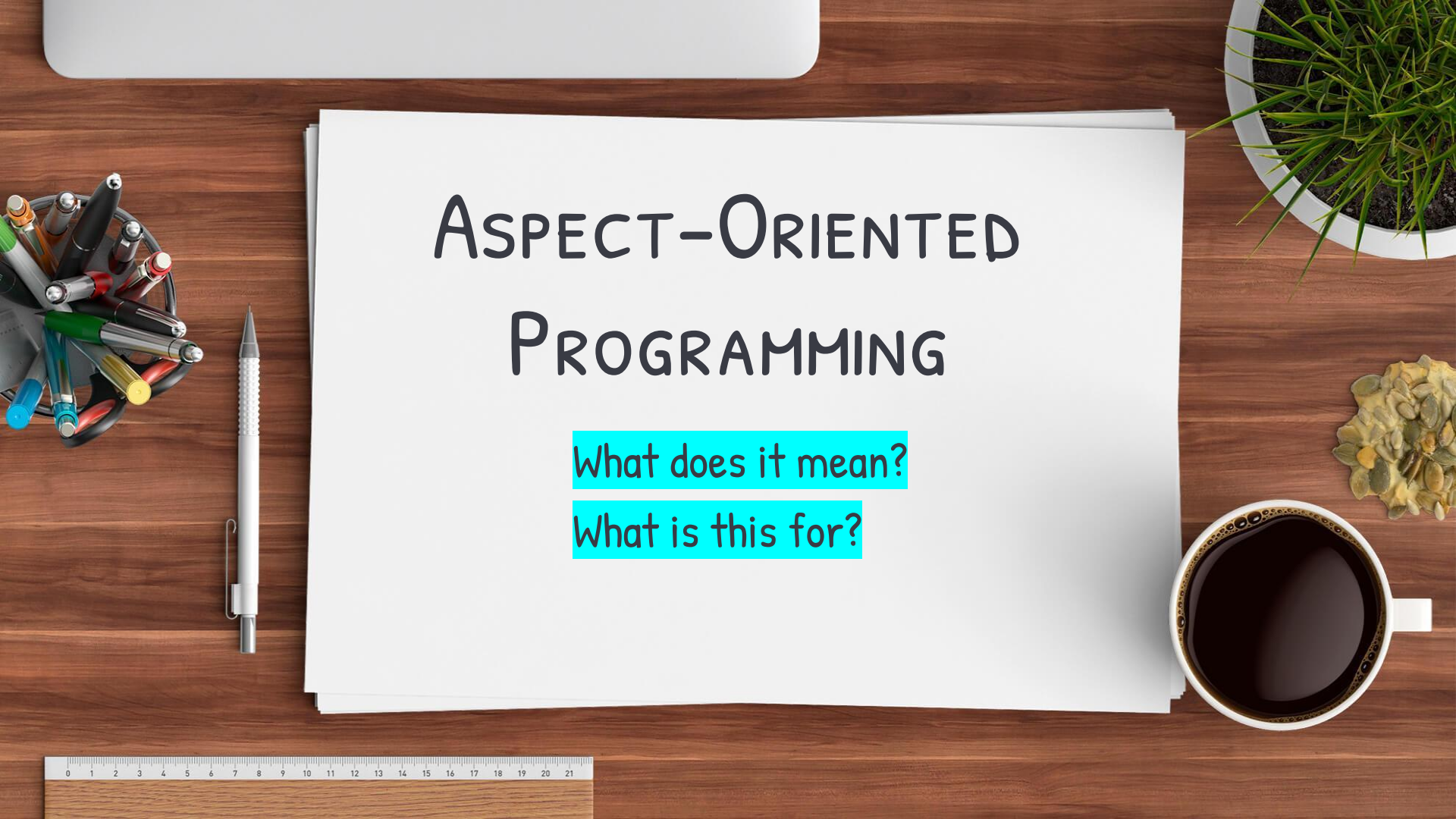
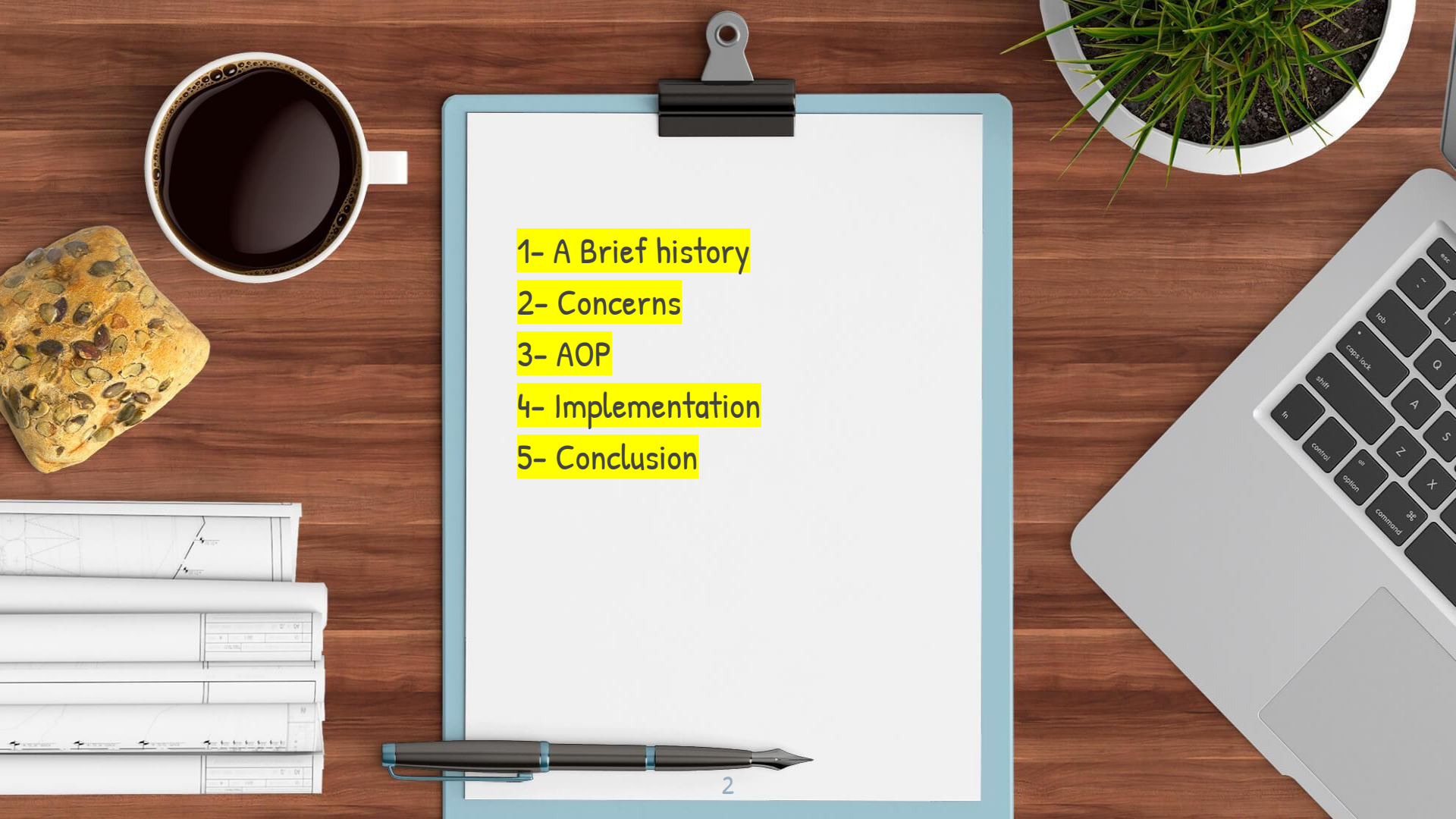

