



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Architektury zorientowane na usługi [S2Inf1-TPD>AZU]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/2

Studia w zakresie (specjalność)
Technologie przetwarzania danych

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
30

Laboratorium
15

Inne
0

Ćwiczenia
0

Projekty/seminaria
15

Liczba punktów ECTS

5,00

Koordynatorzy

dr inż. Juliusz Jezierski
juliusz.jezierski@put.poznan.pl

dr hab. inż. Maciej Zakrzewicz prof. PP
maciej.zakrzewicz@put.poznan.pl

Wykładowcy

Wymagania wstępne

-

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy z zakresu projektowania systemów aplikacji biznesowych zgodnie z modelem architektury zorientowanej na usługi. 2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z wydajnością, niezawodnością i bezpieczeństwem aplikacji biznesowych. 3. Kształtowanie u studentów umiejętności pracy zespołowej.

Przedmiotowe efekty uczenia się

Wiedza:

ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu budowy i wdrażania aplikacji o architekturach zorientowanych na usługi (k2st_w2)

ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu

informatyki, takimi jak: usługi web services, automatyzacja procesów biznesowych, magistrale usługowe (k2st_w3)

ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów zorientowanych na usługi (k2st_w5)

ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w obszarze technologii implementacji i wdrażania usług web services (k2st_w4)

Umiejętności:

potrafi ocenić przydatność i możliwość wykorzystania usług web services (k2st_u6)

potrafi — przy formułowaniu i rozwiązywaniu zadań z zakresu budowy aplikacji o architekturze zorientowanej na usługi — integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne (k2st_u5)

potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie (k2st_u1)

potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie aplikacji zorientowanej na usługi (k2st_u9)

potrafi — zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne — zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt — co najmniej w części — używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia (k2st_u11)

Kompetencje społeczne:

rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, (k2st_k1)

rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych (k2st_k2)

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,
- ocenę i obronę przez studenta sprawozdania z realizacji projektu,
- ocenę wiedzy i umiejętności wykazanych na kolokwium zaliczeniowym pisemnym o charakterze testu wielokrotnego wyboru - 30 pytań, ocena pozytywna po udzieleniu poprawnych odpowiedzi na 15 pytań

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Wprowadzenie do architektury zorientowanej na usługi (SOA – Service-Oriented Architecture): motywacje, pojęcia podstawowe, przykłady zastosowań. Klasyfikacja usług w architekturze zorientowanej na usługi: usługi podstawowe, usługi pośredniczące, usługi procesowe, usługi aranżacyjne, usługi pomocnicze. Przegląd technologii usług implementacji usług podstawowych w technologii SOAP Web Services.. Przegląd podstawowych technologii XML: XML Schema, XPath, XSLT. Charakterystyka technologii podstawowych: protokół SOAP, deskryptory WSDL, rejestry UDDI. Standardy bezpieczeństwa dla usług Web Services: WS-Security, WS-SecurityPolicy, SAML. Funkcje bezpieczeństwa oferowane przez serwery aplikacji Java Enterprise Edition. Standardy transmisji danych binarnych dla usług Web Services: SOAP with Attachments, MTOM/XOP. Wiarygodna komunikacja z usługami Web Services: Web Service Reliable Messaging (WSRM). Konwersacyjne usługi Web Service. Metodyki budowy usług Web Services: Top-down, Bottom-up. Założenia architektury mikrousługowej. Konceptcje i implementacja usług REST. Wprowadzenie do usług procesowych i aranżacyjnych. Przegląd notacji dla opisu procesów biznesowych: BPEL. Implementacja usług procesowych: wywoływanie usług podstawowych, bezpieczeństwo, kompensacja transakcji, wykorzystywanie adapterów usługowych, eksternalizacja reguł decyzyjnych. Wprowadzenie do usług pośredniczących: koncepcja korporacyjnej magistrali usługowej (ESB – Enterprise Service Bus).

Modele komunikacji z usługami pośredniczącymi, bezpieczeństwo. Zadania korporacyjnej magistrali usługowej: dynamiczny routing wywołań, transformacja komunikatów, ochrona dostępu, walidacja komunikatów, weryfikacja wymagań SLA, monitoring wywołań. Strategie konstrukcji środowisk SOA, wzorce projektowe. Analiza systemowa zorientowana na usługi. SOA Governance.

Program laboratorium i zajęć projektowych obejmuje następujące zagadnienia: Modelowanie, projektowanie i implementacja usług WS SOAP oraz RESTful w środowisku Java Spring Boot wg metodyk Top-down i Bottom-up. Implementacja aplikacji klientów dla usług WS SOAP oraz RESTful. Koordynacja wywołań usług podstawowych WS SOAP oraz RESTful z wykorzystaniem środowiska Apache Camel. Implementacja graficznych interfejsów użytkownika dla usług procesowych. Budowa systemów mikroserwisowych z wykorzystaniem brokera komunikatów. Wdrażanie rozwiązań na platformy kontenerowe. Współpraca usług z bazami danych. Realizacja złożonych projektów zaliczeniowych.

Tematyka zajęć

- Usługi sieciowe SOAP: wprowadzenie, protokół SOAP, WSDL, UDDI, BPEL, MTOM/XOP, WS-Security, magistrala usługowa, architektura SOA
- Usługi sieciowe REST: wprowadzenie, architektura mikrousługowa
- Repetytorium: protokół HTTP
- Wprowadzenie do Node.js
- Projektowanie REST API: modelowanie zasobów i operacji, projektowanie URI, wykorzystanie komend HTTP, kodów statusu HTTP, mechanizmów negocjacji treści, wersjonowanie API, implementacja paginacji, filtrowania i sortowania, rate limiting, obsługa błędów
- Współpraca usług REST z bazami danych: wzorce architektoniczne, MongoDB, zarządzanie współbieżnością, współpraca z mechanizmami buforowania treści
- Komunikacja asynchroniczna i brokery komunikatów: komunikacja Publish/Subscribe i kolejkowa, wzorce projektowe Event Sourcing i CQRS, platformy Kafka i RabbitMQ
- Bezpieczeństwo usług REST: CORF, CSRF, uwierzytelnianie i autoryzacja klientów za pomocą JWT, OAuth, ochrona komunikacji sieciowej (HTTPS)
- Testowanie i dokumentowanie usług sieciowych REST: testy jednostkowe i integracyjne, automatyzacja dokumentacji za pomocą Swagger/OpenAPI, modelowanie danych - JSON Schema
- Konteneryzacja i wdrażanie: platformy Docker i Kubernetes

Metody dydaktyczne

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, demonstracja.
2. ćwiczenia laboratoryjne i zajęcia projektowe: ćwiczenia praktyczne, dyskusja, praca w zespole, warsztaty, studium przypadków, demonstracja.

Literatura

Podstawowa

1. REST API Design Rulebook, Mark Masse, O'Reilly, 2012
2. The Design of Web APIs, Arnaud Lauret, Manning, 2025
3. RESTful Java Patterns and Best Practices, Bhakti Mehta, Packt, 2014
4. Enterprise SOA: Service-Oriented Architecture Best Practices, Dirk Kraefzig, Karl Banke, Dirk Slama, Prentice Hall PTR, 2004
5. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Thomas Erl, Prentice Hall PTR, 2004
6. BPEL Cookbook: Best Practices for SOA-based Integration and Applications Development, editors: Harish Gaur, Markus Zirn, PACKT Publishing, 2006
7. Service-Oriented Architectures: Concepts, Technology, and Design, Thomas Erl, Prentice Hall PTR, 2005

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	65	2,50