

Kod szkolenia: **CPP/DP**

Tytuł szkolenia: **Wzorce projektowe w języku C++**

Dni: 4

## Opis:

Uczestnicy szkolenia zapoznają się z klasycznymi wzorcami projektowymi GoF (ang. Gang of Four) oraz metodami refaktoryzacji do poziomu wzorców. W ramach zajęć, zapoznujemy się z siłami wpływającymi na wybór wzorca oraz kosztami jego stosowania. Prezentowane są przykłady prawidłowego i nieprawidłowego implementowania każdego ze wzorców, a w ramach ćwiczeń, uczestnicy mają możliwość bezpiecznego zapoznania się z konsekwencjami stosowania każdego ze wzorców.

Prezentujemy także nowoczesne sposoby projektowania i tworzenia aplikacji z użyciem standardu C++11. Jako techniki tworzenia aplikacji obiektowej, prezentowane są także klasy wytycznych. Szkolenie przeznaczone jest dla średnio zaawansowanych programistów, którzy chcą podnieść swoje kompetencje związane z wyborem i implementacją poprawnych rozwiązań aplikacyjnych. Na zajęciach posługujemy się technologiami otwartymi z dostępem do kodu źródłowego.

Zajęcia prowadzone są przez doświadczonych praktyków, którzy na co dzień stosują technologie związane z językiem C++ w rzeczywistych projektach. W trakcie zajęć używamy technologii i środowisk otwartych.

## Zakres szkolenia

Szkolenie obejmuje:

- wzorce projektowe GoF,
- style implementacji każdego ze wzorców,
- refaktoryzowanie kodu do wzorca projektowego.

## Wymagania

Od uczestników szkolenia wymaga się:

- posługiwania się wybranym środowiskiem IDE (Eclipse, NetBeans, vim),
- znajomości koncepcji programowania obiektowego, systemowego i aplikacyjnego,
- znajomości narzędzi do tworzenia aplikacji z rodziny gcc i binutils,
- umiejętności posługiwania się narzędziem debuggera,

- znajomości koncepcji związanych z programowaniem w języku C++ na poziomie podstawowym i średnio zaawansowanym.

Dodatkowo mile widziana jest również znajomość aspektów sprzętowych platformy x86 (przydzielanie i zarządzanie pamięcią, alokacja rejestrów, konwencje wywołań, optymalizacje).

## Adresaci szkolenia

Szkolenie adresowane jest do:

- średnio zaawansowanych programistów programujących w języku C i C++ w środowisku GNU/Linux lub MS Windows,
- osób znających w podstawowym zakresie system szablonów STL,
- programistów tworzących oprogramowanie na platformie systemowej GNU/Linux lub MS Windows,
- programistów tworzących aplikacje głównego nurtu oraz aplikacje sieciowe.

## Cel szkolenia

Głównymi celami w procesie szkoleniowym są:

- zapoznanie się klasycznymi wzorcami GoF i konsekwencjami ich stosowania,
- implementowanie wzorców z użyciem wydajnych technik,
- zapoznanie się z zaletami i wadami każdego ze wzorców,
- zapoznanie się z zasadami refaktoryzacji do wzorców projektowych.

## Umiejętności zdobywane podczas zajęć

Uczestnicy szkolenia po jego zakończeniu zdobędą następujące umiejętności:

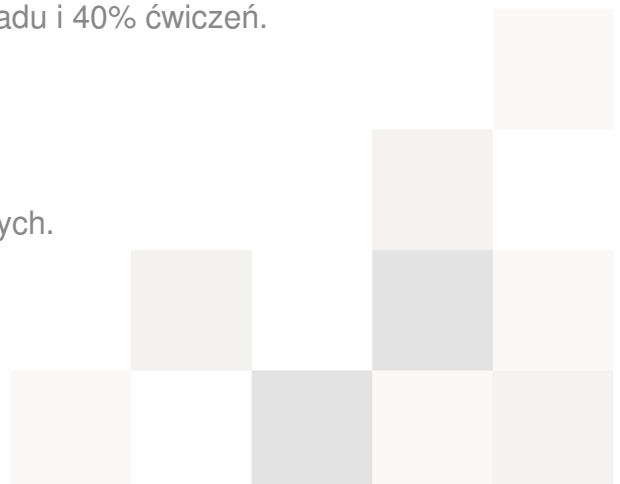
- identyfikowania i implementowania wzorców projektowych GoF we własnych projektach,
- umiejętność określania kosztów i zysku zastosowania wybranego wzorca,
- określanie kontekstu poprawnej implementacji wybranego wzorca projektowego.

## Parametry szkolenia

Szkolenie trwa 4 dni. Szkolenie obejmuje 60% wykładu i 40% ćwiczeń.

## Program szkolenia:

1. Wprowadzenie do wzorców projektowych.
  - I. Rola i znaczenie wzorców projektowych.
  - II. Pojęcie reifikacji wzorca.
  - III. Klasy wzorców projektowych.



- IV. Siły wpływające na wybór wzorca.
  - V. Proponowane rozwiązania.
  - VI. Kontekst rozwiązania.
  - VII. Ogólne zakresy zastosowań wzorców projektowych.
  - VIII. Wzorce GoF a inne wzorce.
  - IX. Klasyfikacja i rodzaje wzorców projektowych.
2. Strukturalne wzorce projektowe.
    - I. Adapter (klasowy i obiektowy).
    - II. Most (ang. Bridge ) (obektowy).
    - III. Kompozyt (ang. Composite ) (obektowy).
    - IV. Dekorator (ang. Decorator ) (obektowy).
    - V. Pyłek (ang. Flyweight ) (obektowy).
    - VI. Fasada (ang. Façade ) (obektowy).
    - VII. Pełnomocnik (ang. Proxy ) (obektowy).
  3. Konstrukcyjne wzorce projektowe.
    - I. Metoda wytwórcza (ang. Factory Method ) (klasowy).
    - II. Fabryka Abstrakcyjna (ang. Abstract Factory ) (obektowy).
    - III. Budowniczy (ang. Builder ) (obektowy).
    - IV. Prototyp (ang. Prototype ) (obektowy).
    - V. Singleton (obektowy).
  4. Behavioralne wzorce projektowe.
    - I. Łańcuch zobowiązań (ang. Chain of Responsibility ) (obektowy).
    - II. Polecenie (ang. Command ) (obektowy).
    - III. Interpreter (ang. Interpreter ) (klasowy).
    - IV. Iterator (obektowy).
    - V. Mediator (ang. Mediator ) (obektowy).
    - VI. Pamiętka (ang. Memento ) (obektowy).
    - VII. Obserwator (ang. Observer ) (obektowy).
    - VIII. Stan (ang. State ) (obektowy).
    - IX. Strategia (ang. Strategy ) (obektowy).
    - X. Metoda szablonowa (ang. Template Method ) (klasowy).
    - XI. Odwiedzający (ang. Visitor) (obektowy).
  5. Idiomy języka C++.
    - I. Zbiór zasobu jest inicjalizacją (ang. RAII - Resource Acquisition Is Initialization ) (obektowy).
    - II. Zdziwiająco często powracający wzorzec – wytyczne.
    - III. Prywatna implementacja – PIPML (ang. Prlvate IMPLemetation ).
    - IV. Klasy wytycznych.
  6. Refaktoryzacja z użyciem wzorców – przegląd.
    - I. Wzorce refaktoryzacji.
    - II. Rola wzorców projektowych w działaniach refaktoryzacyjnych.

