Singleton-> cand iti zice sa ai o singura instanta cv de genul

Factory simplu -> iti zice de enum si familie de obiecte

Factory method -> iti zice sa nu modifici codul si familie de obiecte

Prototype -> iti spune sa folosesti o instanta deja creata

Builder -> ai multe optiuni si ti zice ca unele s optionale si ai valori default sau false

SINGLETON

- Sintagma cheie pentru utilizarea acestui design pattern este: instanță unică sau o singură instanță.

- Corelatii

Factory – o singură fabrică de o obiecte.

Builder – un singur obiect de construit alte obiecte.

FACTORY

- Sintagma după care este recunoscut este: familie de obiete sau obiecte din aceeași familie.

- Usecase: Existența unei familii de obiecte într-o aplicație

- Corelatii

Singleton – fabrica poate fi unică.

Composite – prin intermediul fabricii sunt create nodurile container și nodurile frunză

FACTORY METHOD (Virtual Constructor)

- Este asemănător cu Simple Factory doar că nu mai folosește enum ci abstractizează nivelul de creare

- Sintagma cheie, ca la FACTORY doar ca apare !!!Să nu se utilizeze tipuri enum!!!

PROTOTYPE

- Sintagma recunoastere: te folosesti de un obiect deja creat / durează foarte mult sau consumă resurse foarte multe

- Usecase: Atunci când obiectele create seamănă între ele, iar crearea unui obiect durează foarte mult sau consumă resurse foarte multe.

- Corelatii

Factory – se poate stoca o serie de obiecte și să fie clonate atunci când sunt cerute obiectele de acel tip;

Decorator – se clonează obiectele și apoi se modifică;

Composite – elementele de pe același nivel pot fi clonate.

BUILDER

- Usecase: În general pentru construirea de obiecte complexe cu foarte multe atribute

- Corelatii

Singleton – Clasa Builder poate fi singleton, astfel încât, construirea de obiecte să fie centralizată.