

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Projektowanie zorientowane obiektowo. Wzorce projektowe

Autorzy: Alan Shalloway, James R. Trott

Tłumaczenie: Jaromir Senczyk

ISBN: 83-7197-831-6

Tytuł oryginału: [Design Patterns Explained: A New Perspective on Object-Oriented Design](#)

Format: B5, stron: 256



W książce omówiono programowanie obiektowe (OOP) i wzorce projektowe (design patterns) – dwa terminy, które zrobiły ostatnio zawrotną karierę.

Programowanie obiektowe jest podstawą dominujących obecnie języków programowania, takich jak C++ czy Java. Jego znajomość to jedna z podstawowych kompetencji współczesnego programisty. Kolejnym etapem wtajemniczenia są wzorce projektowe: sprawdzone sposoby radzenia sobie z typowymi zadaniami.

Kto nie pozna podstawowych zasad programowania obiektowego, nie zrozumie wzorców projektowych. Kto nie przyswoi sobie stosowania wzorców, nie wykorzysta pełni możliwości programowania obiektowego. Dlatego oba te – nierozdzielnie związane ze sobą zagadnienia omówiono w jednej pozycji, która uczy programowania obiektowego poprzez naukę wzorców projektowych, nie oddzielając sztucznie tych dwóch dziedzin.

Autorzy tej książki starają się przekonać Czytelnika, że odpowiednie stosowania właściwych wzorców projektowych możliwe jest na każdym etapie tworzenia aplikacji: od analizy, przez projektowanie, do implementacji. Co więcej, pozwala ono wyrobić prawidłowe nawyki programowania obiektowego.

Dwa w jednym: poznaj programowanie obiektowe poprzez wzorce projektowe.

- Podstawowe zasady obiektowości
- Hermetyzacja i jej znaczenie
- Język i notacja UML
- Wzorce: fasady, adaptacji, mostu, fabryki abstrakcyjnej, dekoratora, singletonu, blokowania dwufazowego, metody szablonu i fabryki
- Zasady projektowania z użyciem wzorców



# Spis treści

<b>Przedmowa</b> .....	<b>9</b>
<b>Część I Wprowadzenie do programowania obiektowego</b> .....	<b>17</b>
<b>Rozdział 1. Obiektość</b> .....	<b>19</b>
Przegląd.....	19
Zanim pojawiły się obiekty: dekompozycja funkcjonalna.....	19
Problem określenia wymagań .....	21
Zmiany wymagań a dekompozycja funkcjonalna.....	22
Postępowanie w sytuacji zmieniających się wymagań .....	24
Obiektość.....	26
Programowanie obiektowe w praktyce .....	31
Szczególne rodzaje metod.....	33
Podsumowanie .....	34
<b>Rozdział 2. Język UML</b> .....	<b>37</b>
Przegląd.....	37
Czym jest język UML? .....	37
Zastosowanie języka UML.....	38
Diagram klas .....	39
Diagramy interakcji.....	43
Podsumowanie .....	45
<b>Część II Ograniczenia tradycyjnie pojmowanego projektowania obiektowego</b> ..	<b>47</b>
<b>Rozdział 3. Problem wymagający uniwersalnego rozwiązania</b> .....	<b>49</b>
Przegląd.....	49
Pozyskanie informacji z systemu CAD/CAM .....	49
Terminologia dziedziny zastosowań .....	50
Opis problemu .....	51
Prawdziwe wyzwania.....	53
Podsumowanie .....	56
<b>Rozdział 4. Standardowe rozwiązanie obiektowe</b> .....	<b>57</b>
Przegląd.....	57
Rozwiązanie wykorzystujące specjalizację.....	57
Podsumowanie .....	63
Dodatek: Przykłady w języku C++ .....	63

<b>Część III</b>	<b>Wzorce projektowe .....</b>	<b>67</b>
<b>Rozdział 5.</b>	<b>Wprowadzenie do wzorców projektowych.....</b>	<b>69</b>
	Przegląd.....	69
	Wzorce projektowe wywodzą się z architektury i antropologii.....	70
	Wzorce projektowe — od architektury do programowania .....	73
	Po co studiować wzorce projektowe? .....	75
	Inne zalety studiowania wzorców projektowych .....	78
	Podsumowanie .....	79
<b>Rozdział 6.</b>	<b>Wzorzec fasady.....</b>	<b>81</b>
	Przegląd.....	81
	Wprowadzenie do fasady .....	81
	Fasada.....	82
	Praktyczne uwagi na temat zastosowania fasady.....	84
	Zastosowanie fasady w rozwiązaniu problemu CAD/CAM.....	85
	Podsumowanie .....	85
<b>Rozdział 7.</b>	<b>Wzorzec adaptera .....</b>	<b>87</b>
	Przegląd.....	87
	Wprowadzenie do wzorca adaptera.....	87
	Adapter.....	88
	Praktyczne uwagi na temat zastosowania adaptera.....	92
	Zastosowanie adaptera w celu rozwiązania problemu CAD/CAM .....	94
	Podsumowanie .....	95
	Dodatek: Przykład w języku C++ .....	95
<b>Rozdział 8.</b>	<b>Poszerzamy horyzonty .....</b>	<b>97</b>
	Przegląd.....	97
	Obiekty.....	98
	Hermetyzacja.....	99
	Określ zmienność i hermetyzuj ją.....	101
	Wspólności i zmienności a klasy abstrakcyjne .....	104
	Podsumowanie .....	105
<b>Rozdział 9.</b>	<b>Wzorzec mostu .....</b>	<b>107</b>
	Przegląd.....	107
	Wprowadzenie do wzorca mostu .....	107
	Przykład problemu wymagającego zastosowania mostu .....	108
	Obserwacja dotycząca zastosowań wzorców projektowych.....	116
	Wyprowadzenie wzorca mostu .....	117
	Wzorzec mostu — retrospekcja .....	125
	Praktyczne uwagi na temat zastosowań mostu .....	125
	Podsumowanie .....	128
	Dodatek: przykłady kodu w języku C++ .....	130
<b>Rozdział 10.</b>	<b>Wzorzec fabryki abstrakcyjnej .....</b>	<b>135</b>
	Przegląd.....	135
	Wprowadzenie do wzorca fabryki abstrakcyjnej .....	135
	Fabryka abstrakcyjna — przykład zastosowania .....	136
	Implementacja wzorca fabryki abstrakcyjnej.....	141
	Praktyczne uwagi na temat stosowania fabryki abstrakcyjnej .....	144
	Zastosowanie fabryki abstrakcyjnej w rozwiązaniu problemu CAD/CAM.....	146
	Podsumowanie .....	147
	Dodatek: przykłady w języku C++ .....	147

