



## SINGLETON

# WHEN?

VREI SĂ AI O SINGURĂ  
INSTANȚĂ A UNUI OBIECT

# HOW?

**Pas 1:** Alegem clasa pe care vrem  
să aplicăm Singleton

**Pas2:** Îi punem un câmp static  
instance

**Pas3:** Îi facem constructorul  
privat

**Pas3:** Adăugăm o metodă statică  
prin care să obținem instanța

**Pas4:** La instanțiere folosim  
metoda getInstance a clasei

Exemplu implementare

## VISITOR

# WHEN?

VREI SA FACI O OPERATIE PE  
MAI MULTE OBIECTE  
INDIFERENT DE TIPUL LOR  
(inlocuitor pentru instanceof)

# HOW?

**Pas 1:** Creăm o interfață care  
reprezintă elementul, care conține  
o metoda accept(Visitor visitor)

**Pas2:** Creăm mai multe clase care  
implementează interfața de mai  
sus (suprascrim metoda accept)

**Pas3:** Creăm interfața Visitor care  
conține metode de visit pentru  
fiecare tip de obiect

**Pas4:** Creăm o clasă concretă care  
implementează interfata Visitor  
(suprascrim metodele de visit)

Exemplu implementare

---

## STRATEGY

---

# WHEN?

**VREI SĂ FACI CEVA ÎN MAI MULTE MODURI**

# HOW?

**Pas 1:** Creăm interfața Strategy care conține o metodă doSmth()

**Pas 2:** Creăm clase concrete care implementează interfața Strategy

**Pas 3:** Creăm un Context - care are ca field o strategie

Exemplu implementare

---

## FACTORY

---

# WHEN?

**FOLOSIT PENTRU OBIECTE CARE GENEREAZĂ INSTANȚE DE CLASE ÎNRUDITE**

# HOW?

**Pas 1:** Creăm o interfață

**Pas 2:** Creăm clase concrete care să implementeze interfața de la pasul anterior

**Pas 3:** Creăm o clasă Factory care să genereze un obiect dintr-o clasă concretă pe baza informației date

**Pas 4:** Folosim Factory-ul ca să obținem un obiect din clasa concretă dând un argument (cum ar fi tipul)

Exemplu implementare

---

## OBSERVER

---

# WHEN?

**MECANISM DE SUBSCRIBE (CÂND SE PRODUCE O SCHIMBARE VREI SĂ ÎNȘTIINȚEZI PE CINEVA DE ACEST LUCRU**

# HOW?

**Pas 1:** Creăm clasa Subiect (OBSERVABLE) care va avea o listă de observers, o metodă de adăgat în lista de observers Si una de notifyAllObservers

**Pas 2:** Creăm clasa abstracta Observer - care trebuie să conțină o metodă de update

**Pas 3:** Creăm clase concrete care extind clasa Observer de la pasul anterior (suprascrim metoda de update in functie de observator)

Exemplu implementare