

# Nowoczesne języki programowania obiektowego

Obserwator



# Obserwator (Observer)

## Cel

- ▶ Modelowanie zależności typu jeden do wiele dla obiektów.
- ▶ Informacja o zmianie w stanie obiektu obserwowanego jest propagowana automatycznie do wszystkich obserwatorów.

## Zastosowania:

- ▶ Powiadomianie o zmianach.

# Obserwator

## Interfejs Obiekt obserwowany

`java.util.Observable`

- ▶ utrzymuje rejestr obiektów *Obserwator*
- ▶ umożliwia dołączanie i odłączanie obiektów *Obserwator*

## Interfejs Obserwator

`java.util.Observer`

- ▶ udostępnia interfejs do powiadamiania o zmianach

## Instancja Obiekt obserwowany

- ▶ powiadamia obiekty typu *Obserwator* o zmianie stanu

## Instancja Obserwator

- ▶ aktualizuje swój stan na podstawie powiadomienia

# Przykład i problemy

Materiały wykładowe:

[njpo.wzorce.obserwator.pdf](#)

# Ćwiczenia

Napisz krótki symulator giełdy korzystając ze wzorca Obserwator.

Klasa Giełda odpowiedzialna jest za zmianę wyceny spółek w niej notowanych.

Dodaj kilka spółek, wraz z ich wycenami.

Oprócz tego – zmieniaj im wartości wyceny.

Możesz to zrobić w sposób losowy

Zaimplementuj i wypełnij pozostałe klasy zgodnie z wzorcem projektowym Obserwator.

Źródło: [DP\\_obserwator.zip](#)