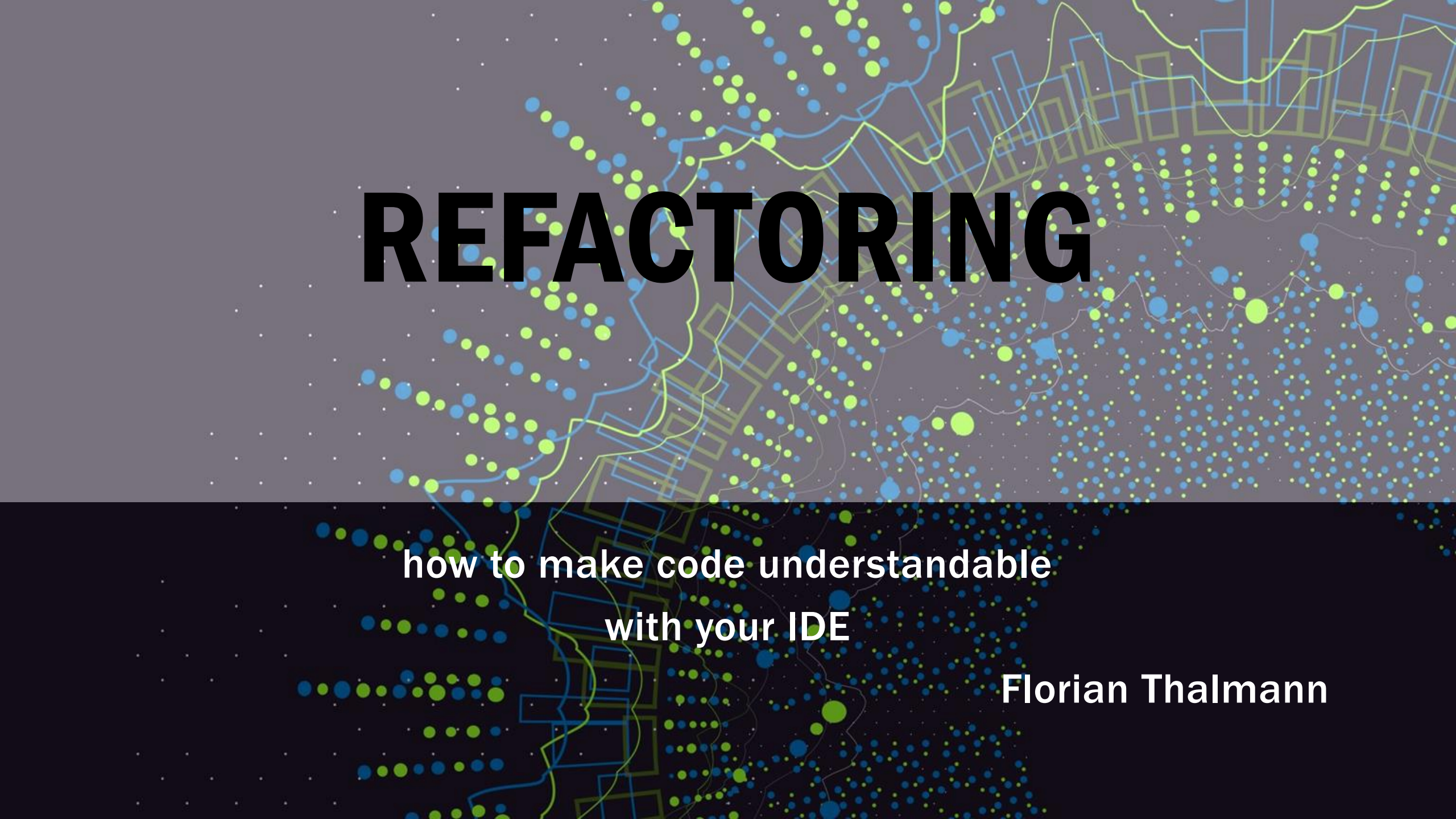
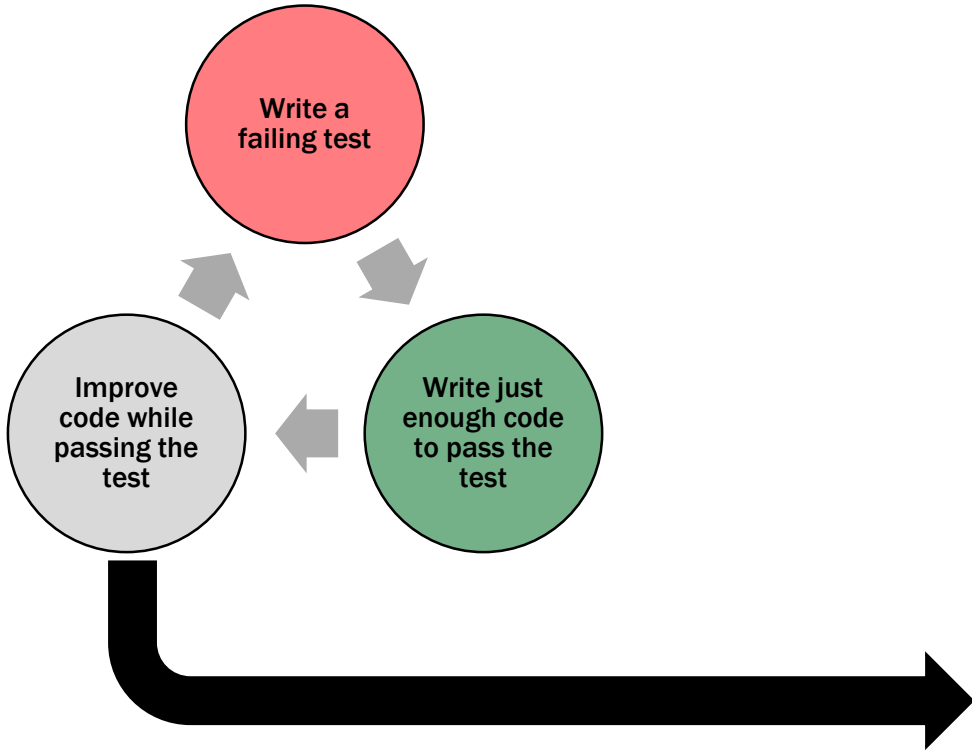
