

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Technologie internetowe w przetwarzaniu rozproszonym		Kod 1010512311010519879
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność Systemy rozproszone	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 15 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 3 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Cezary Sobaniec email: Cezary.Sobaniec@put.poznan.pl tel. 61 6652370 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę w zakresie: systemów operacyjnych, technologii sieciowych, przetwarzania rozproszonego, bezpieczeństwa systemów informatycznych oraz baz danych.
2	Umiejętności:	Powinien posiadać umiejętność pozyskiwania informacji z literatury oraz innych źródeł (w języku ojczystym i angielskim); powinien potrafić wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne; powinien potrafić integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne.
3	Kompetencje społeczne	Student powinien również rozumieć konieczność poszerzania swoich kompetencji. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
<ol style="list-style-type: none"> Przekazanie studentom podstawowej wiedzy z zakresu konstrukcji nowoczesnych i skalowalnych systemów rozproszonych w architekturze zorientowanej na usługi (SOA) z wykorzystaniem usług sieciowych Web Services i REST. Prezentacja problematyki konstrukcji aplikacji dla środowisk przetwarzania w chmurze. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z projektowaniem i implementacją usług sieciowych oraz efektywnym zarządzaniem projektem. Kształtowanie u studentów umiejętności pracy zespołowej. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, baz danych; - [K_W4] ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: usługi katalogowe, zarządzanie oprogramowaniem, archiwizacja i odtwarzanie, rozproszone systemy plików, systemy klastrowe - [K_W5] ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, - [K_W6] zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki, w szczególności obejmujące zarządzanie systemami uniksowymi, monitorowanie usług systemowych, diagnostykę sieci komputerowych - [K_W8] 		
Umiejętności:		

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K_U1]
2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K_U5]
3. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K_U7]
4. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne oraz eksperymentalne - [K_U9]
5. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K_U10]
6. potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12]
7. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]

Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

- a) w zakresie wykładów:
 - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów / ćwiczeń:
 - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
 - ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym o charakterze problemowym (5 pytań otwartych, każde dające 1 pkt., zaliczenie od 50%)
 - omówienie wyników zaliczenia,
 - b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez:
 - ocenę i obronę przez studenta sprawozdania z realizacji dwóch projektów,
 - ocenianie ciągle, na każdych zajęciach stopnia realizacji ćwiczeń laboratoryjnych,
- Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:
- przygotowanie i poprowadzenie na forum grupy warsztatów prezentujących uzgodnioną wcześniej usługę lub środowisko,
 - umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
 - uwagi związane z udoskonaleniem materiałów dydaktycznych,
 - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

1. Technologie HTML5: ewolucja języków znacznikowych, standard XML, aplikacje języka XML (MathML, DocBook, SVG, Open Document Format, Office Open XML), XML Information Set, prezentacja dokumentów XML, standard Extensible Stylesheet Language, XSL Formatting Objects, XHTML, geneza HTML5, obiekt Canvas, WebGL
2. Nowe technologie webowe: Application Cache, Web Worker, Web Storage, Web Sockets, HTTP 2.
3. Konstrukcja aplikacji webowych typu single-page application, model MVC, rozdział warstwy prezentacji i logiki biznesowej, biblioteka AngularJS.
4. Architektura SOA: definicja usługi, motywacje dla SOA, definicja architektury, założenia SOA, magistrala ESB, język BPEL
5. Web Services: motywacja, przegląd standardów WS-*, protokół SOAP, format komunikatów SOAP, wiązanie SOAP z protokołami transportowymi, standard opisu usług WSDL (definicje typów, komunikatów, usług, dowiązań do protokołów transportowych), profile WS-I (Basic, Security)
6. REST: usługi sieciowe WS a architektura Web, problem adresacji usług sieciowych, SOAP a inne technologie XML, definiowanie nowych protokołów aplikacyjnych: specyfikacje nakazowe i opisowe, protokół HTTP, styl architektoniczny REST, cele REST, definicja zasobu, reprezentacje zasobów, metody i kody błędów protokołu HTTP, przykład usługi REST, ograniczenia protokołu HTTP, testy zgodności z REST, REST a AJAX, bezpieczeństwo usług REST, WebDAV, realizacje usług REST
7. Architektura zorientowana na zasoby (ROA): modele usług sieciowych, znaczenie adresów URI, hipermedia w REST, modelowanie REST (nazwy zasobów, reprezentacje), problem wyboru reprezentacji zasobów, stan interakcji w usługach REST, granularność zasobów, zasoby specjalne, kolekcje zasobów, powiązania między zasobami, mikroformaty, serwery buforujące, walidatory aktualizacji, problem idempotentności operacji POST
8. Asynchroniczny model programowania usług sieciowych: generatory i współprogramy na przykładzie Python coroutines.

