



## FRUIZIONE E UTILIZZO DEI MATERIALI DIDATTICI

- ➡ **E' vietata** la copia, la rielaborazione, la riproduzione dei contenuti e immagini presenti nelle lezioni in qualsiasi forma
- ➡ **E' inoltre vietata** la diffusione, la redistribuzione e la pubblicazione dei contenuti e immagini, incluse le registrazioni delle videolezioni con qualsiasi modalità e mezzo non autorizzati espressamente dall'autore o da Unica

# DESIGN PATTERNS

# DESIGN PATTERNS

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice." <sup>(1)</sup>

The idea was adopted by software engineering  
**Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John (GoF) - Design Patterns: Elements of Reusable Object-Oriented Software (1994)**

---

<sup>(1)</sup> Christopher Alexander, "A Pattern Language", 1977. [https://en.wikipedia.org/wiki/Christopher\\_Alexander](https://en.wikipedia.org/wiki/Christopher_Alexander)

## A pattern has four essential elements:

1. The **pattern name** is a handle we can use to describe a design problem, its solutions, and consequences in a word or two. It makes it easier to think about designs and to communicate them and their trade-offs to others.
2. The **problem** describes when to apply the pattern. It explains the problem and its context.
3. The **solution** describes the elements that make up the design, their relationships, responsibilities, and collaborations. **The solution doesn't describe a particular concrete design or implementation**, because a pattern is like a template that can be applied in many different situations. Instead, the pattern provides an abstract description of a design problem and how a general arrangement of elements (classes and objects in our case) solves it.
4. The **consequences** are the results and trade-offs of applying the pattern.

# Categories of patterns

- Architectural patterns
- Design patterns
- Language level patterns. This is the lowest level of the pattern-categories, also known as idioms

# Idioms

# repeat n times

```
n = 5
i = 0
while i < n:
    i = i + 1
    print("DONE")
```

# stop when input == 0

```
for _ in iter(int, 1):
    v = int(input())
    if v == 0:
        break
```

```
//c
for(;;){
    // input
    if (condition){break;}
}
```

# repeat n times

```
n = 5
for _ in range(n):
    print("DONE")
```

# stop when input == 0

```
flag = True
print('0-> exit')
while (flag):
    v = int(input())
    flag = v != 0
```