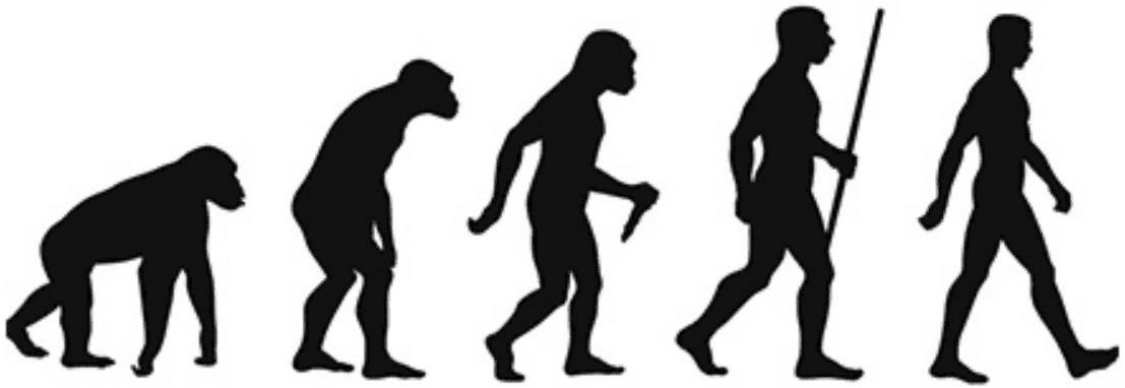