<b>Część IV</b>	<b>Projektowanie z wykorzystaniem wzorców .....</b>	<b>149</b>
<b>Rozdział 11. W jaki sposób projektują eksperci? .....</b>		<b>151</b>
Przegląd .....		151
Tworzenie przez dodawanie wyróżnień .....		151
Podsumowanie .....		156
<b>Rozdział 12. Rozwiązanie problemu CAD/CAM z wykorzystaniem wzorców projektowych .....</b>		<b>159</b>
Przegląd .....		159
Przypomnienie problemu CAD/CAM .....		159
Projektowanie z wykorzystaniem wzorców .....		160
Projektowanie z wykorzystaniem wzorców: etap 1 .....		161
Projektowanie z wykorzystaniem wzorców: etap 2a .....		161
Projektowanie z wykorzystaniem wzorców: etap 2b .....		165
Projektowanie z wykorzystaniem wzorców: etap 2c .....		168
Projektowanie z wykorzystaniem wzorców: etap 2d (fasada) .....		168
Projektowanie z wykorzystaniem wzorców: etap 2d (adapter) .....		169
Projektowanie z wykorzystaniem wzorców: etap 2d (fabryka abstrakcyjna) .....		170
Projektowanie z wykorzystaniem wzorców: etap 3 .....		170
Porównanie z poprzednimi wersjami rozwiązania .....		170
Podsumowanie .....		172
<b>Rozdział 13. Zasady i strategie projektowania z wykorzystaniem wzorców .....</b>		<b>173</b>
Przegląd .....		173
Zasada otwarcia i zamknięcia .....		174
Zasada projektowania w kontekście .....		174
Zasada zawierania zmienności .....		177
Podsumowanie .....		178
<b>Część V</b>	<b>Obsługa zmienności za pomocą wzorców projektowych .....</b>	<b>179</b>
<b>Rozdział 14. Wzorzec strategii .....</b>		<b>181</b>
Omówienie .....		181
Sposób obsługi nowych wymagań .....		181
Studium problemu i początkowe wymagania .....		183
Obsługa nowych wymagań .....		184
Wzorzec strategii .....		187
Praktyczne uwagi na temat stosowania wzorca strategii .....		188
Podsumowanie .....		189
<b>Rozdział 15. Wzorzec dekoratora .....</b>		<b>191</b>
Omówienie .....		191
Nowe szczegóły .....		191
Wzorzec dekoratora .....		193
Zastosowanie dekoratora w omawianym studium problemu .....		194
Inne zastosowania: operacje wejścia i (lub) wyjścia .....		196
Praktyczne uwagi na temat stosowania dekoratora .....		198
Podsumowanie .....		199
Dodatek: Przykłady w języku C++ .....		199
<b>Rozdział 16. Wzorzec singletonu oraz wzorzec blokowania dwufazowego .....</b>		<b>201</b>
Przegląd .....		201
Wprowadzenie do wzorca singletonu .....		202
Zastosowanie wzorca singletonu .....		202
Wariant: wzorzec blokowania dwufazowego .....		204
Praktyczne uwagi na temat zastosowania singletonu i blokowania dwufazowego .....		205
Podsumowanie .....		206
Dodatek: przykłady w języku C++ .....		206

