Patterns Overview
Observer

* Deferred (abstract)
+ Effective (implemented)

Client (uses)

Inherits from
Command

execute* undo*

* COMMAND

DELETION

execute* undo* line index

INSERTION

execute* undo* index

...
Visitor

VEHICLE

\( \ast \)

TAXI

\( \ast \)

BUS

\( \ast \)

VISITOR

\( \ast \)

MAINTENANCE_VISITOR

\( \ast \)

SCHEDULE_VISITOR

\( v \cdot T\_visit\) (Current)

accept\(^{+}\)

visit_taksi\(^{+}\)

visit_bus\(^{+}\)

visit_taxi\(^{+}\)

visit_bus\(^{+}\)

visit_bus\(^{+}\)

visit_bus\(^{+}\)

accept\(^{*}\)

accept\(^{+}\)

accept\(^{+}\)
Strategy

+ CONTEXT

do_something

strategy

+ STRATEGY_A

do_something+

+ STRATEGY_B

do_something+

+ STRATEGY_C

do_something+
Chain of responsibility

APPLICATION → HANDLER[\*G] → INTERMEDIATE_HANDLER[\*G] → FINAL_HANDLER[\*G]

- can_handle+
- do_handle+
- next
- handle
- can_handle* 
- do_handle* 
- handled
- set_next
State

CONTEXT

+ state

INITIAL_STATE

INTERMEDIATE_STATE

FINAL_STATE

* context

do_something*

do_something

do_something+

do_something+

do_something+
Factory Method

```
* PRODUCT
  + CONCRETE_PRODUCT

* CREATOR
  + CONCRETE_CREATOR

new_product* new_product+
```
Abstract Factory

PRODUCT_A

PRODUCT_A1

PRODUCT_A2

FACTORY_1

FACTORY_2

PRODUCT_B

PRODUCT_B1

PRODUCT_B2

FACTORY

new_product_a*

new_product_a+

new_product_b*

new_product_b+

new_product_a*

new_product_a+
Builder

CLIENT

* my_builder

build

BUilder

build*

last_product*

MY_BUILDER

+ last_product+

build+

build_product

build_part_a

build_part_b

MY_PRODUCT

set_part_a

set_part_b

PART_A

part_a

PART_B

part_b
Singleton

**Shared Singleton**

Global point of access

**Instance**

**Singleton**
Bridge

APPLICATION

impl

IMPLEMENTATION

APP

APP

APP

APP

perform

perform*

perform+

perform+
Composite

Transparency version

Safety version
Decorator

COMPONENT

+ CONCRETE_COMPONENT
  - do_something+

+ DECORATED_COMPONENT
  - do_something+

+ CONCRETE_DECORATED_COMPONENT_1
  - do_something+

+ CONCRETE_DECORATED_COMPONENT_2
  - do_something+
Façade

CLIENT

FACADE

internal
Flyweight

- **FLYWEIGHT_FACTORY**: produces flyweights
- **FLYWEIGHT**: represents shared state
- **CLIENT**: uses flyweights
- **SHARED_FLYWEIGHT**: shared state
- **UNSHARED_FLYWEIGHT**: unshared state

- `new_flyweight`
- `flyweights`
- `intrinsic_state`
- `perform+`
- `entire_state`
- `perform+`