R
h2 do_CODE ndrstndbl
// TODO: Can’t remember the tool sorry guys :_(
flth


REFACTORING
how to make code understandable
with your IDE
Florian Thalmann

WHAT WILL I COVER TODAY





Refactoring

Improving the Design of Existing Code

and Readabilty

https://lvivity.com/what-is-code-refactoring

BUT HOW?

https://forum.rocketbeans.tv/t/beans-on-rice-3-was-soll-gekocht-werden-voting/2434

KATAS

https://pixabay.com/de/vectors/silhouette-karate-k%C3%A4mpfen-stark-37220048/

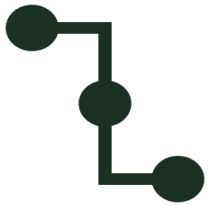
FIRST THINGS FIRST - THEORY

Refactor aggressively and constantly

When?

- Rule of three
- Break Object Calisthenics rules

FIRST THINGS FIRST - IDE



know your SHORTCUTS!

5 main atomic refactors:

Rename

Extract

Inline

Move

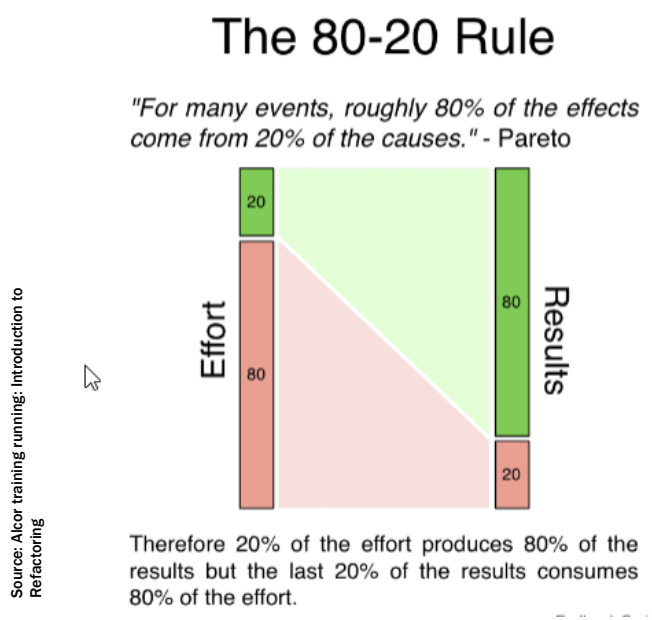
Safe delete

FIRST THINGS FIRST - GUIDELINES

- stay in GREEN!
- if tests are coupled with implementation, refactor tests first
- be strict about staying in green, otherwise let go of something..

FIRST THINGS FIRST - GUIDELINES

- readability before design
- better names for variables methods and classes



REFACTORING 80-20 RULE

80% of the value in refactoring comes from ***improving readability***.

20% of the remaining value comes from ***design change***.

FIRST THINGS FIRST – REFACTOR READABILITY

1. Format
2. Rename
3. Remove
4. Extract
5. Reorder

FIRST THINGS FIRST – REFACTOR DESIGN

- extract private methods from **deep conditionals**
- extract smaller private methods from **long methods**
- encapsulate cryptic code in **private methods**
- **return** from methods as soon as possible
- **encapsulate** where we find missing encapsulation
- **remove duplication**

LETS GO TO THE GYM WITH EMILY

“.. stretch your refactoring muscles and get you to explore your IDE to see what’s really possible using shortcuts and automation.”

https://github.com/emilybache/RefactoringGolf

MY “SOLUTION” FOR ROUND 2

Legend: <points> <command> (<shortcut>) | E = EDIT code manually

2 E -> Type FootballScoreStats() in FootballScoreStats-Constructor (!!COMPILE ERROR!!)
1 let constructor create (ALT ENTER)
1 create field, make final (ALT + ENTER)
2 E -> Change static access of FootballData to Instance Variable
2 E -> Delete static modifier in FootballData.geAllPlayed method
1 extract interface (Refactor -> Extract Interface) Do not let variable names change!
1 extract method (CTRL + ALT + M)
1 inline parameter total (!!COMPILE ERROR!!)
2 E -> initialize variable correct | !!Failing tests!!
4 E -> add '+' for return type | green tests
1 move method to class 'Game' (F6)
1 inline public method (CTRL + ALT + N)
1 inline public method (CTRL + ALT + N)
1 inline public method (CTRL + ALT + N)
1 inline public method (CTRL + ALT + N)

