

Când vreau să ...		, folosesc ...	, iar în cod...	13 minute
create and populate an object in a readable way	a	Adapter	1	My subclass must behave exactly as the super class
I need a logic similar to what I've already implemented	b	i Abstract Factory 3	2	I write each algorithm as an implementation of a common interface
leave subclasses to define certain steps of an algorithm	c	Fluent Builder	3	I define an abstract method createChair():Chair (Chair is an interface)
choose at runtime one of the algorithms in a family	d	Proxy	4	My function will receive as a parameter another function !
I subclass correctly	e	Strategy	5	Method Chaining : methods that set fields and always return this
Avoid adding Patterns and other Complications from Day1. YAGNI!	f	Template Method	6	I expose beautiful methods that transform the calls and delegate to the wrapped instance
Call a very UGLY external class/system from the core logic of my application	g	Pass-a-Block	7	In a class/method I solve just one task, I don't pile everything together
I design simple components, easy to understand and reuse	h	SRP	8	I decompose existing logic in smaller methods and call them such I achieve what I want
decouple the creation of a family of objects	i	Liskov	9	I write the generic algorithm in a concrete method that invokes abstract methods
intermediate interactions with an object (for example, to do AOP)	j	DRY	10	I look the simplest implementation possible. I refactor.
execute arbitrary bits of code in my method	k	KISS	11	I give my client a surrogate/replacement that implements the interface of the subject to call