

Lekcja: Wzorce projektowe

Wzorce projektowe to sprawdzone rozwiązania typowych problemów, które można napotkać podczas projektowania oprogramowania. Możesz myśleć o nich jak o szablonach, które pomagają rozwiązywać określone problemy projektowe w bardziej uporządkowany sposób. Są one szczególnie przydatne, ponieważ pozwalają unikać ponownego wynajdywania koła i korzystać z doświadczeń innych programistów.

Wzorce projektowe, jakie znamy dzisiaj, w dużej mierze zyskały na popularności dzięki książce "Design Patterns: Elements of Reusable Object-Oriented Software", napisanej przez Ericha Gamma, Richarda Helm, Ralpha Johnsona i Johna Vlissidesa. Autorzy ci są często nazywani "Gangiem Czworga" (Gang of Four, GoF).

Początki koncepcji wzorców

Koncepcja wzorców w inżynierii oprogramowania została zainspirowana pracami Christophera Alexandra, architekta i teoretyka urbanistyki, który w latach 70. XX wieku wprowadził ideę wzorców projektowych w architekturze.

Alexander zauważył, że pewne rozwiązania projektowe były skutecznie stosowane w wielu różnych kontekstach i zaczął dokumentować te "wzorce" jako sposób na ułatwienie tworzenia dobrze zaprojektowanych budynków i miast.

Przeniesienie idei do inżynierii oprogramowania

W latach 80. i na początku lat 90. XX wieku, idea wzorców projektowych zaczęła być przenoszona do dziedziny inżynierii oprogramowania.

Wzorce projektowe zaczęły być rozumiane jako sprawdzone rozwiązania dla często występujących problemów projektowych w tworzeniu oprogramowania.

"Design Patterns: Elements of Reusable Object-Oriented Software"

Książka "Design Patterns", opublikowana w 1994 roku, jest uznawana za przełomową pracę w dziedzinie inżynierii oprogramowania.

Autorzy skatalogowali 23 wzorce projektowe, które uznali za fundamentalne dla projektowania oprogramowania zorientowanego obiektowo.

Książka ta nie tylko przedstawiała konkretne wzorce, ale także wprowadzała język do opisywania i rozumienia architektury oprogramowania.

Wpływ na branżę oprogramowania

Po opublikowaniu, książka szybko stała się bestsellerem i fundamentalnym źródłem wiedzy dla programistów i projektantów oprogramowania.

Wzorce projektowe pomogły w standardyzacji niektórych aspektów projektowania oprogramowania, co ułatwiło współpracę i komunikację między programistami.

W efekcie, wzorce projektowe, zdefiniowane i opisane przez Gang Czworga, stały się nieodłącznym elementem nowoczesnego projektowania oprogramowania. Pomagają one programistom i architektom oprogramowania w tworzeniu elastycznego, efektywnego i łatwego w utrzymaniu kodu.

Wzorce projektowe można podzielić na trzy główne kategorie:

Wzorce kreacyjne: Skupiają się na sposobach tworzenia obiektów. Pomagają one w zarządzaniu procesem tworzenia obiektów, zwłaszcza gdy standardowe podejścia, takie jak używanie konstruktorów, mogą być niewystarczające lub nieefektywne.

Wzorce strukturalne: Dotyczą one sposobów organizacji klas i obiektów w większe struktury. Pomagają one w zapewnieniu, że gdy klasy są razem łączone, całość jest większa niż suma jej części.

Wzorce behawioralne: Koncentrują się na efektywnej komunikacji i zarządzaniu relacjami między obiektami. Przykłady:

Wzorzec strukturalny: Singleton

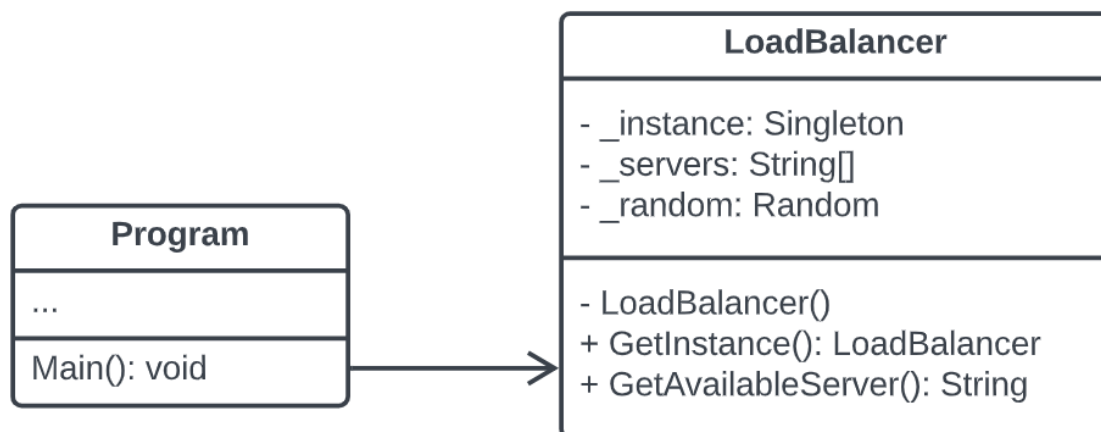
Singleton to wzorzec projektowy należący do kategorii wzorców kreacyjnych, a nie strukturalnych. Jego głównym celem jest zapewnienie, że dana klasa będzie miała tylko jedną instancję w całym programie oraz dostarczenie globalnego punktu dostępu do tej instancji.