Aspect-Oriented Programming
What does it mean?
What is this for?


1- A Brief history
2- Concerns
3- AOP
4- Implementation
5- Conclusion

A Brief History

Machine Languages

0101110101010101.....

Needs an operators

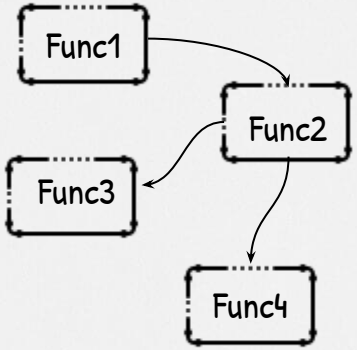
Which is really hard.

Higher level Programming paradigms

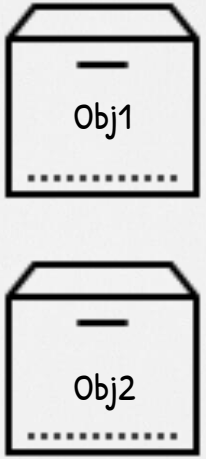
Result in Structured Programming

POP

(Functional Programming)



OOP



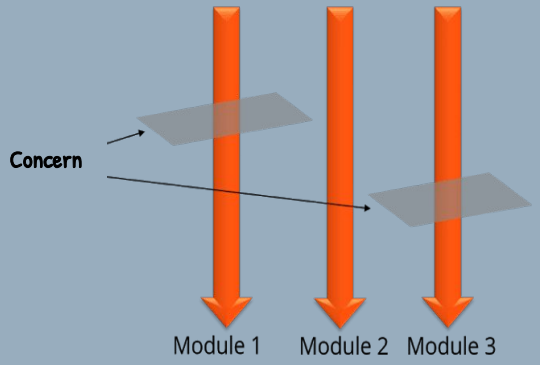
Concerns

In short, a concern is a particular issue, concept, or area of interest for an application: typically a goal the application must meet.

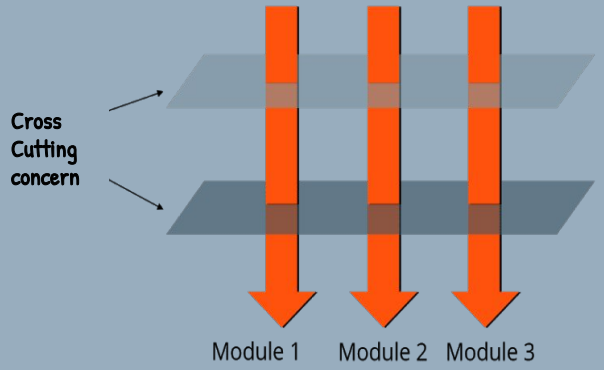
We have two main types of concerns:

- **Core Concerns :** related to functionality of primary requirements like Business logics,...
- **Cross-cutting Concerns :** Cross cutting concerns are concerns that cut across multiple application modules. The goal is to respond to secondary requirements. Like transactions, logging, security,.....