<b>Rozdział 17. Wzorzec obserwatora .....</b>	<b>207</b>
Omówienie .....	207
Kategorie wzorców .....	207
Nowe wymagania .....	208
Wzorzec obserwatora .....	209
Zastosowanie wzorca obserwatora .....	210
Praktyczne uwagi na temat zastosowania obserwatora .....	215
Podsumowanie .....	216
Dodatek: przykłady kodu w języku C++ .....	216
<b>Rozdział 18. Wzorzec metody szablonu .....</b>	<b>219</b>
Przegląd .....	219
Nowe wymagania .....	219
Wzorzec metody szablonu .....	220
Zastosowanie wzorca metody szablonu .....	220
Praktyczne uwagi na temat zastosowania szablonu metody .....	221
Podsumowanie .....	222
<b>Rozdział 19. Wzorzec metody fabryki .....</b>	<b>223</b>
Przegląd .....	223
Nowe wymaganie .....	223
Wzorzec metody fabryki .....	224
Praktyczne uwagi dotyczące zastosowania wzorca metody fabryki .....	225
Podsumowanie .....	226
<b>Rozdział 20. Macierz analizy .....</b>	<b>227</b>
Przegląd .....	227
Zmienność w świecie rzeczywistym .....	227
Studium zmienności: system sprzedaży przez Internet .....	228
Uwagi praktyczne .....	233
Podsumowanie .....	234
<b>Część VI Zakończenie i początek .....</b>	<b>235</b>
<b>Rozdział 21. Wzorce projektowe i nowa perspektywa projektowania obiektowego .....</b>	<b>237</b>
Przegląd .....	237
Podsumowanie zasad obiektowości .....	238
Hermetyzacja implementacji za pomocą wzorców projektowych .....	238
Analiza wspólności i zmienności i wzorce projektowe .....	239
Dekompozycja dziedziny problemu poprzez określenie odpowiedzialności .....	239
Powiązania wewnątrz wzorców .....	240
Wzorce i projektowanie w kontekście .....	241
Uwagi praktyczne .....	242
Podsumowanie .....	242
<b>Rozdział 22. Bibliografia .....</b>	<b>243</b>
Programowanie zorientowane obiektowo: strony WWW .....	243
Zalecana lektura na temat wzorców projektowych i obiektowości .....	244
Lektura przeznaczona dla programistów korzystających z języka Java .....	245
Lektura przeznaczona dla programistów korzystających z języka C++ .....	246
Lektura przeznaczona dla programistów korzystających z języka COBOL .....	246
Lektura dotycząca metodyki Extreme Programming .....	246
Zalecana lektura dotycząca programowania .....	247
Ulubiona lektura autorów .....	247
<b>Skorowidz .....</b>	<b>249</b>

Rozdział 5.

# Wprowadzenie do wzorców projektowych

## Przegląd

Rozdział ten wprowadza pojęcie wzorców projektowych.

Omawiam w nim:

- ◆ powstanie wzorców projektowych w architekturze i ich zastosowanie w dziedzinie projektowania oprogramowania;
- ◆ powody studiowania wzorców projektowych.

Wzorce projektowe stanowią jedno z najważniejszych osiągnięć programowania obiektowego. Są one przedmiotem zainteresowania autorów książek fachowych, przedmiotem szkoleń oraz podstawą opracowania nowych narzędzi analitycznych. Często można spotkać się przy tym ze stwierdzeniem, że aby rozpocząć studiowanie wzorców projektowych, niezbędne jest doskonałe opanowanie podstaw programowania obiektowego. Z mojej praktyki wynika coś zupełnie przeciwnego: wczesne przyswojenie sobie tematyki wzorców projektowych w procesie uczenia się technik obiektowych znakomicie ułatwia właściwe zrozumienie analizy i projektowania obiektowego.

W pozostałej części książki omówię nie tylko wzorce projektowe, ale także sposób, w jaki ujawniają one i wymuszają zasady poprawnego projektowania obiektowego. Mam nadzieję, że w ten sposób ułatwię czytelnikowi głębsze zrozumienie tych zasad, a także będę mógł pokazać mu, dlaczego wzorce projektowe stanowią przykłady doskonałych, obiektowych rozwiązań.

Część przedstawionego tu materiału może wydać się czytelnikowi nieco zbyt abstrakcyjna i filozoficzna. Warto jednak zapoznać się z nim. Rozdział ten kładzie bowiem podwaliny pod właściwe zrozumienie tematyki wzorców projektowych. Zrozumienie zawarte w nim materiału ułatwi czytelnikowi posługiwanie się wzorcami projektowymi.

Wiele z zaprezentowanych pomysłów pochodzi z książki „The Timeless Way of Building”<sup>1</sup> autorstwa Christophera Alexandra. Wykorzystać je będą w całej książce.

## Wzorce projektowe wywodzą się z architektury i antropologii

Wiele lat temu architekt o nazwisku Christopher Alexander zadał sobie pytanie: „czy jakość jest cechą obiektywną?”. Czy piękno doskonałych rozwiązań istnieje jedynie dla tych, którzy je dostrzegają, czy też istnieje powszechna zgoda w kwestii tego, co jest piękne, a co nie jest? Alexander był szczególnie zainteresowany pojęciem piękna w odniesieniu do jakości rozwiązań architektonicznych i sposobem, w jaki określamy ich doskonałość. Jeśli mamy na przykład zaprojektować mamy wejście domu, to skąd będziemy wiedzieć, że uzyskane rozwiązanie jest dobre? Czy potrafimy to właściwie rozpoznać? Czy istnieje jakaś obiektywna podstawa takiej oceny? Czy tkwi ona w samym projekcie, czy powstaje raczej na zasadzie opinii większości?

