

WZORCE PROJEKTOWE A ROBOTYKA

Maciej Hołodniuk

AGENDA

- Wstęp do wzorców projektowych: definicja oraz rola w tworzeniu oprogramowania
- Przegląd typów wzorców projektowych
- Wzorzec "Adapter" a integracja sprzętu z robotyką
- Wzorzec "Obserwator" a systemy czujników w robotyce
- Wzorzec "Strategia" a sterowanie robotem
- Wzorzec "Stan" a sterowanie robotami w zmiennych warunkach
- Podsumowanie

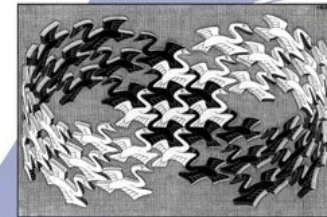
WZORZEC PROJEKTOWY

- **Definicja** – uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz utrzymanie kodu źródłowego. Jest opisem rozwiązania, a nie jego implementacją. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

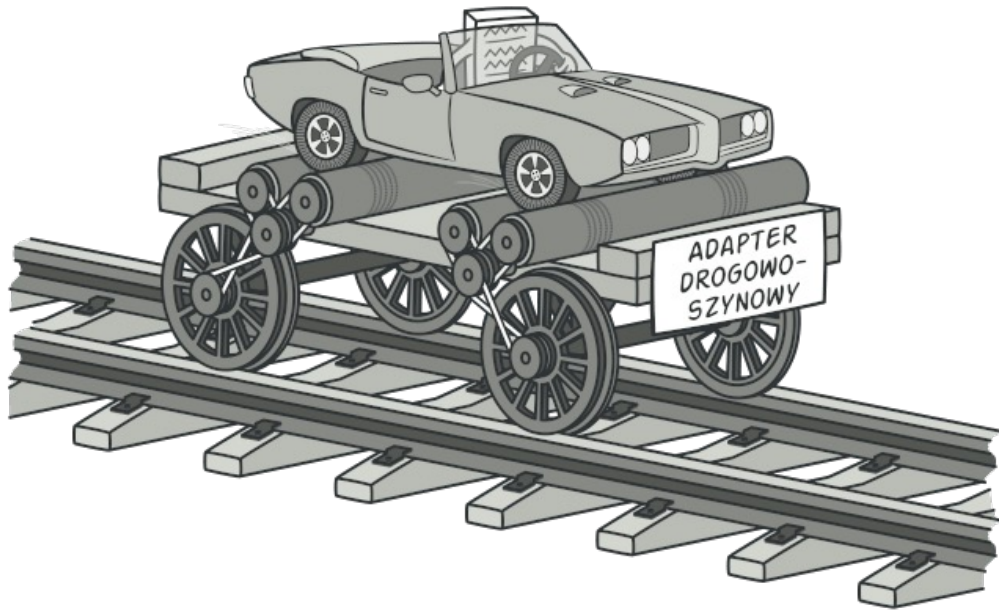
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES



KLASYFIKACJA WZORCÓW PROJEKTOWYCH

Kreacyjne	Strukturalne	Operacyjne (behawioralne)
<ul style="list-style-type: none">• Budowniczy• Fabryka abstrakcyjna• Metoda wytwórcza• Prototyp• Singleton	<ul style="list-style-type: none">• Adapter• Dekorator• Fasada• Kompozyt• Most• Pełnomocnik• Pyłek	<ul style="list-style-type: none">• Łańcuch zobowiązań• Polecenie• Interpreter• Iterator• Mediator• Memento• Metoda szablonowa• Obserwator• Strategia• Wizytator• Stan

