Projektowanie Obiektowe Oprogramowania 2022

Wykład skierowany jest do przyszłych architektów i projektantów systemów informatycznych oraz do wszystkich programistów zainteresowanych udoskonaleniem swojego warsztatu. Celem wykładu jest zapoznanie studentów z kanonem współczesnych narzędzi w zakresie projektowania obiektowego oprogramowania.

1 Wymagania

Wymagany ukończony kurs języka Java lub C# - wszystkie przykłady będą bazowały na języku C#, a większość zadań studenci będą rozwiązywali w językach C#, Java lub Scala (do wyboru). Inne języki proszę skonsultować z prowadzącymi grupy ćwiczeniowe.

2 Plan wykładu

Analiza obiektowa (1)
Zarys meta metodyki wytwarzania oprogramowania, Unified Process
Zbieranie wymagań funkcjonalnych i niefunkcjonalnych, FURPS, S.M.A.R.T.
Projektowanie analityczne – przypadki użycia
Projektowanie analityczne – modele pojęciowe
Projektowanie analityczne – mapy procesów biznesowych

Przypomnienie języka UML (1)
Diagramy klas, obiektów, stanów, czynności, sekwencji, komunikacji

Projektowanie obiektów i przydział odpowiedzialności (1)
Zasady SOLID I GRASP

Wzorce projektowe (5)
Przegląd wzorców projektowych GoF, m.in. Factory, Singleton, Adapter, Decorator, Builder, Interpreter, Bridge, Visitor, Memento, Mediator, Observer, Event Aggregator, Chain of Responsibility, State, Strategy, Template Method

Wzorce aplikacyjne (5)
Object-Relational Mapping
Repository
Model-View-Controller, Model-View-Presenter
Inversion of Control/Dependency Injection
Mock Object

Wzorce architektury (3)
Enterprise Single Sign-on
SOA, Enterprise Service Bus
Command-Query Responsibility Separation

O czym nie będzie mowy na wykładzie
Zarządzanie projektami informatycznymi (IO)
Analiza czasochłonności (IO)
Szczegóły operacyjne metodyk wytwarzania oprogramowania (IO)
Serwery kontroli wersji, Continuous Integration (wiele innych zajęć)

3 Literatura podstawowa

1. Wrycza, Marcinkowski, Wyrzykowski - Język UML 2.0 w modelowaniu systemów informatycznych
2. Evans – Domain Driven Design
3. Fowler - Refactoring: Improving the Design of Existing Code
4. Gamma, Helm, Johnson, Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software
5. B.Martin, M.Martin. Programowanie zwinne: zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C#
6. Larman - UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji (ciekawostka – polski tytuł)
7. Fowler - Patterns of Enterprise Application Architecture
8. Microsoft Patterns & Practices - Application Architecture Guide 2.0

4 Literatura uzupełniająca

1. Seeman - Dependency Injection in .NET
2. Hohpe, Wolf - Enterprise Integration Patterns
3. Betts, Dominguez et al. - Exploring CQRS and Event Sourcing

Wiktor Zychla

2022.03.01