Alexander postuluje istnienie obiektywnych podstaw oceny rozwiązań architektonicznych. Według niego taka, a nie inna ocena rozwiązania architektonicznego nie wynika z czyjegoś gustu. Doskonałość rozwiązania możemy ocenić na podstawie obiektywnych przesłanek.

Podobne wnioski uzyskano w dziedzinie antropologii kultury. Sugeruje się, że w danej kulturze, jednostki zgadzają się ze sobą w znacznym stopniu co do tego, czy dane rozwiązanie jest doskonałe. Wspólna kultura sprawia, że obiektywny sposób oceny góruje nad indywidualnymi gustami. Sądzę więc, że istnieją transcendentne wzorce, które stanowią obiektywną podstawę oceny danego rozwiązania. Jedną z głównych gałęzi antropologii kultury zajmuje się właśnie poszukiwaniem wzorców służących do opisu zachowań i wartości wewnątrz poszczególnych kultur<sup>2</sup>.

Koncepcja wzorców projektowych także zakłada, że jakość oprogramowania można zmierzyć w obiektywny sposób.

Jeśli jednak zaakceptujemy możliwość obiektywnego rozpoznania i opisanie dobrych rozwiązań, czy nie powinniśmy także umieć je skonstruować? Wydaje mi się, że Alexander musiał zadać sobie pytanie:

*Co takiego możemy odnaleźć w rozwiązaniu dobrym, czego brakuje rozwiązaniom złym?*

A także:

*Co takiego możemy odnaleźć w rozwiązaniu złym, czego nie ma dobre rozwiązanie?*

---

<sup>1</sup> Alexander C., Ishikawa S., Silverstein M.: *The Timeless Way of Building*, Nowy Jork: Oxford University Press, 1979.

<sup>2</sup> Antropolog Ruth Benedict jest pionierem analizy kultur wykorzystujących wzorce. Przykłady znajdzie czytelnik w pracy Benedict, R.: *The Chrysanthemum and the Sword*, Boston: Houghton Mifflin, 1946.

Pytania te wynikają z przekonania Alexandra dotyczącego tego, że jeśli jakość rozwiązań jest ich cechą obiektywną, to powinniśmy potrafić zidentyfikować przyczyny tego, że jedne rozwiązania są dobre, a inne złe.

Aby znaleźć właściwe odpowiedzi Alexander, studiował rozwiązania budynków, ulic, miast, właściwie wszelkie aspekty przestrzeni, którą otacza się człowiek. Rezultatem tych obserwacji było odkrycie, że dobre rozwiązania mają szereg cech wspólnych.

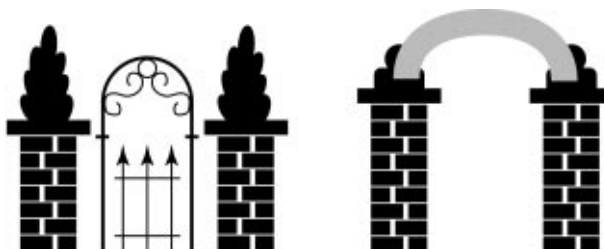
Rozwiązania architektoniczne różnią się między sobą, nawet jeśli są tego samego typu. Mimo tych różnic mogą jednak szczyścić się doskonałą jakością.

Na przykład dwa różne portyki mogą różnić się swoją konstrukcją, a jednak mieć równie wysoką jakość. Mogą bowiem stanowić rozwiązanie różnych problemów w przypadku różnych budynków. Jeden z portyków służyć może przejściu z ulicy do drzwi frontowych budynku, a drugi dodatkowo dostarczać cienia w słoneczny dzień. Oba portyki mogą także rozwiązywać ten sam problem na różne sposoby.

Alexander zdawał sobie sprawę z tego, że struktur architektonicznych nie wolno rozpatrywać w oderwaniu od problemów, których rozwiązanie stanowią. Próbuąc zidentyfikować pojęcie jakości projektów zrozumiał, że powinien poszukiwać go analizując różne rozwiązania tego samego problemu. Na rysunku 5.1 przedstawiono dwa rozwiązania problemu wyróżnienia wejścia.

#### Rysunek 5.1.

*Struktury architektoniczne mogą różnie wyglądać, ale nadal rozwiązywać ten sam problem*



Alexander odkrył, że zawężając pole analizy do rozwiązań tego samego problemu, można dokładniej określić podobieństwa rozwiązań o wysokiej jakości. Podobieństwa te nazwał *wzorcami*.

Zdefiniował pojęcie wzorca jako „rozwiązanie problemu w określonym kontekście”:

„Każdy wzorec opisuje problem, który ciągle pojawia się w naszej dziedzinie, a następnie określa zasadniczą część jego rozwiązania w taki sposób, by można było zastosować je nawet milion razy, za każdym razem w nieco inny sposób”<sup>3</sup>.

Dokonajmy przeglądu pracy Alexandra, aby zilustrować to stwierdzenie. W tabeli 5.1 prezentuję fragmenty jego książki zatytułowanej „The Timeless Way of Building”<sup>4</sup>, która stanowi doskonale przedstawienie filozofii wzorców.

<sup>3</sup> Alexander C., Ishikawa S., Silverstein M.: *A Pattern Language*, Nowy Jork: Oxford University Press, 1977, s. x.

<sup>4</sup> Alexander C., Ishikawa S., Silverstein M.: *The Timeless Way of Building*, Nowy Jork: Oxford University Press, 1979.



