

# Inżynieria oprogramowania

## LAB 4 Wzorce projektowe. Wzorzec Singleton

### WZORCE PROJEKTOWE

<https://stormit.pl/singleton/>

**Wzorzec projektowy** (*ang. design pattern*) – w inżynierii oprogramowania, uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Wzorzec projektowy nie jest gotową implementacją rozwiązania, lecz jego opisem. Stosowane są w projektach wykorzystujących programowanie obiektowe.

### KLASYFIKACJA PODSTAWOWYCH WZORCÓW

#### Wzorce kreacyjne

- Budowniczy (obiektywy),
- Fabryka abstrakcyjna (obiektywy),
- Metoda wytwórcza (klasowy),
- Prototyp (obiektywy),
- Singleton (obiektywy);

#### Wzorce strukturalne

- Adapter (klasowy oraz obiektywy),
- Dekorator (obiektywy),
- Fasada (obiektywy),
- Kompozyt (obiektywy),
- Most (obiektywy),
- Pełnomocnik (obiektywy),
- Pyłek (obiektywy);

#### Wzorce czynnościowe

- Interpreter (klasowy),
- Iterator (obiektywy),
- Łańcuch zobowiązań (obiektywy),
- Mediator (obiektywy),
- Metoda szablonowa (klasowy),
- Obserwator (obiektywy),
- Odwiedzający (obiektywy),
- Pamiątka (obiektywy),
- Polecenie (obiektywy),
- Stan (obiektywy),
- Strategia (obiektywy).

## **SINGLETON**

Singleton jest to jeden z kreatywnych wzorców projektowych (obiektowy), którego celem jest ograniczenie możliwości tworzenia obiektów danej klasy do jednej instancji oraz zapewnienie globalnego dostępu do stworzonego obiektu. Niekiedy wzorzec uogólnia się do przypadku wprowadzenia pewnej maksymalnej liczby obiektów, jakie mogą istnieć w systemie. Niektórzy programiści uznają go za antywzorzec, ponieważ łamie zasady projektowania obiektowego, często bywa nadużywany lub sprowadza się do stworzenia obiektowego zamiennika dla zmiennej globalnej.

Zapewnia utworzenie i dostarczenie wyłącznie jednego egzemplarza obiektu danego typu. Obiekt ten powinien być globalnie dostępny i zazwyczaj ma charakter rejestru lub usługi wyszukiwanej, używany do pozyskiwania referencji do innych obiektów. Uczynienie klasy singletona jako final spowoduje brak możliwości wywodzenia klas z klasy singletona a, co za tym idzie, brak możliwości upublicznienia metody clone() która mogłaby być wykorzystana do utworzenia innej instancji obiektu.

### **METODA 1**

```
public final class Singleton {
    private final static Singleton s = new Singleton();
    private Singleton() {}
    public static Singleton getRef()
    {
        return s;
    }
}
```

### **METODA 2**

```
final class Singleton {
    private static volatile Singleton instance = null;
    private Singleton() {};
    public static Singleton getRef() {
        if (instance == null) {
            synchronized (Singleton.class) {
                if (instance == null) {
                    instance = new Singleton();
                }
            }
        }
        return instance;
    }
}
```

### **METODA 3**

```
protected Singleton() {
    private static class SingletonHolder {
        private final static Singleton instance = new Singleton();
    }
    public static Singleton getRef() {
        return SingletonHolder.instance;
    }
}
```

### **ZAD.1.**

Wykorzystując język obiektowy utworzyć model przekazywania głosów do Centralnej Komisji Wyborczej -CKW (Singleton) przez Okręgową Komisję Wyborczą (OKW).

- Utworzyć klasę OKW
- Utworzyć klasę CKW (jako singleton)
- W klasie CKW utworzyć:
  - Zmienną do zapamiętywania liczby głosów (**G**)
  - Metodę PrzekazGlosyCKW – nic nie zwraca, pobiera jako parametr liczbę głosów i zwiększa liczbę głosów **G** o podaną wartość
  - Metodę PokazGlosyCKW – zwraca w postaci liczby liczbę głosów **G**
- W klasie OKW w metodzie *main*:
  - Utworzyć obiekt ckw1 klasy CKW
  - Na obiekcie ckw1 wywołać metodę PokazGlosy
  - Na obiekcie ckw1 wywołać metodę PrzekazGlosy
  - Na obiekcie ckw1 wywołać metodę PokazGlosy
  - Utworzyć obiekt ckw2 klasy CKW
  - Na obiekcie ckw2 wywołać metodę PrzekazGlosy
  - Na obiekcie ckw2 wywołać metodę PokazGlosy