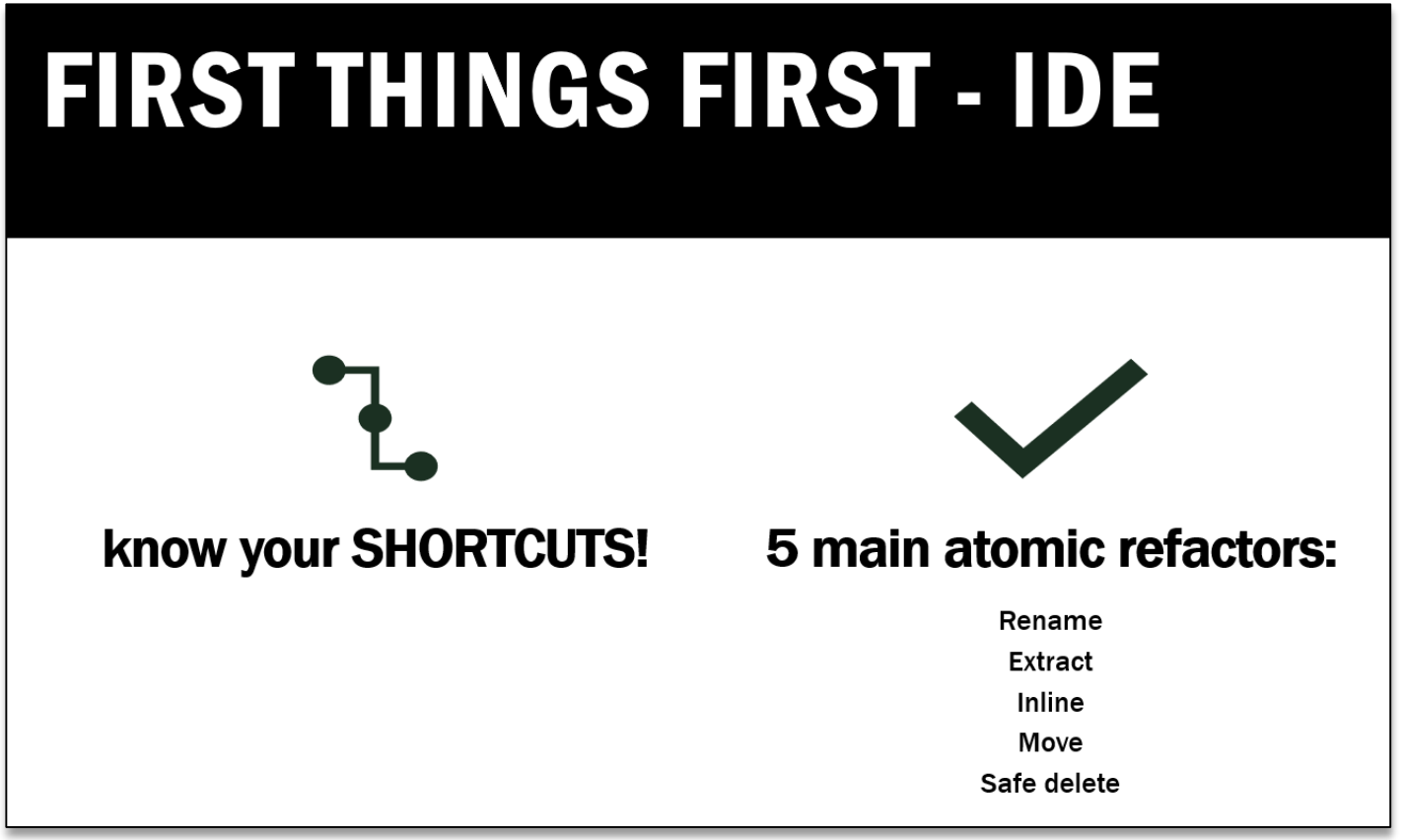
22 -> Can you beat it? :-)