**Tabela 5.1.** *Fragmety „The Timeless Way of Building”*

<b>Alexander pisze...</b>	<b>Mój komentarz</b>
<p>W ten sam sposób prawidłowo zaprojektowane podwórze pomaga mieszkańcom korzystać z niego.</p>	<p>Wzorzec posiada nazwę oraz zastosowanie. Tutaj nazwą jest <i>podwórze</i>, a jego zadaniem — pomoc mieszkańcom w korzystaniu z niego.</p>
<p>Zastanówmy się nad siłami, które rządzą na podwórzu. Najbardziej podstawowa kieruje tam mieszkańców w poszukiwaniu przestrzeni na zewnątrz budynku, w której zachować mogą swoją prywatność. Popatrzyć w gwiazdy, nacieszyć się słońcem czy rosnącymi tam roślinami.</p>	<p>Chociaż czasami jest to oczywiste, ważne jest to, by określić wprost rozwiązywany problem, który jest przecież powodem zastosowania danego wzorca. Tak czyni też Alexander w przypadku podwórza.</p>
<p>Działają tam także bardziej subtelne siły. Jeśli na przykład podwórze jest zamknięte ze wszystkich stron, mieszkańcy nie czują się tam dobrze i unikają przebywania na nim. Widok na odległą, większą przestrzeń okazuje się niezbędny.</p>	<p>Wskazuje trudności wynikające z uproszczonego rozwiązania oraz metody jego poprawy.</p>
<p>Jeśli mieszkańcy codziennie przechodzą przez podwórze w drodze do swoich normalnych zajęć, to przyzwyczajają się do niego, będą uważać za naturalne miejsce pobytu i chętniej będą przebywać na podwórzu.</p>	<p>Przyzwyczajenie i rutyna często są przyczyną, dla której nie dostrzegamy rzeczy oczywistych. Wartość wzorca polega na tym, że mniej doświadczeni projektanci mogą wykorzystać zawartą we wzorcu wiedzę dotyczącą poprawnego rozwiązania problemu.</p>
<p>Natomiast podwórze, na które mieszkańcy muszą wejść, jeśli tego „chcą”, nie jest dobrze znajomym miejscem i zwykle pozostaje nieużywane. Ludzie preferują spędzanie czasu w miejscach, do których są przyzwyczajeni. Istnieje także pewna bariera, która w subtelny, ale skuteczny sposób powstrzymuje nas przed przekroczeniem granicy pomiędzy wnętrzem a podwórzem.</p>	<p>Autor proponuje rozwiązanie jednego z istotnych, a często pomijanych aspektów problemu podwórza.</p>
<p>Jeśli natomiast istnieje pewna przestrzeń — na przykład portyk lub weranda, częściowo otwarta na zewnątrz i psychologicznie znajdująca się „w połowie drogi” pomiędzy wnętrzem a podwórzem, to sprawia ona, że łatwiej jest postawić każdy krok zbliżający nas do wyjścia na zewnątrz...</p>	<p>Alexander opowiada, w jaki sposób zaprojektować dobre podwórze...</p>
<p>Gdy podwórze posiada widok na większą przestrzeń, przecinają je ścieżki wiodące do różnych pomieszczeń, a ponadto ozdabia je portyk lub weranda, to działające na podwórzu siły same skłaniają mieszkańców do przebywania na nim. Widok na nieograniczoną przestrzeń stwarza poczucie komfortu, przecinające podwórze ścieżki pozwalają łatwiej się do niego przyzwyczać, a portyk ułatwia wyjście na podwórze. W ten sposób stopniowo podwórze staje się miejscem, w którym mieszkańcy chętnie przebywają.</p>	<p>... a ponadto wskazuje, skąd pochodzi wartość takiego rozwiązania.</p>

Tak więc opis wzorca według Alexandra składa się z czterech elementów:

- ♦ z nazwy wzorca;
- ♦ z zadania wzorca, czyli problemu, którego stanowi rozwiązanie;
- ♦ ze sposobu rozwiązania problemu;
- ♦ z ograniczeń i sił, które należy wziąć pod uwagę stosując rozwiązanie.

Alexander postuluje istnienie wzorców dla dowolnych problemów w dziedzinie architektury. Zaleca także wspólne zastosowanie wielu wzorców w celu rozwiązania złożonych problemów architektonicznych.

W dalszej części książki przedstawię sposoby jednoczesnego wykorzystania wielu wzorców. Teraz jednak skoncentruję się na stwierdzeniu Alexandra, które dotyczy tego, że wzorce są przydatne do rozwiązywania poszczególnych problemów.

## Wzorce projektowe — od architektury do programowania

Czytelnikowi może nasunąć się tutaj pytanie — co wspólnego mają zagadnienie z dziedziny architektury z programowaniem?

Na początku lat 90. grupa programistów zetknęła się z książką Alexandra dotyczącą wzorców projektowych. Zaczęli się zastanawiać, czy wnioski autora dotyczące wzorców w architekturze nie są także prawdziwe w przypadku projektowania oprogramowania<sup>5</sup>:

- ♦ czy tworząc programy spotykamy się z problemami, które rozwiązać można w podobny sposób?
- ♦ czy w przypadku projektowania oprogramowania można zastosować wzorce tak, by pomagały one w tworzeniu konkretnych rozwiązań?