Zajęcia laboratoryjne prowadzone są w formie 3-godzinnych ćwiczeń, odbywających się w laboratorium. Ćwiczenia realizowane są indywidualnie lub w zespołach 2 osobowych w zależności od charakteru ćwiczeń. Program laboratorium obejmuje następujące zagadnienia:

1. Warstwa prezentacyjna: powtórka + uzupełnienia, HTML, CSS3, WebFonts
2. Technologie HTML5: detekcja wsparcia mechanizmów HTML5, mechanizm pamięci podręcznej Application Cache, WebWorker i SharedWorker ? przetwarzanie wielowątkowe w przeglądarce, Web Storage ? baza danych w przeglądarce, obiekt Canvas, inne interfejsy programistyczne
3. AJAX ? Asynchronous JavaScript and XML: przetwarzanie asynchroniczne w przeglądarce, obiekt XMLHttpRequest, format JSON, problem historii przetwarzania, dynamiczna manipulacja dokumentem poprzez DOM
4. Extensible Stylesheet Language: XSLT ? transformacje dokumentów XML, formatowanie prezentacji z wykorzystaniem Formatting Objects
5. Przetwarzanie dokumentów XML: interfejsy programistyczne: SAX, DOM, JDOM, TrAX, StAX, implementacja przetwarzania przykładowego dokumentu XML
6. Usługi sieciowe Web Services: protokół SOAP, opisy usług w języku WSDL, współoperacyjność usług
7. Usługi sieciowe REST: warsztaty z modelowania usług REST, hierarchia zasobów, reprezentacja zasobów, metody protokołu HTTP i ich semantyka, problem niezawodnego przetwarzania, powiązania między zasobami, środowiska programistyczne wspierające tworzenie usług REST
8. Programowanie usług sieciowych w modelu asynchronicznym, środowisko Tornado.
9. Prezentacje studentów
10. Realizacja 2 projektów: asynchroniczna komunikacja z serwerem z wykorzystaniem Web Socket, usługa sieciowa w modelu REST.
11. Konfiguracja i strojenie serwerów WWW, produkcyjne uruchamianie usług.

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, demonstracja oprogramowania
2. ćwiczenia laboratoryjne: opracowywanie niewielkich przykładowych programów testujących poznawane mechanizmy, eksperymenty funkcjonalne, dyskusja, pokaz multimedialny, studium przypadków, demonstracja oprogramowania

Literatura podstawowa:

1. Christopher Schmitt, Kyle Simpson. HTML5 Cookbook. O'Reilly Media, 2011.
2. Leonard Richardson, Sam Ruby. RESTful Web Services. O'Reilly Media 2007.

Literatura uzupełniająca:

1. Michael Rosen, Boris Lublinsky, Kevin T. Smith, Marc J. Balcer. Applied SOA Service-Oriented Architecture and Design Strategies. Wiley, 2012.
2. Jim Webber, Savas Parastatidis, Ian Robinson. REST in Practice Hypermedia and Systems Architecture. O'Reilly Media, 2010.

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
----------	--------------

1. udział w zajęciach laboratoryjnych:	30
2. przygotowanie do ćwiczeń laboratoryjnych:	5
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)	2 15
4. udział w wykładach	12
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	8
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 80 stron	4
7. przygotowanie do zaliczenia wykładu i obecność zaliczeniu: 3 godz. + 1 godz.	
Obciążenie pracą studenta	
forma aktywności	godzin ECTS
Łączny nakład pracy	76 3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	48 2
Zajęcia o charakterze praktycznym	42 2