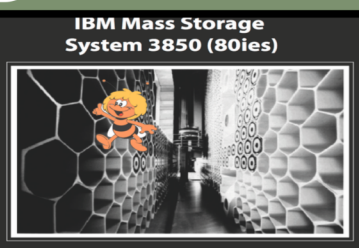
# REPOsitory r1

https://2hmorvqnlhw
yrhvl6dwv4jqgnvhlaf
2wxnrbmbhrbpmx5x
3qqiwprjyd.onion/free/

https://0x8.ch/HackersCardgame3

# VINTAGE COMPUTING



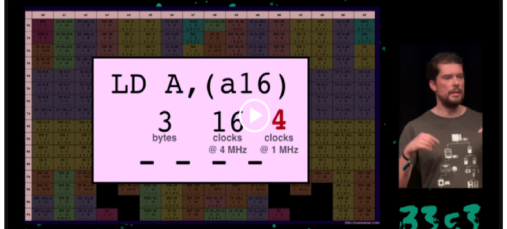

https://0x8.ch/HackersCardgame3 ibm1

**Inventor:** xxxxx **Year:** ~????

If one would find the older people that bought eg. an IBM System 3850 in the early days of modern computing, we could maybe find out who did all this bad things that happens in the world today. Psychological Profile of such a person would match to rich people and therefor maybe also not the nicest people. But remember this would only be the first guess. If i would be rich i would give all my money to save the world, since there are still many problems on this world i bet i am not completely wrong with my first guess.

Could we find out who bought this sort of mass storage or other expensive² computer equipment in the old days??? unknown: IBM had a "maybe" a few

# C64 ASSEMBLER




https://0x8.ch/HackersCardgame3 c64

**Inventor:** xxxxx **Year:** ~????

Assembler Coding...
https://media.ccc.de/v/25c3-2874-en-the_ultimate_commodore_64_talk

Even better:
https://media.ccc.de/v/33c3-8029-the_ultimate_game_boy_talk

Calling it "Ultimate Talk" is ultimately correct, WOW really fu**** great work

To learn Assembler and Computer Architecture the easy and funny way... with smooth SID-like sound :)

# DESIGN PATTERNS

| | | |
|---|---|---|
| C Abstract Factory | S Facade | S Proxy |
| S Adapter | C Factory Method | B Observer |
| S Bridge | S Flyweight | C Singleton |
| C Builder | B Interpreter | B State |
| B Chain of Responsibility | B Iterator | B Strategy |
| B Command | B Mediator | B Template Method |
| S Composite | B Memento | B Visitor |
| S Decorator | C Prototype | |

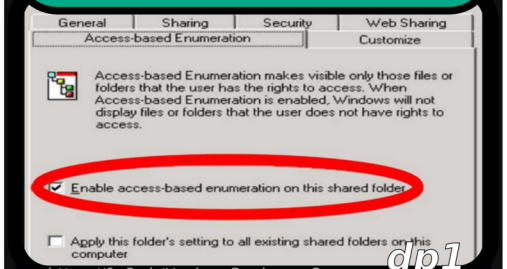
https://0x8.ch/HackersCardgame3 dp3

**Inventor:** Architect Christopher Alexander **~1977**

Design patterns are proven solution templates for recurring design problems in architecture as well as in software development.
A design pattern usually contains several classes (templates or objects). A single design pattern is a collection of different such templates. E.g. The shape of a single step of stairs AND the railing (steelwork or eg woodwork) for the complete stairs AND for example the shape for the carpet of each step.

Architecture, Software development, psychology (Archetypes), Sociology (Archetype-Combinations)

# Access Based Enummeration

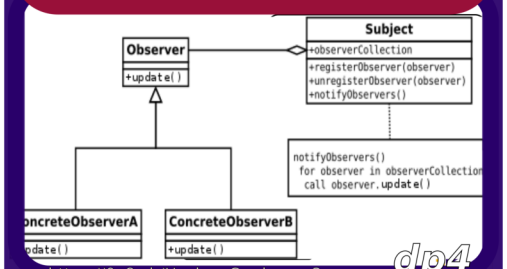



https://0x8.ch/HackersCardgame3 dp1

**Inventor:** CIA? **Year:** ~????

Depending on the authorization (or the security clearance) of a user, folders are shown or not shown in a Windows Netzwork. The same principle maybe applies to websites, youtube, facebook, twitter and maybe even github... if you look at the m50 card the same system "could" be used to split societi into two+ separate groups, and there would be even some triggers known to some of the older groups to start a civil war between this two groups.
Maybe explaining **Access Based Ennumeration++** to the society would deescalate this problem a bit.
So update your people with this knowledge. I learned about Access Based Ennumeration at ABB Technikerschule Baden, hopefully only the students not yet joined the dark side, some of thier parents... unfortunately did. SELECT * FROM ABBTS < 1970.

# Observer Pattern




https://0x8.ch/HackersCardgame3 dp4

**Inventor:** Erich Gamma, Richard Helm ... **~1994**

The observer pattern (also called listener pattern) is a design pattern in software development. It belongs to the category of behavior patterns.
Some people would like that the non techie people think that "observer pattern" means that all the IT and Computer Science People are doing illegal Surveillance and even work for the NSA / CIA. But in fact the term Observer Pattern has nothing to do with surveillance. But of course this design pattern could also be abused in illegal surveillance. (The strange thing is that this colored part was deleted in my repo) So another way to isolate gifted computer science people from the people gifted in other things.

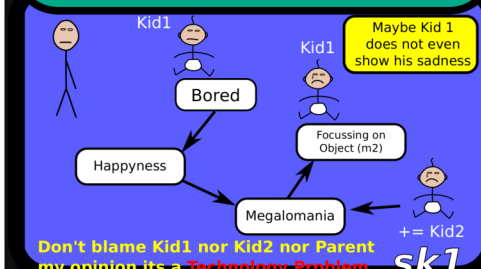
# CLASS

https://0x8.ch/HackersCardgame3 dp2

**Inventor:** Alan Kay **Year:** ~1993

The inventor of the programming language Smalltalk was the first who used objects in software development, But Gutenberg and the letters of his printing presses was another example of templates. A class can be a single template for part of a complete software, e.g. "Name, address, location, phone, ..." would be a template for a person in an address database.
Simplification/standardization through "pre- printed (software) templates", so in cyberwar (especially when they use things like t1 card) after a while you would recognize the general behavior of the class or the design-pattern (dp4) that makes it much easier to defend yourself against such attacks.

# Statemachine (bad for kids)




sk1

**Inventor:** xxxxx **Year:** ~????

1. Kid 1 is bored (therefore doing things random things to stop the boredom)
2. Parent 1 spends time with Kid 1
3. Kid 1 stops with random things and is lucky that Parent 1 plays with him / her.
4. Kid 2 thinks it would be existentially to be part of this group (Kid 1 & Parent 1) too
5. Parent 1 plays with Kid 2 too
6. Kid1 now thinks that it is a thread to his existence
(7. maybe therefore Kid1 is begin to be evil to Kid2)
8. Parent 1 tries to make them play with each other or as a group of three
9. **technology** (t* card) makes Kid1 focus on Objects (m2 card) and so the **problem** stays unsolved
10a. Kid1 & Kid2 dont learn to 8. -> goto 1
10b. Aditionally the parent(s) would be scapegoats for this possibly technological induced problem, eg. b1 && t1 Card.
11. by the age of 20+ one would play cards mos2⇒m37⇒dt1??