Odpowiedź na tak postawione pytania była według nich jednoznacznie pozytywna. Należało wobec tego zidentyfikować przynajmniej kilka wzorców i opracować metody dalszego ich katalogowania.

Na początku lat 90. wiele osób zajmowało się problematyką wzorców projektowych. Jednak największy wpływ na prace tej społeczności wywarła książka „Design Patterns: Elements of Reusable Object-Oriented Software”<sup>6</sup>, której autorami byli Gamma,

---

<sup>5</sup> Podobną problematyką zajmowało się w Europie w latach osiemdziesiątych konsorcjum ESPRIT. W ramach projektów 1098 oraz 5248 rozwinięto opartą na wzorcach metodologię projektowania nazwaną KADS (*Knowledge Analysis and Design System*) służącą do wykorzystania wzorców do tworzenia systemów ekspertowych. Karen Gardner rozszerzyła później wzorce KADS o problematykę obiektowości. Patrz: Gardner, K.: *Cognitive Patterns: Problem-Solving Frameworks for Object Technology*, Nowy Jork: Cambridge University Press, 1998.

<sup>6</sup> Gamma E., Helm R., Johnson R., Vlissides J.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, Mass.: Addison-Wesley, 1995.

Helm, Johnson i Vlissides. Podkreślając znaczenie tej pracy autorów często określa się mianem Bandy Czworoga.

Pozycja ta spełniła następujące, ważne zadania:

- ♦ pokazała zastosowanie wzorców projektowych w dziedzinie projektowania oprogramowania;
- ♦ zaproponowała sposób katalogowania i opisu wzorców projektowych;
- ♦ skatalogowała 23 wzorce projektowe;
- ♦ postulowała wprowadzenie do projektowania obiektowego szeregu zasad i strategii opartych na wzorcach projektowych.

Ważne jest to, aby czytelnik zrozumiał, że autorzy tej książki nie stworzyli opisanych w niej wzorców projektowych. Zidentyfikowali raczej wzorce używane przez programistów i uważane powszechnie za przykłady doskonałych rozwiązań pewnych problemów (zwróćmy w tym miejscu uwagę na podobieństwo do pracy Alexandra).

Do dnia dzisiejszego zaproponowano szereg różnych sposobów opisu wzorców projektowych. Jako że książka niniejsza nie jest poświęcona procesowi wyszukiwania nowych wzorców, nie jest ważne, który z tych sposobów jest najlepszy. Każdy ze sposobów opisu wzorców jest dobry, jeśli uwzględnia elementy wymienione w tabeli 5.2.

**Tabela 5.2.** Kluczowe elementy opisu wzorców

Element	Opis
Nazwa	Identyfikuje wzorzec.
Intencja	Zastosowanie wzorca.
Problem	Dany wzorzec próbuje go rozwiązać.
Rozwiązanie	Sposób, w jaki wzorzec rozwiązuje problem w danym kontekście.
Uczestnicy i współpracownicy	Byty w obrębie wzorca.
Konsekwencje	Dodatkowe efekty wynikające z zastosowania wzorca.
Implementacja	Sposób, w jaki można zaimplementować wzorzec. <i>Uwaga:</i> implementacja dotyczy zawsze konkretnego zastosowania wzorca i nie powinna być mylona z konstrukcją wzorca.
Referencja do pracy Bandy Czworoga	Wskazanie źródła, w którym znajdziemy więcej informacji na temat wzorca.

Dla każdego z opisanych w książce wzorców przedstawię jednostronicowe zestawienie najważniejszych jego cech.

#### **Konsekwencje**

Termin *konsekwencje* użyty w odniesieniu do opisu wzorców często nie jest rozumiany właściwie. Zastosowany na co dzień zwykle posiada wydźwięk negatywny (rzadko powiemy: „Wygrałem na loterii! *Konsekwencją* będzie to, że nie muszę już chodzić do pracy”). W przypadku wzorców projektowych konsekwencje oznaczają po prostu związek przyczynowo-skutkowy.

## Po co studiować wzorce projektowe?

Kiedy czytelnik ma już ogólne pojęcie o tym, czym są wzorce projektowe, może zastanawiać się, czy warto się nimi zajmować. Istnieje wiele powodów — mniej lub bardziej oczywistych — dla których warto to robić.

Najczęściej podaje się określony zestaw powodów, dla których warto poznać wzorce projektowe.

- ♦ Możliwość wykorzystania istniejących, sprawdzonych wcześniej sposobów projektowania przyspiesza pracę nad projektem i pozwala uniknąć błędów na etapie projektowania. Stosując wzorce projektowe wykorzystujemy sumę doświadczeń innych projektantów; nie musimy już samodzielnie wymyślać rozwiązań typowych problemów.
- ♦ Praca w zespole i komunikacja pomiędzy jego członkami wymaga istnienia podstawowego, wspólnego zasobu terminologii oraz wspólnej perspektywy oglądu problemu. Wzorce projektowe zapewniają wspólny punkt odniesienia w fazie analizy i projektowania.

Według mnie istnieje jeszcze trzeci, równie ważny powód, dla którego każdy projektant powinien zapoznać się z tą tematyką.

