

Kod szkolenia: **J/SOLID**

Tytuł szkolenia: **Wzorce Solid, grasp, GoF - praktycznie**

Wzorce Solid, grasp, GoF - praktycznie

Dni: 5

## Opis:

### Adresaci szkolenia

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

### Cel szkolenia

- Zapoznanie, zrozumienie i przećwiczenie dobrych praktyk, zasad i wzorców programowania obiektowego w Javie
- Nabranie umiejętności oceny jakości kodu oraz alternatywnych rozwiązań
- Poznanie zasad Clean Code
- Poznanie zasad SOLID
- Poznanie wzorców GRASP
- Poznanie wybranych wzorców GoF (różna ilość w zależności od czasu trwania szkolenia)

### Mocne strony szkolenia

Szkolenie to cechuje duży nacisk na praktyczną stronę poznawanych zasad.

### Wymagania

Uczestnik szkolenia powinien posiadać podstawowe doświadczenie w programowaniu obiektowym za pomocą języka Java oraz podstawową znajomość UML.

### Parametry szkolenia

5\*8 godzin (5\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów - 60%).

### Program szkolenia:



1. Wprowadzenie
  - Problemy i ich źródła
2. Jakość kodu źródłowego
  - Czym jest jakość, różne jej wymiary
  - Dlaczego jakość jest ważna
  - Jak mierzyć jakość
  - Statyczna analiza kodu
  - Proces tworzenia oprogramowania wysokiej jakości
3. Czytelny kod (ang. Clean Code)
  - Nazewnictwo pakietów, klas, metod
  - Komentarze
  - Klasy
  - Metody
  - Struktury danych
  - Obsługa wyjątków i błędów
4. Wzorce GRASP w oparciu o warsztaty problemowe
  - Low Coupling, High Cohesion
  - Information Expert
  - Creator
  - Pure Fabrication
  - Polymorphism
  - Indirection
  - Protected Variations
  - Controller
5. Zasady SOLID w oparciu o warsztaty problemowe.
  - The Single Responsibility Principle
  - The Open/Closed Principle
  - The Liskov Substitution Principle
  - The Interface Segregation Principle
  - The Dependency Inversion Principle
6. Wzorce GoF w oparciu o warsztaty problemowe.
  - Wzorce konstrukcyjne:
    - Builder,
    - Prototype,
    - Factory Method,
    - Abstract Factory,
    - Singleton
  - Wzorce strukturalne:
    - Facade,
    - Proxy,
    - Composite,
    - Adapter,
    - Decorator,
    - Bridge,
    - Flyweight
  - Wzorce behawioralne:



- Command,
- Observer,
- State,
- Strategy,
- Chain of Responsibility,
- Mediator,
- Visitor,
- Template Method,
- Interpreter,
- Iterator,
- Memento

## 7. Inne popularne wzorce (slajdy trzeba dopisac)

- Null Object
- Optional

