Agnieszka Iżuk
Numer albumu: 43120
Kierunek: Magento PHP Developer
Link GitHub: https://github.com/student43120/Wzorce-projektowe.git

Spis Treści

Wzorce projektowe odnoszą się do wielu aspektów życia tj. sfery służbowej lub prywatnej, czego najprostszym przykładem jest osobowość Christophera Alexander ’a. Architekt jako pierwszy zbadał wzorce występujące w budynkach i środowiskach oraz opracował „język wzorców" do ich generowania. Struktura każdego ze wzorców sprowadza się do określenia problemu, następnie zebrania informacji w jaki sposób można go rozwiązać oraz ostatecznej decyzja wyboru najkorzystniejszego i postepowania według przyjętego schematu. W poniżej pracy skupimy się na metodach, które wykorzystywane są w programowaniu, czyli mówiąc szerzej w obszarze IT.

Programowanie obiektowe opiera się na operacjach i procedurach, które przeprowadzane są na obiektach. Obiekt uruchamia metodę, kiedy otrzyma żądanie (lub komunikat) od klienta. Zgłoszenie żądania to jedyny sposób na zmuszenie obiektu do uruchomienia operacji. Z kolei jej wywołanie to jedyny sposób na zmodyfikowanie wewnętrznych danych obiektu. Tak “zakapsułkowany” obiekt zostaje zabezpieczony i nie można bez-pośrednio uzyskać dostępu do stanu obiektu, a jego reprezentacja jest niewidoczna poza nim. W projektowaniu obiektowym trudność sprawia podział systemu na obiekty, ponieważ musimy brać pod uwagę m.in: kapsułkowanie, szczegóło-wość, zależności, elastyczność, wydajność, możliwość powtórnego wykorzystania. Wszystkie te aspekty wpływają na podział systemu i często proponowane rozwiązania są opozycyjne względem siebie. Dodatkową komplikacją jest fakt, że obiekty mogą znacznie różnić się między sobą pod względem wielkości i liczby.

Interfejsy są podstawowym elementem systemów obiektowych, dlatego nie można zażądać od obiektu wy-konania operacji z pominięciem interfejsu, nie określa on jednak implementacji obiektu. Dodatkowym mankamentem jest fakt, że na róż-nych obiektach żądania mogą być realizowane w inny sposób. Oznacza to, że dwa obiekty o zupełnie innej implementacji mogą mieć identyczny interfejs, dlatego wzorce projektowe określają też relacje między interfejsami.

Przykłady użycia wzorców projektowych:

- **Typ: Czynnościowy**

Wzorzec strategia bywa użyteczny, gdy klasa może zachowywać się różnie w zależności od potrzeb. Gdy wykonanie metody wymaga użycia różnych algorytmów, możemy zapisać je w postaci klas i używać dokładnie jednego z nich w danym czasie.

- **Typ: Strukturalny**

Wzorzec dekorator stosowany jest tam, gdzie dziedziczenie nie jest optymalnym sposobem rozszerzania funkcjonalności klasy. Polega na dołączeniu nowych atrybutów poprzez “opakowanie” obiektu bazowego obiektem zwanym dekoratorem.

- **Typ: Kreacyjny**

Celem wzorcu “Fabryka” jest dostarczenie interfejsu dla klas odpowiedzialnych za tworzenie konkretnego typu obiektów.

Główne zadania spełniane przez wzorce to m.in: przedstawianie zarysu powiązań pomiędzy klasami i obiektami, ułatwianie tworzenia, utrzymania i edycji kodu źródłowego, zmniejszanie kosztów utrzymania i rozwoju projektu oraz umożliwiają wizualizację rozwiązania problemu przed implementacją .

Podsumowując wzorców nie należy używać bez zastano-wienia. Często pozwalają one uzyskać elastyczność, możliwość wprowadzania zmian oraz zwiększają prawdopodobieństwo reużywalności wytworzonego rozwiązania. Musimy mieć, jednak na uwadze, że istnieje ryzyko skomplikowania projekt lub pogorszenia wydajności. Wzorce projektowe należy stosować tylko wtedy, kiedy większa elastyczność jest naprawdę potrzebna lub w przypadku braku doświadczenia developerów tj. team składający się z juniorów.