- ♦ Wzorce projektowe dają ogólną perspektywę widzenia problemów, a także projektowania obiektowego, co uwalnia projektanta od konieczności zbyt wczesnego zgłębiania szczegółów.

Mam nadzieję, że po przeczytaniu tej książki czytelnik zgodzi się ze mną, że jest to jeden z najważniejszych powodów studiowania wzorców projektowych.

Aby zilustrować tę korzyść płynącą ze studiowania wzorców, posłużę się przykładem rozmowy dwu stolarzy, która dotyczy konstrukcji szuflad<sup>7</sup>.

**Pierwszy Stolarz:** W jaki sposób wykonamy te szuflady?



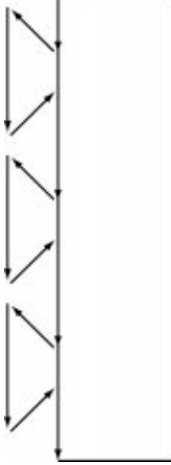
**Drugi Stolarz:** Myślę, że połączenia zrobimy w ten sposób, że przetniemy drewno najpierw prosto w dół, potem z powrotem pod kątem 45°, a później znowu w przeciwnym kierunku prosto w dół, a potem cofniemy się do góry pod kątem 45°, potem znów prosto, a później...

Czy czytelnik potrafi już powiedzieć, o czym oni rozmawiają?

Czy opis ten nie jest zbyt zagmatwany? O czym właściwie mówi Drugi Stolarz? Zbyt wiele szczegółów utrudnia nam prawidłowe zrozumienie. Spróbujmy wykonać rysunek na podstawie usłyszanego opisu.

---

<sup>7</sup> Autorzy zainspirowani zostali jednym z wykładów Ralpha Johnsona i zaadaptowali go na potrzeby książki.

Drugi Stolarz mówi:	A na rysunku wygląda to następująco:
„Myślę, że połączenia zrobimy w ten sposób, że przetniemy drewno najpierw prosto w dół, potem z powrotem pod kątem 45°...”	
„później znowu w przeciwnym kierunku prosto w dół, a potem cofniemy się do góry pod kątem 45°, potem znów prosto, a później...”	
„... aż otrzymamy połączenie „na wpust””	

Czy nie przypomina to wielu opisów kodu programu? Takich, w których programista opisuje jego działanie następująco:

„A tutaj w pętli wykonam..., po czym rozwiążę to za pomocą serii instrukcji warunkowych...i dodam instrukcję wyboru, aby odpowiednio obsłużyć...”

Z takiego opisu uzyskamy być może informacje o szczegółach działania kodu, ale na pewno nie o tym, *co program w rzeczywistości robi i w jaki sposób działa!*

Wiadomo, że w praktyce żaden stolarz nie wypowiada się w ten sposób. Przedstawiony dialog powinien rozpocząć się inaczej.

**Pierwszy Stolarz:** Użyjemy połączeń wpustowych czy „na skos”?

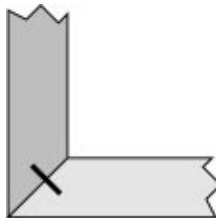
Od razu możemy zauważyć jakościową różnicę. Stolarz omawia teraz problem na wyższym poziomie abstrakcji, koncentrując się na problemie jakości rozwiązania. Unika zagłębiania się w szczegóły związane z realizacją konkretnego rozwiązania.

Mówiąc o połączeniu „na skos” ma on na myśli następującą charakterystykę tego rozwiązania:

- ♦ *prostotę* — wystarczy przyciąć łączone krawędzie pod kątem 45°, a następnie połączyć, zbicić lub skleić je razem (patrz: rysunek 5.2);

**Rysunek 5.2.**

Połączenie „na skos”



- ♦ *delikatność* — jako że to połączenie jest ono mniej wytrzymałe od połączenia „na wpust”, nie powinno być poddawane dużym obciążeniom;
- ♦ *niepozorność* — połączenie takie trudniej zauważyć niż połączenie „na wpust”.

Kiedy natomiast omawia połączenie „na wpust” (którego opis wykonania przedstawiłem wcześniej), to ma na myśli zupełnie inną charakterystykę rozwiązania:

- ♦ *wiekszy stopień złożoności* — wykonanie połączenia „na wpust” jest bardziej skomplikowane i dlatego jest ono droższe;
- ♦ *odporność na zmiany temperatury i wilgotności* — gdy czynniki te ulegają zmianie, drewno rozszerza się lub kurczy, a połączenie pozostaje trwałe;
- ♦ *niezależność od innych technologii łączenia* — połączenia „na wpust” nie wymagają zastosowania kleju;
- ♦ *estetykę* — porządnie wykonane połączenie „na wpust” cieszy oko swoim wzorem.

Innymi słowy — połączenie „na wpust” jest trwałe, odporne, estetyczne i skomplikowane (a tym samym — drogie).

Kiedy więc Pierwszy Stolarz zapytał:

Użyjemy połączeń wpustowych czy „na skos”?...

... to prawdziwą intencją tego pytania było:

Czy powinniśmy wykonać połączenie w kosztowny, ale za to trwałe i estetyczny sposób, czy raczej zrobimy to szybko i tanio?