Concern



Cross-Cutting Concern

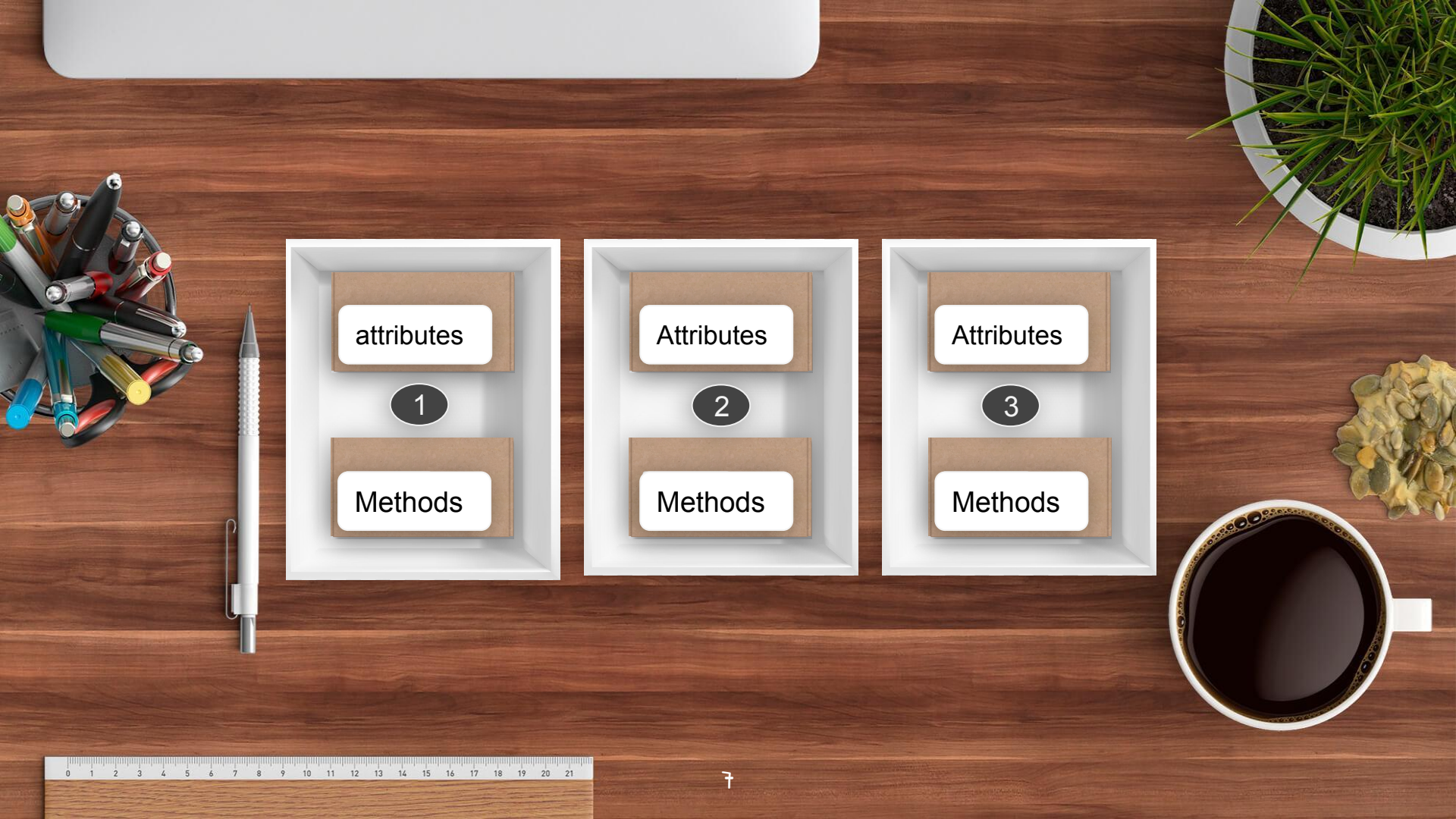


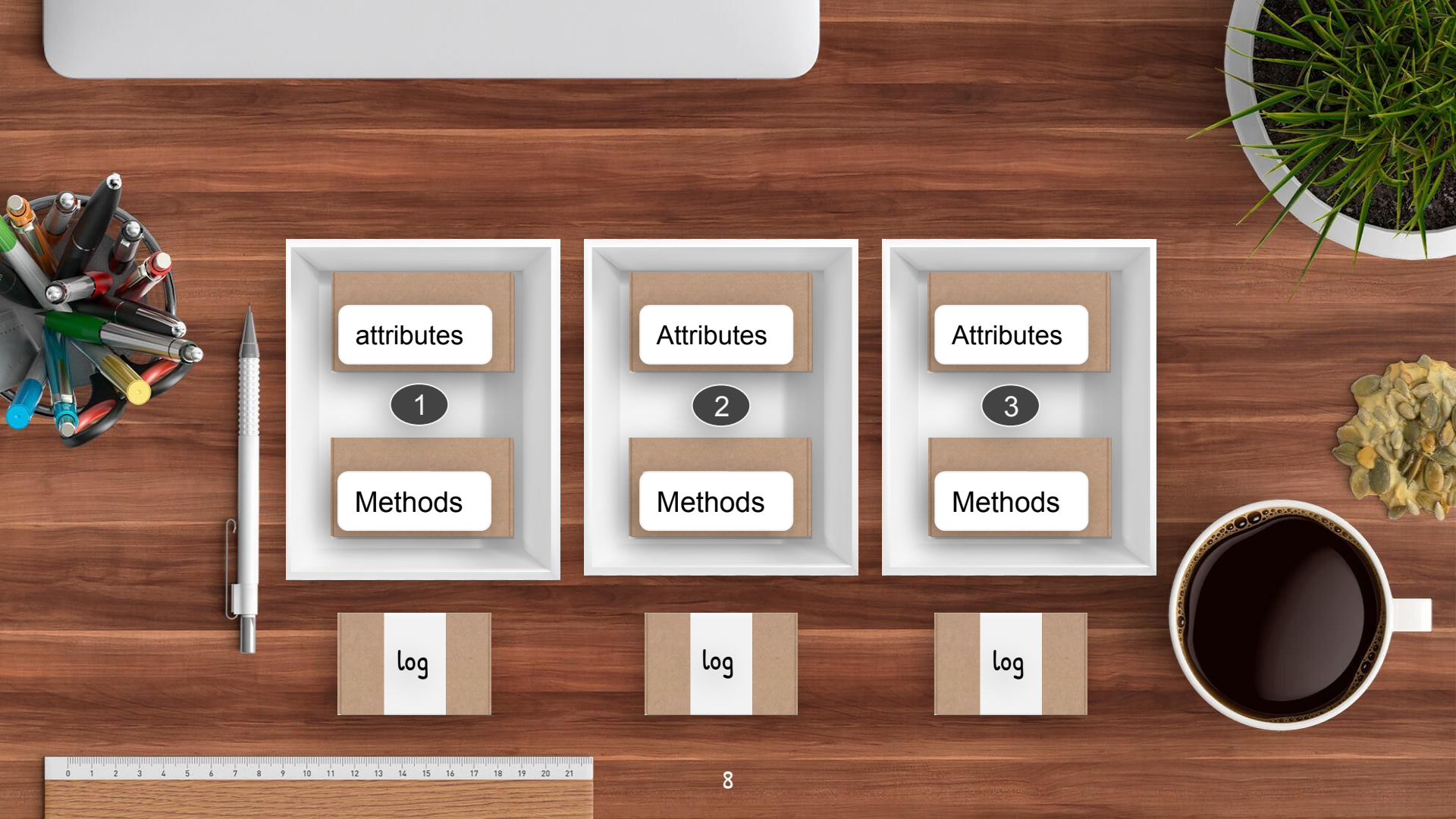
Example:
Consider we have three Object from different classes with their own Methods and Attributes. The goal is to log in one or more method(s) of each objects.

