**Nazwa przedmiotu:**

Projektowanie oprogramowania

**Koordynator przedmiotu:**

dr inż. Michał Śmiałek, michal.smialek@ee.pw.edu.pl, +48222345387

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

7 / rok ak. 2009/2010

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 0h |
| Projekt:  | 30h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Podstawy inżynierii oprogramowania, Modelowanie oprog. w języku UML, Inżynieria wymagań oprog.

**Limit liczby studentów:**

**Cel przedmiotu:**

**Treści kształcenia:**

Wykład
1. Od wymagań do architektury. Co to jest architektura oprogramowania? Architektury komponentowe. Architektury w cyklu iteracyjnym. Architektury dla systemów rozproszonych. Architektura realizująca wymagania oprogramowania.
2. Statyczne modele architektoniczne. Komponenty i diagramy komponentów. Komponenty i architektury warstwowe. Diagramy montażowe. Mapowanie wymagań na statyczne modele architektoniczne.
3. Dynamiczne modele architektoniczne. Dynamika działania komponentów. Diagramy sekwencji na poziomie architektonicznym. Linie życia i komunikaty. Mapowanie wymagań na dynamiczne modele architektoniczne.
4. Techniki modelowania architektonicznego. Architektura a technologie programistyczne. Strategie dla szkieletów architektonicznych. Strategie modelowania dynamiki systemu.
5. Od architektury do projektu szczegółowego. Jak zaprojektować i zaimplementować szczegóły architektoniczne? Realizacja interfejsów. Projekt szczegółowy w cyklu iteracyjnym. Organizowanie modeli dla programistów.
6. Statyczne modele projektowe. Semantyka projektowych diagramów klas. Interfejsy, klasy, atrybuty, operacje. Relacje między klasami. Diagramy klas a kod.
7. Dynamiczne modele projektowe. Diagramy sekwencji na poziomie szczegółowego projektu. Impelementacja dynamiki działania interfejsów. Diagramy sekwencji a kod.
8. Techniki modelowania dla projektu szczegółowego. Projekt
Stworzenie pełnego modelu architektonicznego oraz projektu szczegółowego. Na projekt składają się: realizacja przypadków użycia, realizacja słownika w strukturze systemu, realizacja interfejsów, projekt struktury komponentów. Projekt wykonywany jest w zespołach po ok. 3 osoby. Każdy zespół tworzy projekt niedużego systemu oprogramowania (ok. 15-20 przypadków użycia).

**Metody oceny:**

brak

**Egzamin:**

**Literatura:**

1. Michał Śmiałek – Zrozumieć UML 2.0. Metody modelowania obiektowego, Helion, 2005; 2. Ian Sommerville – Inżynieria oprogramowania, WNT, 2003; 3. Roger S. Pressman – Praktyczne podejście do inżynierii oprogramowania, WNT, 2004

**Witryna www przedmiotu:**

**Uwagi:**

## Efekty przedmiotowe