USED SHORTCUTS IN INTELLIJ

- Rename (SHIFT + F6)
- Extract method (CTRL + ALT + M)
- Inline method/param (CTRL + ALT + N)
- Move statements (CTRL + SHIFT + (ARROW UP | DOWN)
  - Move line (CTRL + ALT + (ARROW UP | DOWN)
  - Move members (F6)
- (Save delete (ALT + DELETE))

USED SHORTCUTS IN INTELLIJ

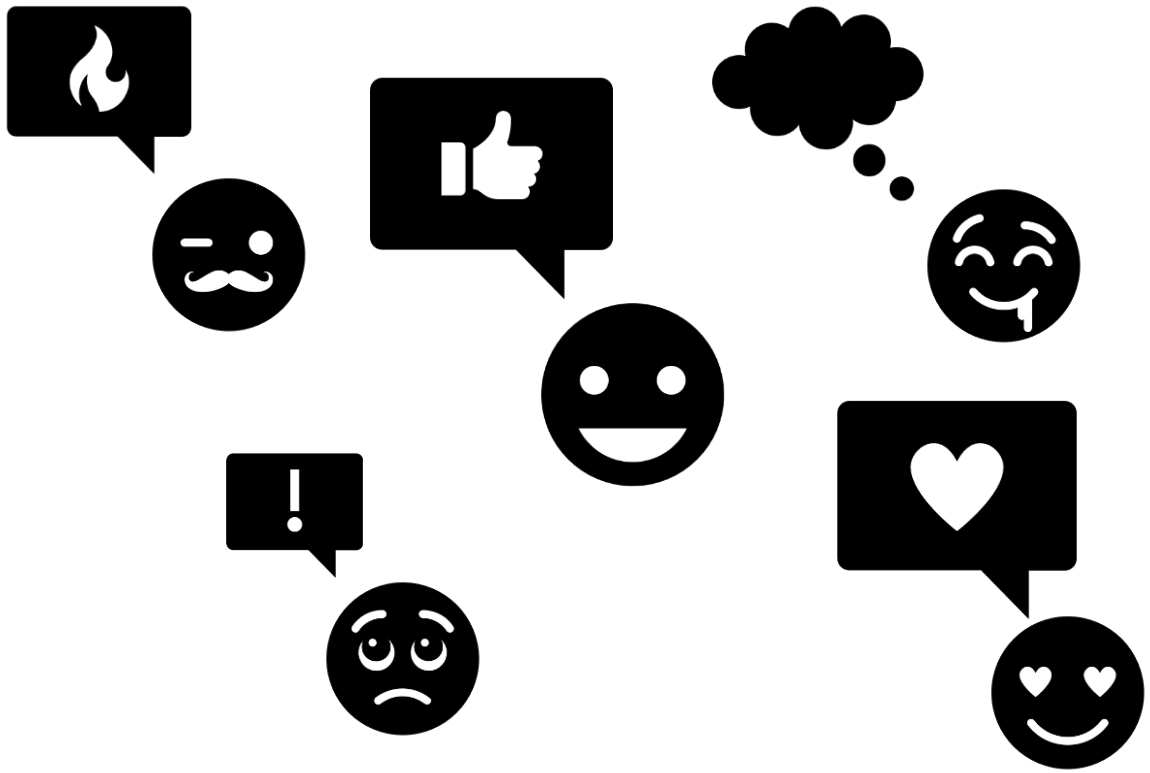
- Rename
- Extract method
- Inline method/param
- Move statements
- Save delete



MORE HANDY SHORTCUTS FOR INTELLIJ

- In test class, run tests (CTRL + SHIFT + F10)
- Rerun last tests (CTRL + F5)
- Generate (ALT + INSERT) -> Can be used nearly everywhere
- When you’re lost (ALT + ENTER)

QUESTIONS - FEEDBACK



florian.thalmann@css.ch

https://www.linkedin.com/in/florian-thalmann-22a68413b/