

Wzorce projektowe w inżynierii oprogramowania



Przeznaczenie szkolenia

Szkolenie dla osób chcących nauczyć się tworzenia oprogramowania opartego o dobre wzorce, polepszania i usprawnienia procesów wytwórczych oprogramowania.



Korzyści wynikające z ukończenia szkolenia

Dzięki zastosowaniu poruszanych zagadnień uczestnicy poznają uniwersalne i sprawdzone w praktyce rozwiązania często pojawiających się problemów projektowania systemów informatycznych, a tworzony kod aplikacji jest czytelny i zrozumiały dla innych uczestników projektu. Skraca się czas poświęcony na dodawanie nowych funkcjonalności, dzięki strukturze aplikacji otwartej nad rozbudowę i modyfikacje. Wzorce projektowe zapewniają łatwe rozbudowywanie systemu.



Oczekiwane przygotowanie słuchaczy

Umiejętność programowania w języku Java lub C++.



Język szkolenia

- Szkolenie: polski
- Materiały: polski



Czas trwania

3 dni / 21 godzin

Agenda szkolenia

1. Wprowadzenie do wzorców projektowych

- zasada pojedynczej odpowiedzialności (Single-Responsibility Principle)
- zasada otwarte-zamknięte (Open-Close Principle)
- zasada podstawiania
- zasada odwracania zależności
- segregacja interfejsów.

2. Wzorce kreacyjne:

- budowniczy (obiektyowy)
- fabryka abstrakcyjna (obiektyowy)
- metoda wytwórcza (klasowy)
- prototyp (obiektyowy)
- singleton (obiektyowy)

3. Wzorce strukturalne:

- adapter (klasowy oraz obiektyowy)
- dekorator (obiektyowy)
- fasada (obiektyowy)
- kompozyt (obiektyowy)
- most (obiektyowy)
- pełnomocnik (obiektyowy)
- pyłek (obiektyowy)

4. Wzorce czynnościowe:

- interpreter (klasowy)
- iterator (obiektyowy)
- łańcuch zobowiązań (obiektyowy)
- mediator (obiektyowy)
- metoda szablonowa (klasowy)
- obserwator (obiektyowy)
- odwiedzający (obiektyowy)
- pamiętka (obiektyowy)
- polecenie (obiektyowy)
- stan (obiektyowy)
- strategia (obiektyowy)
- zabór Zasobu Jest Inicjalizacją (obiektyowy).

5. Antywzorce w projektowaniu oprogramowania.
6. Refaktoryzacja kodu, a wykorzystanie wzorców projektowych
 - zmiana struktury programu w celu poprawy jakości kodu
 - wybrane aspekty refaktoryzacji kodu
 - zmiana nazwy zmiennej
 - zmiana nazwy metody
 - zmiana nazwy klasy
 - wydzielenie metody
 - wydzielenie zmiennej
 - odzwierciedlenie kroków algorytmu
 - kierunek wprowadzania wzorców
 - refaktoryzacja kodu z wykorzystaniem wzorców projektowych.