Możemy więc powiedzieć, że rozmowa stolarzy odbywa się teraz na dwu poziomach: jednym, który jest związany z dosłownym znaczeniem ich wypowiedzi, oraz na drugim — tzw. „metapoziomie”, ukrytym dla laika i bogatszym w znaczenia. Poziom ten moglibyśmy nazwać poziomem „wzorców stolarskich”. Stanowi on odbicie zagadnień związanych z rozwiązaniami stosowanymi przez stolarzy.

W pierwszej wersji dialogu te istotne zagadnienia zostały zastąpione omówieniem szczegółów implementacji połączeń. W drugim przypadku — Pierwszy Stolarz zamierzał podjąć decyzję o wyborze rozwiązania kierując się jego jakością i kosztem.

Który z tych sposobów jest bardziej efektywny?

Zastosowanie wzorców z pewnością podnosi poziom abstrakcji naszego myślenia o problemie. W trakcie lektury dalszej części książki czytelnik przekona się, że dzięki temu uzyskujemy nowe, doskonalsze sposoby projektowania. Na tym właśnie polega potęga wzorców projektowych.

## Inne zalety studiowania wzorców projektowych

Z moich doświadczeń związanych z pracą w grupach wykorzystujących wzorce projektowe wynika, że ich zastosowanie nie tylko pomaga w opracowaniu poprawnych rozwiązań, ale także sprzyja edukacji członków zespołu. Mniej doświadczeni projektanci mają okazję obserwować swoich starszych kolegów skutecznie stosujących wzorce projektowe, co wzmacnia to ich motywację do zapoznania się z tematyką wzorców.

Większość wzorców projektowych sprzyja tworzeniu rozwiązań, które łatwo jest później modyfikować. Ponieważ na opracowanie rozwiązań oferowanych przez wzorce składa się suma doświadczeń i wysiłków wielu projektantów, to z reguły są to bardzo przemyślane struktury, które łatwiej radzą sobie ze wszelkimi zmianami niż rozwiązania utworzone ad hoc.

Studiowanie wzorców projektowych prowadzić może także do zdecydowanego pogłębienia rozumienia podstawowych zasad projektowania obiektowego. Zaobserwowałem to niezliczoną ilość razy podczas prowadzonych przeze mnie kursów. Rozpoczywałem je od krótkiego wprowadzenia do problemu obiektowości, po czym od razu przechodziłem do przedstawiania wzorców projektowych, które służyły mi także jako ilustracja zastosowania podstawowych koncepcji obiektowych (hermetyzacji, dziedziczenia i polimorfizmu). Pod koniec trzeciego dnia szkolenia uczestnicy zwykle doskonale już rozumieli istotę tych koncepcji, mimo że większość czasu poświęcona była wzorcom projektowym.

Banda Czworoga sugeruje kilka strategii służących do tworzenia poprawnych rozwiązań obiektowych. W szczególności należą do nich:

- ♦ projektowanie z wykorzystaniem interfejsów;
- ♦ preferowanie wykorzystania kompozycji zamiast dziedziczenia;
- ♦ odnajdowanie zmienności i hermetyzowanie jej.

Strategie te znajdują zastosowanie w przypadku większości wzorców projektowych przedstawionych w tej książce. Nawet jeśli czytelnik pozna tylko niektóre z nich, to powinno to wystarczyć, aby przekonać się o użyteczności przedstawionych strategii. Pomaga to później wykorzystać je w praktyce (nawet w przypadkach, w których wzorce projektowe nie znajdują bezpośredniego zastosowania).

Inną zaletą wzorców projektowych jest to, że pozwalają one tworzyć rozwiązania skomplikowanych problemów bez nadmiernie rozbudowanych hierarchii dziedziczenia. Nawet jeśli wzorce nie znajdują w nich czasami bezpośredniego zastosowania, to i tak uniknięcie wielopoziomowych hierarchii dziedziczenia zawsze prowadzi do doskonalszych rozwiązań.

## Podsumowanie

W rozdziale tym przedstawiłem to, czym są wzorce projektowe. Christopher Alexander napisał, że „wzorce stanowią rozwiązanie problemu w kontekście”. Wzorce stanowią zatem coś więcej niż tylko rodzaj szablonu oferującego gotowe rozwiązanie danego problemu. Są też sposobem opisu motywacji, ponieważ zawierają informację zarówno o prześladowającym nas problemie, jak i rozwiązaniu, które chcemy stworzyć.

Dokonałem także przeglądu powodów, dla których warto studiować wzorce projektowe. Studia takie pomagają projektantom:

- ♦ tworzyć sprawdzone, solidne rozwiązania powtarzających się problemów;
- ♦ uzyskać wspólną terminologię opisu problemu poprawiającą komunikację w zespole;
- ♦ myśleć o rozwiązaniu problemu na wyższym poziomie abstrakcji;
- ♦ przeanalizować jakość uzyskanego rozwiązania, a nie tylko jego działanie;
- ♦ usprawnić edukację zespołu jako całości i poszczególnych jego członków;
- ♦ zwiększyć możliwość wprowadzania modyfikacji w oprogramowaniu;
- ♦ tworzyć doskonalsze rozwiązania (nawet gdy nie wykorzystują one wzorców);
- ♦ znaleźć rozwiązania alternatywne do wykorzystujących rozbudowane hierarchie dziedziczenia.