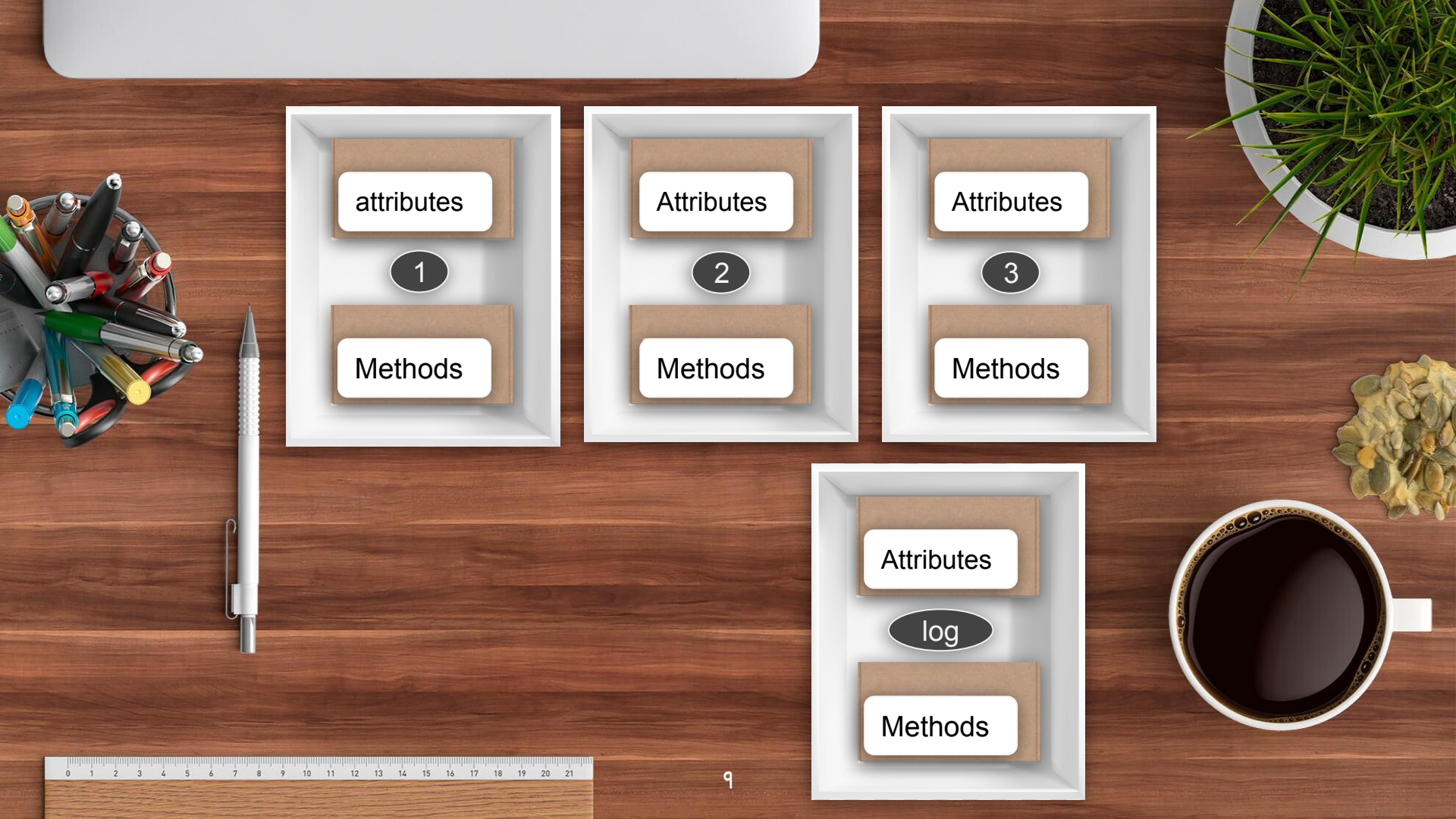
First solution : write code for each of the methods.
Which leads to duplicate of codes.

