

# Podstawy teoretyczne

Tomasz Borzyszkowski

## Czym jest wzorzec?

Christopher Alexander



"Każdy wzorzec opisuje problem powtarzający się w danym środowisku i istotę rozwiązania tego problemu w taki sposób, że można wykorzystać określone rozwiązania milion razy i nigdy nie zrobić tego tak samo."

## Elementy opisu wzorca

1. **Nazwa wzorca:** 1-2 słowa opisujące istotę
2. **Opis problemu:** warunki stosowalności, jaki jest problem?
3. **Rozwiązanie:** opis elementów składających się na projekt, ich przeznaczenia, relacji i współdziałania między nimi
4. **Konsekwencje:** efekty, koszty, zyski

## Podział wzorców projektowych

		Rodzaj		
		Konstrukcyjne	Strukturalne	Operacyjne
Zasięg	Klasa	Metoda wytwórcza	Adapter	Interpreter Metoda szablonowa
	Obiekt	Fabryka abstrakcyjna Budowniczy Prototyp Singleton	Adapter Most Kompozyt Dekorator Fasada Pyłek Pełnomocnik	Łańcuch zobowiązań Polecenie Iterator Mediator Pamiętka Obserwator Stan Strategia Odwiedzający

# Singleton

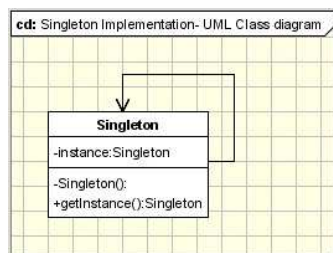
## Przeznaczenie

- Klasa ma tylko jeden egzemplarz

## Warunki stosowania

- Musi istnieć dokładnie jeden egzemplarz klasy dostępny klientom w znanym miejscu
- Po rozszerzeniu egzemplarza przez dziedziczenie, klienci mogą z niego korzystać bez zmian w kodzie

## Struktura



# Singleton przykłady

1. **Logger class:** zapewnia globalny punkt dostępu do rejestrowania wszystkich komponentów aplikacji bez konieczności tworzenia obiektu za każdym razem
2. **Configuration class:** j.w. + jeżeli wartości są odczytywane np. z bazy danych, pozwala uniknąć ponownego odczytywania wartości każdorazowo, gdy parametry konfiguracyjne są używane
3. **Dostęp do współdzielonych zasobów:** np. pule połączeń we współbieżnym środowisku klientów
4. **Fabryki implementowane jako singletony:** zapobiega współbieżnemu inicjalizowaniu fabryk przez wątki

# Singleton konsekwencje

1. Zapewnia kontrolę dostępu do jednego egzemplarza
2. Pozwala zmniejszyć przestrzeń nazw
3. Umożliwia dopracowywanie operacji i reprezentacji
4. Umożliwia określenie dowolnego limitu liczby egzemplarzy
5. Bardziej elastyczny od operacji statycznych

# Singleton problemy

1. **Dziedziczenie:** chcemy dziedziczyć po klasie Singleton w taki sposób by metoda `getInstance()` wywołana z klasy potomnej oddawała instancję typu klasy potomnej.  
*Zaproponuj implementację*
2. **Serializacja:** serializujemy singleton do pliku, następnie dwa niezależne wątki go deserializują, jak zapobiec powstaniu w ten sposób dwóch kopii singletona? (w Java ? w innych językach ?)  
*Zaproponuj implementację*

# Singleton współbieżność

