

Kod szkolenia: **C/OOBP**

Tytuł szkolenia: **Dobre praktyki programowania obiektowego z zastosowaniem w języku C++**

Dni: 3

Opis:

Adresaci Szkolenia:

Szkolenie skierowane jest do programistów, projektantów i architektów, którzy pragną poznać, pogłębić lub usystematyzować swoje wiadomości na temat dobrych praktyk, zasad i wzorców programowania obiektowego w C++.

Cel szkolenia:

Celem szkolenia jest zapoznanie i przećwiczenie dobrych praktyk, zasad i wzorców programowania obiektowego w C++, w tym:

- Zasad SOLID
- Wybranych wzorców GRASP
- Wybranych wzorców Bandy Czworka (Gang of Four)
- Wybranych idiomów programowania w C++

Wymagania:

Uczestnicy muszą posiadać umiejętność programowania w języku C++.

Parametry szkolenia:

3*8 godzin (3*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

1. Zasady SOLID
 - I. Zasada jednej odpowiedzialności (The Single Responsibility Principle)
 - i. Wstęp, definicja, odpowiedzialności
 - ii. Warsztaty problemowe
 - iii. Zasady refaktoringu
 - iv. Wybrane wzorce Bandy Czworka (GoF) w kontekście zasady

- II. Zasada otwarte/zamknięte (The Open/Closed Principle)
 - i. Wstęp, definicja, odpowiedzialności
 - ii. Warsztaty problemowe
 - iii. Zasady refaktoringu
 - iv. Wybrane wzorce Bandy Czworga (GoF) w kontekście zasady
 - III. Zasada podstawienia Liskov (The Liskov Substitution Principle)
 - i. Wstęp, definicja, odpowiedzialności
 - ii. Warsztaty problemowe
 - iii. Zasady refaktoringu
 - iv. Wybrane wzorce Bandy Czworga (GoF) w kontekście zasady
 - IV. Zasada segregacji interfejsów (The Interface Segregation Principle)
 - i. Wstęp, definicja, odpowiedzialności
 - ii. Warsztaty problemowe
 - iii. Zasady refaktoringu
 - iv. Wybrane wzorce Bandy Czworga (GoF) w kontekście zasady
 - V. Zasada odwrócenia zależności (The Dependency Inversion Principle)
 - i. Wstęp, definicja, odpowiedzialności
 - ii. Warsztaty problemowe
 - iii. Zasady refaktoringu
 - iv. Wybrane wzorce Bandy Czworga (GoF) w kontekście zasady
2. Wzorce Bandy Czworga (GoF)
- I. Wzorce konstrukcyjne
 - i. Wzorce: Singleton, Factory Method, Abstract Factory, Prototype, Builder
 - ii. Warsztaty problemowe
 - II. Wzorce strukturalne
 - i. Wzorce: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
 - ii. Warsztaty problemowe
 - III. Wzorce behawioralne
 - i. Wzorce: Chain of Responsibility, Command, Iterator, Mediator, Memento, Oserwer, State, Strategy, Template Method, Visitor
 - ii. Warsztaty problemowe
3. Wzorce GRASP
- I. Low Coupling, High Cohesion
 - II. Information Expert, Creator, Controller
 - III. Polymorphism, Pure Fabrication
 - IV. Indirection, Protected Variations
 - V. Warsztaty problemowe
4. Idiomy programowania w C++
- I. Zarządzanie zasobami
 - i. Idiomy: Release Return, Move Constructor, Resource Acquisition Is Initialization, Scope Guard
 - ii. Warsztaty problemowe
 - II. Zarządzanie pamięcią
 - i. Idiomy: Intrusive Reference Counting, Non-intrusive Reference

- Counting, Const auto_ptr, Checked Delete, Concrete Data Type, Copy and Swap
- ii. Warsztaty problemowe
- III. Optymalizacja pamięci i przetwarzania
 - i. Idiomy: Shrink to Fit, Clear and Minimize, Non-throwing Swap , Erase-Remove, Boost Mutant, Computational Constructor, Copy on Write, Empty Base Optimization
 - ii. Warsztaty problemowe
- IV. Typy i bezpieczeństwo typów
 - i. Idiomy: Type Safe Enum, Type Selection, Type Generator, Traits, Capability Query, Coercion by Member Template, Mixin from Above, Int to Type
 - ii. Warsztaty problemowe
- V. Konstrukcja i inicjalizacja
 - i. Idiomy: Construction Tracker, Construct on First Use, Base from Member, Runtime Static Initialization Idiom
 - ii. Warsztaty problemowe
- VI. Polimorfizm
 - i. Idiomy: Interface Class, Inner Class, Virtual Friend Function, Polymorphic Exception, Virtual Constructor, Calling Virtuals During Initialization, Polymorphic Value Types
 - ii. Warsztaty problemowe

