



# WHAT IS BEHAVIOR?

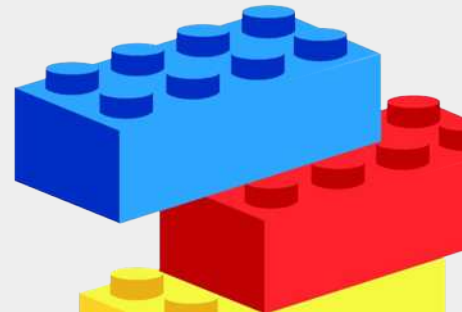
Through the lens of modularity

# A MODULE IS

- A building block
- Composable
- Hides information

i.e.

A **composable** unit with an **interface contract**.



# THE BEHAVIOR IS THE API

# THE API IS THE BEHAVIOR

We decide what exactly goes in the API,  
because *we* write the code \*

We can choose to make that part PUBLIC  
and known, *or...* to HIDE it.

\* Sometimes behavior isn't in the code.



# A SPEC IS

A description of behavior.

A test is a runnable specification.

A test that tests something accidental is unnecessarily bloating the spec.



# OF EVERY TEST/SPEC YOU CAN ASK YOURSELF

Is it essential to the problem? Or is it accidental?

Am I adding to the spec? Or am I subtracting in order to reveal the spec?





# BOWLING KATA

# ARE THE SCORING RULES THE BEHAVIOR?

X X X X X X X X X X X	0 100 100 100 100 100 100 100 100 100 100
5 10 10 10 10 10 10 10 10 10 10	80 (10 frames x 8)
5 15 15 15 15 15 15 15 15 15 15	150 (10 frames x 15)
X 7 10 10 8 8 10 10 10 10 10	167

## Scoring Rule

- Each game, or "line" of bowling, includes ten turns, or "frames" for the bowler.
- In each frame, the bowler gets up to two tries to knock down all ten pins.
- If the first ball in a frame knocks down all ten pins, this is called a "strike". The frame is over. The score for the frame is ten plus the total of the pins knocked down in the next two balls.
- If the second ball in a frame knocks down all ten pins, this is called a "spare". The frame is over. The score for the frame is ten plus the number of pins knocked down in the next ball.
- If, after both balls, there is still at least one of the ten pins standing the score for that frame is simply the total number of pins knocked down in those two balls.
- If you get a spare in the last (10th) frame you get one more bonus ball. If you get a strike in the last (10th) frame you get two more bonus balls.
- These bonus throws are taken as part of the same turn. If a bonus ball knocks down all the pins, the process does not repeat. The bonus balls are only used to calculate the score of the final frame.
- The game score is the total of all frame scores.

Perhaps...

Or, perhaps, it's the set of all known inputs and outputs.

But good luck writing a test for that...



*IMHO the human readable spec is not the final spec, but it does lead you to one.*





## Not part of the “human readable” spec. Should we add it?

```
expect(bowlingScore("--|--|--|--|--|--|--|--|--|--||--")) .toBe(0)
```

“The characters after the || are bonus balls.”

In terms of modularity: We’re “exposing” (no longer hiding) this information.



## Both tests are technically right, but which best models the (total) behavior?

```
expect(bowlingScore("X|--|--|--|--|--|--|--|--|")) .toBe(10)
```

*Too "granular"?*

V.S.

```
expect(bowlingScore("X|11|1-|--|--|--|--|--|--|")) .toBe(15)
```

If the first ball in a frame knocks down all ten pins, this is called a "strike". The frame is over.

The score for the frame is ten plus the total of the pins knocked down in the next two balls.



**DECISION DEFERMENT**

# **IF THE API DOESN'T CHANGE, THE CALLER DOESN'T HAVE TO CHANGE**

**THE HOLY GRAIL OF SOFTWARE DESIGN: EASY CHANGE**