Oto kluczowe aspekty wzorca Singleton:

Singleton gwarantuje, że z danej klasy istnieje tylko jedna instancja. Oznacza to, że niezależnie od tego, ile razy próbujesz utworzyć nowy obiekt tej klasy, zawsze otrzymasz tę samą instancję.

Globalny Punkt Dostępu:

Singleton zapewnia globalny dostęp do swojej jednej instancji. Oznacza to, że instancja jest dostępna z każdego miejsca w programie, co ułatwia dostęp do wspólnych zasobów, takich jak konfiguracja aplikacji, połączenia z bazą danych itp.



Zaimplementuj wzorzec Singleton na podstawie powyższego diagramu.

Wzorzec projektowy: Obserwator

Wzorzec projektowy Obserwator to wzorzec behawioralny, który jest używany do tworzenia projektu, w którym obiekty są informowane o zmianach stanu innych obiektów. Jest on szczególnie przydatny w sytuacjach, gdzie jeden obiekt (nazywany „subject” lub „observable”) musi automatycznie powiadamiać wielu innych obiektów (nazywanych „observers” lub „listeners”) o zmianach swojego stanu.

Podział na Obserwowanych i Obserwatorów:

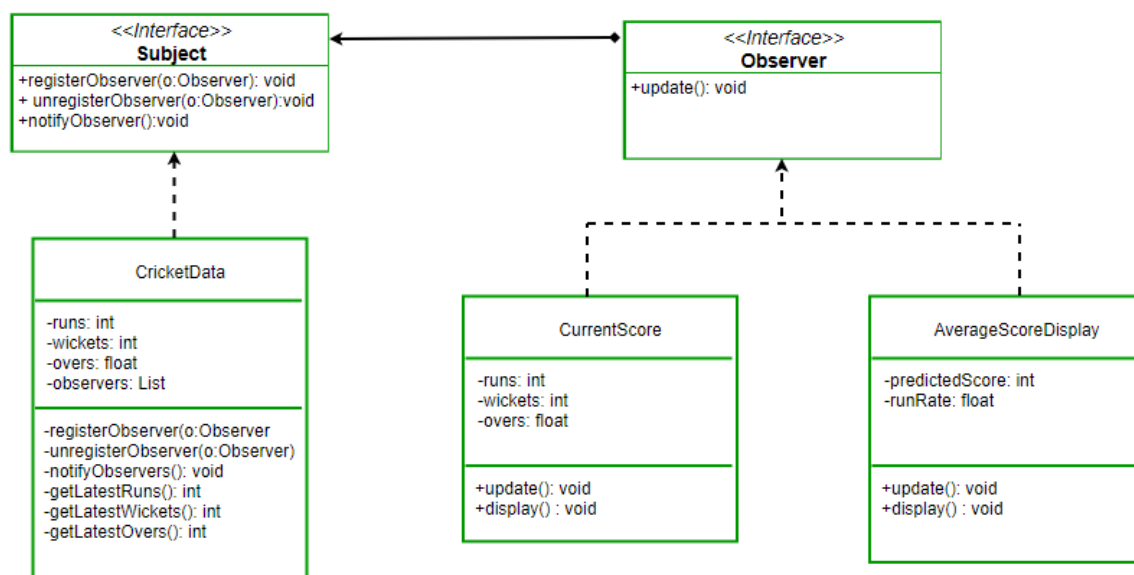
Wzorzec ten dzieli obiekty na dwie kategorie: obserwowanych (observables) i obserwatorów (observers). Obserwowany jest obiektem, który wysyła powiadomienia, gdy jego stan ulega zmianie. Obserwatorzy rejestrują się u obserwowanego i otrzymują powiadomienia o zmianach.

Zarejestrowanie i Powiadamianie:

Obserwatorzy rejestrują się do obserwowanego, aby otrzymywać powiadomienia. Gdy stan obserwowanego ulega zmianie, automatycznie powiadamia on wszystkich zarejestrowanych obserwatorów o tej zmianie.

Loose Coupling:

Wzorzec Obserwator promuje luźne powiązanie (loose coupling) między obiektami. Oznacza to, że obserwowany nie musi wiedzieć nic o obserwatorach poza tym, że implementują one określony interfejs.



Zaimplementuj wspomniany wzorzec na podstawie powyższego diagramu.