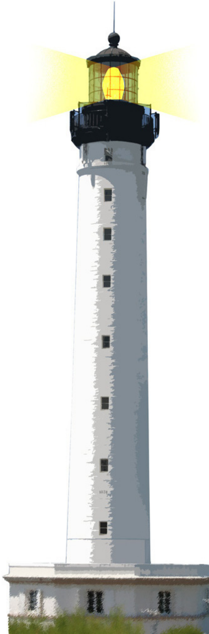
Advanced Object-Oriented Design

About results

The case of Stone Paper Scissors

S. Ducasse



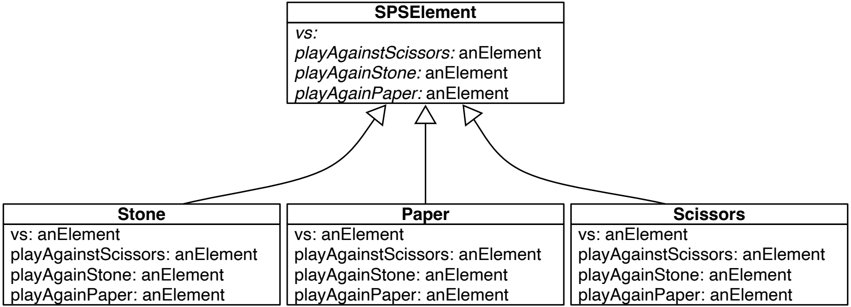
http://www.pharo.org

Goals

- Think about results
- What to do with them

Remember

```
>>> (Stone new vs: Paper new)
#paper
```



SPSElement

vs:
playAgainstScissors: anElement
playAgainStone: anElement
playAgainPaper: anElement

Stone

vs: anElement
playAgainstScissors: anElement
playAgainStone: anElement
playAgainPaper: anElement

Paper

vs: anElement
playAgainstScissors: anElement
playAgainStone: anElement
playAgainPaper: anElement

Scissors

vs: anElement
playAgainstScissors: anElement
playAgainStone: anElement
playAgainPaper: anElement

Returning numbers

```
StonePaperScissorsTest >> testPaperIsWinning
  self assert: (Stone new vs: Paper new) equals: −1
```

```
StonePaperScissorsTest >> testStoneAgainsStone
  self assert: (Stone new vs: Stone new) equals: 0
```

```
StonePaperScissorsTest >> testStoneIsWinning
  self assert: (Stone new vs: Scissors new) equals: 1
```

Returning number analysis

- We have to know the call to know how to interpret the number.
- Here paper is winning but it got different results!

```
>>> Stone new vs: Paper new
−1

>>> Paper new vs: Stone new
1
```

Returning symbols analysis

```
>>> Stone new vs: Paper new
#paper

>>> Paper new vs: Stone new
#paper
```

- With symbols we know who won.
- We will have to compare to do something

```
(aPlayer vs: anotherPlay) = #draw
  ifTrue: [ ... ]
```

Alternate solution

```
Paper new vs: Paper new
  onDraw: [ Game incrementDraw ]
  onReceiverWin: [ ]
  onReceiverLose: [ ]
```

```
Paper >> playAgainstStone: aStone
  onDraw: aDrawerBlock
  onReceiverWin: aWinnerBlock
  onReceiverLose: aLoserBlock

  ^ aWinnerBlock value
```

It feels that there are too many parameters.

Simply with objects

```
Paper new vs: Paper new withResultHandler: ResultDisplayer new
```

```
Paper >> playAgainstStone: aStone withResultHandler: aResultHandler
  ^ aResultHandler paperWon
```

- It feels like the "Do not not ask tell"
- We can have different result handlers

Stepping back

- Symbols looks better than numbers for returning information
- Creating result handlers can be heavy
- Context and scenarios often give you information to see
  - where to invest
  - what is worth to support reuse

Conclusion

Objects are

- Powerful
- Modular
- Just send a message :)

A course by

S. Ducasse, G. Polito, and Pablo Tesone