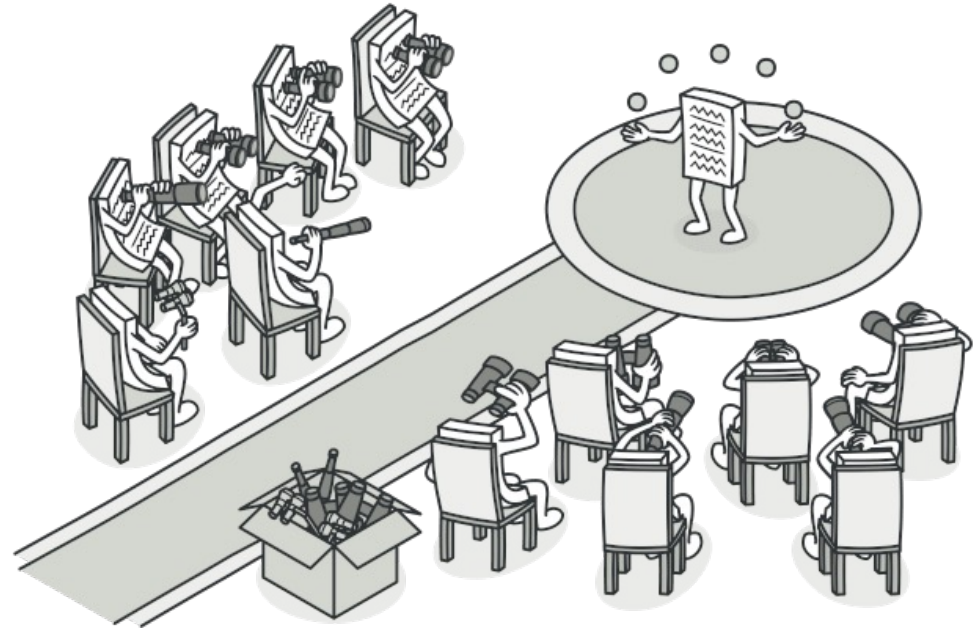
ADAPTER



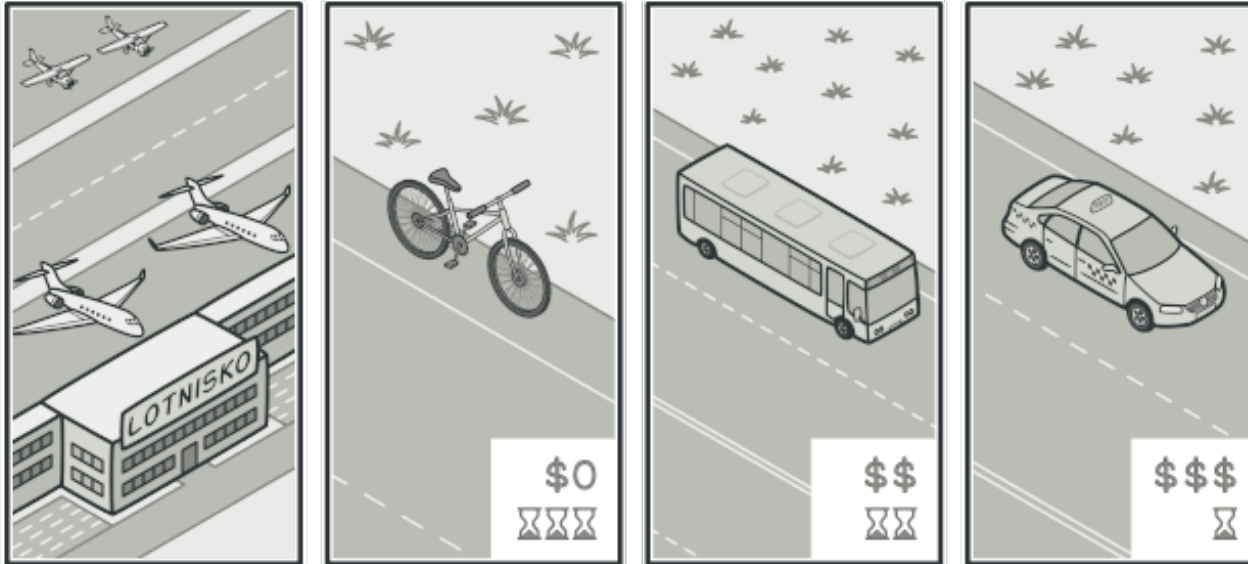
- **Definicja** – Przekształca interfejs klasy na inny, oczekiwany przez klienta. Adapter umożliwia współdziałanie klasom, które z uwagi na niezgodne interfejsy standardowo nie mogą współpracować ze sobą.
- W robotyce można znaleźć analogię do tego wzorca, gdy chcemy zintegrować różne sprzętowe komponenty, które mają różne interfejsy. Wykorzystując adaptery, możemy stworzyć jednolity interfejs, który pozwoli nam komunikować się z różnymi rodzajami sprzętu.

