

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2015/2016

Wydział Fizyki, Matematyki i Informatyki

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: I

Stopień studiów: II

Specjalności: Grafika komputerowa i multimedia dla inżynierów, Informatyka stosowana dla inżynierów, Inżynieria obliczeniowa dla inżynierów, Teleinformatyka dla inżynierów

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Zaawansowane techniki programowania
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	
KOD PRZEDMIOTU	WFMiI I oIIS B1 15/16
KATEGORIA PRZEDMIOTU	Przedmioty podstawowe
LICZBA PUNKTÓW ECTS	5.00
SEMESTRY	1

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	SEMINARIUM	PROJEKT
1	30	0	30	0	0	0

3 CELE PRZEDMIOTU

Cel 1 Zapoznanie słuchaczy ze stanem obecnym i przewidywanymi kierunkami rozwoju w zakresie metod tworzenia (implementacji) i utrzymania (modyfikacji, adaptacji) systemów oprogramowania w całym cyklu życia.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

- 1 Znajomość obiektowych technik analizy, projektowania i implementacji systemów oprogramowania. Podstawowa znajomość notacji UML. Znajomość języka programowania Java (ewentualnie C#)

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Rozwój kompetencji w zakresie współczesnych platform tworzenia systemów oprogramowania (J2EE, .Net)

EK2 Wiedza Rozwój kompetencji w zakresie programowania obiektowego i aspektowego

EK3 Umiejętności Opanowanie umiejętności praktycznego tworzenia rozwiązań komponentowych

EK4 Umiejętności Opanowanie umiejętności stosowania typowych wzorców architektury w procesie tworzenia komponentowych systemów oprogramowania

6 TREŚCI PROGRAMOWE

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Proste rozwiązania komponentowe realizowane w technologii servlet'ów. Przykład rozwiązania wymagającego zastosowania więcej niż jednego komponentu, realizowany w technologii serlet'ów.	6
L2	Obiekt połączenia (connection), obiekt sesji, funkcje kontrolera (obsługa stanu aplikacji)	2
L3	Wstrzykiwanie zależności, przykłady wykorzystania	2
L4	Proste komponenty EJB. Beans sesyjne bezstanowe oraz z pełną kontrolą stanu.	6
L5	Proste rozwiązania wymagające komunikacji programu klienta z warstwą logiki aplikacyjnej realizowaną w technologii EJB.	4
L6	Proste rozwiązania wymagające współpracy komponentów realizowanych w technologii servlet'ów z równoczesnym wykorzystaniem komponentów EJB w warstwie logiki aplikacyjnej	4
L7	Interfejs zdalny. Komunikacja pomiędzy komponentami EJB w systemie rozproszonym.	2
L8	Entity beans - przykłady wykorzystania w warstwie integracji.	4

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Java zalety i wady. Technologie aplety, serwety, EJB. J2EE koncepcja i charakterystyka platformy. Architektura fizyczna rozwiązań. Architektura oprogramowania. Rozproszenie danych i aplikacji. Niezawodność, skalowalność elastyczność rozwiązań. Wzorce projektowe koncepcja i cele stosowania. Wzorce konstrukcyjne strukturalne i czynnościowe. Wzorce architektury. Architektury trójwarstwowe i wielowarstwowe. Model, widok, kontroler (MVC). Architektury zorientowane na usługi. Wzorce projektowe w warstwie prezentacji (front controler, view dispatcher). Wzorce warstwy logiki aplikacyjnej (application service, session facade, bussines delegate. Warstwa integracji. Efektywność rozwiązań a stosowane praktyki kodowania. Technologia JSP. JavaBeans oraz Enterprise JB. Rola kontenera EJB, oferowane usługi. Pojęcie sesji. Rozwiązania bezstanowe oraz z kontrolą stanu. Przykłady rozwiązań. Transakcyjność w rozwiązaniach rozproszonych. Elementy integracji aplikacyjnej.	6
W2	Architektura fizyczna rozwiązań. Architektura oprogramowania. Rozproszenie danych i aplikacji. Niezawodność, skalowalność elastyczność rozwiązań.	6
W3	Wzorce architektury. Architektury trójwarstwowe i wielowarstwowe. Model, widok, kontroler (MVC). Architektury zorientowane na usługi. Wzorce projektowe w warstwie prezentacji (front controler, view dispatcher). Wzorce warstwy logiki aplikacyjnej (application service, session facade, bussines delegate. Warstwa integracji.	6
W4	Efektywność rozwiązań a stosowane praktyki kodowania. Technologia JSP. JavaBeans oraz Enterprise JB. Rola kontenera EJB, oferowane usługi. Pojęcie sesji. Rozwiązania bezstanowe oraz z kontrolą stanu. Przykłady rozwiązań. Transakcyjność w rozwiązaniach rozproszonych. Elementy integracji aplikacyjnej.	6
W5	Lekkie metodyki tworzenia oprogramowania - zasady eXtreme Programming. Tworzenie oprogramowania sterowane testami. Refactoring kodu - wprowadzenie.	6

7 NARZĘDZIA DYDAKTYCZNE

N1 Ćwiczenia laboratoryjne

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	60
Konsultacje przedmiotowe	0
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	90
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	0
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	150
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	5.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Sprawozdanie z ćwiczenia laboratoryjnego

OCENA PODSUMOWUJĄCA

P1 Egzamin pisemny

P2 Egzamin ustny

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	—
NA OCENĘ 3.0	Wiedza podstawowa o zasadach projektowania i implementacji systemów komponentowych z wykorzystaniem przynajmniej jednej z platform (.Net, J2EE), znajomość podstawowych koncepcji i technologii, rozumienie zagadnień technicznych związanych ze skalowalnością, bezpieczeństwem, wysoką dostępnością.
NA OCENĘ 3.5	Zdolność do samodzielnego analizowania i doboru właściwych rozwiązań technologicznych w oparciu o otrzymaną specyfikację wymagań.