Second Solution : make a separated class for log.
Have some Downsides as well.

Real solution ⟶ AOP


attributes
1
Methods
Attributes
2
Methods
Attributes
3
Methods


attributes
1
Methods
Attributes
2
Methods
Attributes
3
Methods
log
log
log


attributes
1
Methods
Attributes
2
Methods
Attributes
3
Methods
Attributes
log
Methods

DOWNSIDES

- Too many relationships to CrossCutting object
- Code is still repeated in all methods
- Cannot all be changed at once

So What's The Solution?

AOP

AOP is a programming approach that aims to solve crosscutting concerns throughout better modularization of the code.

It is often defined as a programming technique that promotes separation of crosscutting concerns with in a software system.

SOC


attributes
1
Methods
Attributes
2
Methods
Attributes
3
Methods
Methods
Log Aspect

- **Aspect:** is a modularized implementation for a crosscutting concern. It amalgamates the distributed code that of a crosscutting concern in one module.

- **Join point:** A point during the execution of a program, such as the execution of a method or the handling of an exception.

- **Advice:** Action taken by an aspect at a particular joins point. Different types of advice include "around," "before" and "after" advice.

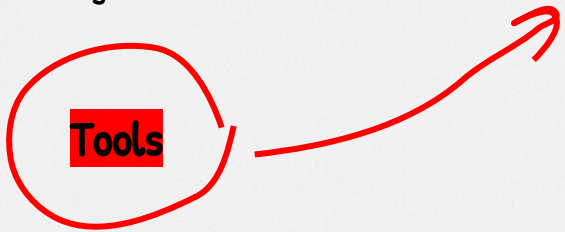
- **Point cut:** A predicate that matches join points. Advice is associated with a pointcut expression and runs at any join point matched by the pointcut (for example, the execution of a method with a certain name).

Implementation

The implementation steps:

1- modularize Cross-Cutting Concerns

2- Configure the advice to our Classes



AspectJ : it's a Java extension which can be downloaded a Java software development kit (SDK).

There are some Frameworks and Application servers to support AOP on already existed programming language like Spring.

Conclusion

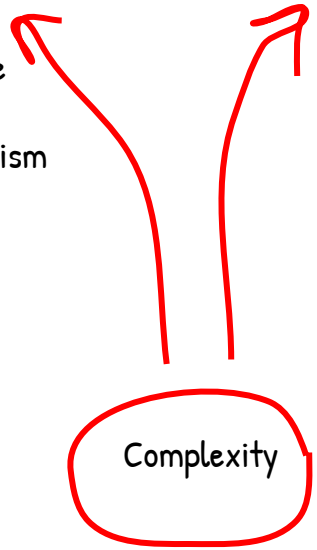
Pros.

1- Smaller Code Size

2- Language Mechanism

3- Evolvability

4- Modularity



Cons.

1- making the debugging process harder and requiring

2- more understanding of the core module and crosscutting concerns implementation.

REFERENCES

1- G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C.Videira Lopes, J.-M. Loingtier, J. Irwin, "Aspect Oriented Programming", In Proc. Europ. Conf. on Object-Oriented Prog.(ECOOP), Finnland, Springer Verlag LNCS 1241, June 1997.

2- Heba A. Kurdi, "Review on Aspect Oriented Programming",(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 9, 2013

3-https://medium.com/@mithunsasidharan/aspect-oriented-programming-overview-6c03235e666a

CONTACT ME :

Give me feedback, i'll appreciate it so much.

Have a Nice Day!

github.com/Benyamin0700

benjaminghahremani@gmail.com

benyamin-ghahremani-nejad

@BenyaminGhahremani