OBSERWATOR

- **Definicja** – Określa zależność jeden do wielu między obiektami. Kiedy zmieni się stan jednego z obiektów, wszystkie obiekty zależne od niego są o tym automatycznie powiadamiane i aktualizowane.
- W robotyce można to zastosować w kontekście systemów czujników, gdzie obserwator (np. oprogramowanie) reaguje na zmiany w danym czujniku, takich jak ruch, zmiana temperatury itp., i podejmuje odpowiednie działania.



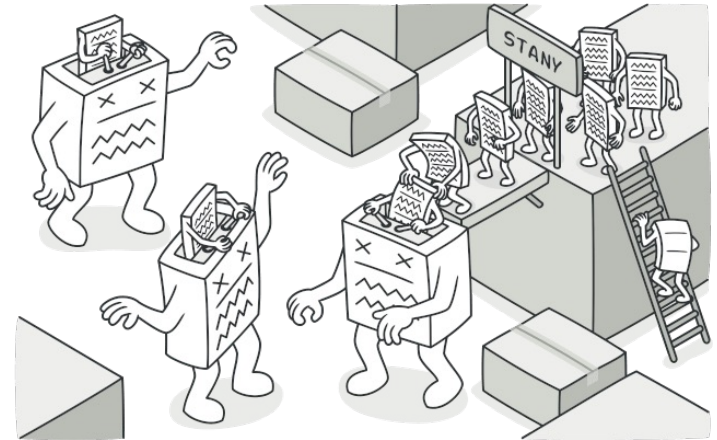
STRATEGIA



- **Definicja** – Określa rodzinę algorytmów, kapsułkuje każdy z nich i umożliwia ich zainne stosowanie. Wzorzec ten pozwala zmienić algorytmy niezależnie od korzystających z nich klientów.
- W robotyce możemy wykorzystać ten wzorzec do sterowania ruchem robotów, gdzie różne strategie mogą być wykorzystywane w zależności od określonych warunków lub zadań, takich jak unikanie przeszkód, planowanie trasy, czy gładkie poruszanie się.

STAN

- **Definicja** – Umożliwia obiektowi modyfikację zachowania w wyniku zmiany wewnętrznego stanu. Wygląda to tak, jakby obiekt zmienił klasę.
- W robotyce można zastosować ten wzorzec, aby sterować robotami w różnych warunkach środowiskowych. Na przykład, robot może mieć różne stany, takie jak "chód", "bieg", "uniesienie przedmiotu" itp., a wzorzec "Stan" pozwoli na dynamiczne przejście między tymi stanami w zależności od kontekstu.



PODSUMOWANIE

- Wzorce projektowe są sprawdzonymi rozwiązaniami w programowaniu, które zapewniają elastyczność, łatwość modyfikacji i rozszerzalność oprogramowania.
- Część wzorców ma swoje analogie w dziedzinie robotyki, które pozwalają na sprawną implementację wybranych funkcjonalności.
- Zastosowanie wzorców projektowych w robotyce sprawia, że tworzenie zaawansowanych systemów staje się bardziej intuicyjne, elastyczne i łatwiejsze w utrzymaniu.

BIBLIOGRAFIA

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software.
- https://youtu.be/tAuRQs_d9F8 - 8 Design patterns every developer should know
- <https://refactoring.guru/> - Refactoring guru
- [https://pl.wikipedia.org/wiki/Wzorzec_projektowy_\(informatyka\)](https://pl.wikipedia.org/wiki/Wzorzec_projektowy_(informatyka))