NA OCENĘ 4.0	Zdolność do samodzielnego wyspecyfikowania wymagań, analizy oraz doboru odpowiedniej technologii realizacji w oparciu o otrzymane założenia techniczne dotyczące wymaganych do osiągnięcia parametrów wydajnościowych, niezawodnościowych, funkcjonalnych.
NA OCENĘ 4.5	Zdolność do samodzielnego wyboru właściwych rozwiązań projektowych w zakresie transakcyjności (w tym transakcji rozproszonych) oraz metod i środków komunikacji pomiędzy komponentami.
NA OCENĘ 5.0	Zdolność do samodzielnego opracowania zarysu projektu systemu komponentowego z uwzględnieniem zagadnień bezpieczeństwa aplikacyjnego.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	—
NA OCENĘ 3.0	Znajomość budowy modelu obiektowego systemu rozproszonego, z uwzględnieniem wymagań dotyczących możliwości realizacji w oparciu o podejście komponentowe, z wykorzystaniem jednej z poznanych technologii.
NA OCENĘ 3.5	Znajomość podstawowych reguł budowy rozwiązań rozproszonych w warstwie aplikacji, zasad doboru proponowanej technologii realizacji systemu komponentowego, zasad doboru serwera aplikacyjnego oraz metod implementacji w przyjętym środowisku.
NA OCENĘ 4.0	Zdolność do uzasadnienia przyjętych rozwiązań projektowych oraz krytycznego porównania podstawowych cech wybranej technologii realizacji z co najmniej jednym ze znanych rozwiązań alternatywnych.
NA OCENĘ 4.5	Umiejętność wskazania elementów rozwiązania uzasadniających zastosowanie podejścia aspektowego oraz uzasadnienia ewentualnych korzyści wynikające z przyjęcia tej techniki implementacji.
NA OCENĘ 5.0	Umiejętność wyboru właściwych rozwiązań technologicznych (.Net, J2EE) oraz stworzenia zarysu projektu systemu komponentowego z wykorzystaniem uwzględnieniem rozproszenia w warstwie danych.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	—
NA OCENĘ 3.0	Znajomość podstawowych wzorców architektury systemów komponentowych, zakresu ich stosowalności, praktycznych metod implementacji.
NA OCENĘ 3.5	Umiejętność identyfikacji obszarów rozwiązania wskazujących na konieczność zastosowania standardowych wzorców projektowych, praktyczna umiejętność implementacji z wykorzystaniem wskazanego do zastosowania wzorca projektowego.
NA OCENĘ 4.0	Praktyczna znajomość zasad projektowania aplikacji, umiejętność poprawnej implementacji przyjętych rozwiązań z wykorzystaniem jednego z języków obiektowych.

NA OCENĘ 4.5	Umiejętność praktycznego stosowania zasad implementacji z wykorzystaniem techniki 'od testu do programu (test driven development) oraz zasad refactoringu tworzonego kodu.
NA OCENĘ 5.0	Umiejętność samodzielnego zaprojektowania kompletnego rozwiązania, zdolność do uzasadnienia przyjętych rozwiązań projektowych, umiejętność dyskusji rozwiązań alternatywnych.
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 2.0	–
NA OCENĘ 3.0	Znajomość podstawowych wzorców architektury systemów rozproszonych, zakresu ich stosowalności, praktycznych metod implementacji.
NA OCENĘ 3.5	Umiejętność praktycznego określenia właściwych rozwiązań w zakresie architektury aplikacji, z uwzględnieniem wymagań wynikających z transakcyjności, skalowalności, wysokiej dostępności oraz zagadnień bezpieczeństwa aplikacyjnego
NA OCENĘ 4.0	Praktyczna umiejętność dekompozycji modelu obiektowego umożliwiającej skuteczną implementację w modelu trójwarstwowym lub wielowarstwowym. Umiejętność wyodrębnienia co najmniej elementów warstwy prezentacji oraz warstwy integracji w celu precyzyjnego określenia granic warstwy logiki biznesowej.
NA OCENĘ 4.5	Praktyczna umiejętność realizacji warstwy integracji w przypadku tworzenia rozwiązań z rozproszonym repozytorium danych.
NA OCENĘ 5.0	Praktyczna umiejętność doboru koncepcji rozwiązania (SOA, MVC) oraz dyskusji cech proponowanego rozwiązania w stosunku do znanych rozwiązań alternatywnych.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1		Cel 1	L1 L2 L3 L4 L5	N1	F1 P1 P2
EK2		Cel 1	L1 L2 L3 L4 L5	N1	F1 P1 P2
EK3		Cel 1	L1 L2 L3 L4 L5 L6 L7 L8	N1	F1
EK4		Cel 1	L1 L2 L3 L4 L5 L6 L7 L8	N1	F1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] **GoF** — *Design Patterns Elements Of Reusable Object Oriented Software*, —, 2000, —
- [2] **GoF** — *Design Patterns with examples using Java and UML2*, —, 2000, —
- [3] **James William Cooper** — *Java. Wzorce projektowe*, —, 2001, Helion
- [4] **Allan Shalloway, James R. Trott** — *Projektowanie zorientowane obiektowo*, —, 2001, Helion

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Jerzy Jaworowski (kontakt: jrj@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr inż. Jerzy Jaworowski (kontakt: jrj@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejsowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....