

Kod szkolenia: **PYTHON/DP**

Tytuł szkolenia: **Wzorce projektowe w języku Python**

Dni: 4

Opis:

Adresaci szkolenia:

Kurs jest przeznaczony dla programujących w języku Python na poziomie średnio zaawansowanym, oraz dla osób które chcą zapoznać się z zastosowaniem i implementowaniem wzorców projektowych. Prezentujemy na rzeczywistych przykładach zakres zastosowań wzorców projektowych w języku Python wraz z idiomami właściwymi tylko dla tego języka. Zakres zagadnień ćwiczeń wykonywanych w trakcie zajęć obejmuje także podejmowanie i weryfikowanie decyzji architektonicznych związanych z zastosowaniem konkretnych wzorców projektowych w rozbudowanych aplikacjach.

Wymagania:

- Umiejętność programowania w języku Python na poziomie średnio zaawansowanego programisty
- Znajomość podstaw analizy obiektowej
- Biegłe posługiwanie się wybranym środowiskiem IDE

Parametry szkolenia:

4*8 godzin (4*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

1. Wprowadzenie
 - Wartości dobrego oprogramowania
 - ojęcie jakości i jej rodzaje w oprogramowaniu
 - Proces wytwórczy w branży oprogramowania
2. Analiza i modelowanie obiektowe z wykorzystaniem notacji UML – wstęp
 - Diagram klas
 - Diagram sekwencji i komunikacji
3. Projektowanie obiektowe
 - Zasady projektowania obiektowego
 - Zasady S.O.L.I.D



- Zapachy kodu
- 4. Wzorce projektowe
 - Idea wzorców
 - Wzorce GoF
 - Inne wzorce - przegląd
 - Zagadnienia związane z refaktoryzacją – wstęp
- 5. Wzorce
 - Rodzaje wzorców i ich klasyfikacja
 - Techniki identyfikacji wzorców
- 6. Strukturalne wzorce projektowe
 - Adapter (klasowy i obiektowy)
 - Most (ang. Bridge) (obektowy)
 - Kompozyt (ang. Composite) (obektowy)
 - Dekorator (ang. Decorator) (obektowy)
 - Pyłek (ang. Flyweight) (obektowy)
 - Fasada (ang. Façade) (obektowy)
 - Pełnomocnik (ang. Proxy) (obektowy)
- 7. Warsztat analizy aplikacji z wykorzystaniem poznanych wzorców projektowych
- 8. Konstrukcyjne wzorce projektowe
 - Metoda wytwórcza (ang. Factory Method) (klasowy)
 - Fabryka Abstrakcyjna (ang. Abstract Factory) (obektowy)
 - Budowniczy (ang. Builder) (obektowy)
 - Prototyp (ang. Prototype) (obektowy)
 - Singleton (obektowy)
- 9. Warsztat analizy aplikacji z wykorzystaniem poznanych wzorców projektowych
- 10. Behavioralne wzorce projektowe
 - Łańcuch zobowiązań (ang. Chain of Responsibility) (obektowy)
 - Polecenie (ang. Command) (obektowy)
 - Interpreter (ang. Interpreter) (klasowy)
 - Interator (obektowy)
 - Mediator (ang. Mediator) (obektowy)
 - Pamiętka (ang. Memento) (obektowy)
 - Obserwator (ang. Observer) (obektowy)
 - Stan (ang. State) (obektowy)
 - Strategia (ang. Strategy) (obektowy)
 - Metoda szablonowa (ang. Template Method) (klasowy)
 - Odwiedzający (ang. Visitor) (obektowy)
- 11. Warsztat analizy aplikacji z wykorzystaniem poznanych wzorców projektowych
- 12. Idiomy języka programowania
 - Pojęcie idiomu językowego
 - Wzorce i idiomy specyficzne dla języka Python
 - Wzorzec EFAP (ang. It's easier to ask for forgiveness than permission)
 - Wzorzec Metaklasy
 - Borg
 - Klasa domieszkowa w języku Python (ang. Mixin)