KARTA OPISU MODUŁU KSZTAŁCENIA

Nazwa modułu/przedmiotu	Kod
Technologie internetowe w przetwarzaniu rozproszonym	**1010512311010519879**

Kierunek studiów	Profil kształcenia (ogólnoakademicki, praktyczny)	Rok / Semestr
Informatyka	**ogólnoakademicki**	**1 / 1**
Ścieżka obieralności/specjalność	Przedmiot oferowany w języku:	Kurs (obligatoryjny/obieralny)
Systemy rozproszone	**polski**	**obligatoryjny**

Stopień studiów:	**Forma studiów** (stacjonarna/niestacjonarna)
II stopień	**stacjonarna**

Godziny	Liczba punktów
Wykłady: **15** Ćwiczenia: **-** Laboratoria: **30** Projekty/seminaria: **-**	**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	**Podział ECTS (liczba i %)**
nauki techniczne	**3 100%**
nauki techniczne	**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

Odpowiedzialny za przedmiot / wykładowca:

mgr inż. Dariusz Dwornikowski
email: dariusz.dwornikowski@put.poznan.pl
tel. 61 6652371
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:

1. 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.

2. Prezentacja problematyki konstrukcji aplikacji dla środowisk przetwarzania w chmurze.

3. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z projektowaniem i implementacją usług sieciowych.

Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia

Wiedza:

1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z projektowaniem i konstrukcją usług sieciowych, z uwzględnieniem mechanizmów programowania i komunikacji asynchronicznej - [K2st_W2]

2. ma zaawansowaną i pogłębioną wiedzę z zakresu: współczesnych technologii internetowych, języków programowania dla usług Web, projektowania usług sieciowych - [K2st_W3]

3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w zakresie technologii internetowych i webowych - [K2st_W4]

4. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich - [K2st_W6]

Umiejętności:

1. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K2st_U2]

2. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K2st_U4]

3. 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 - [K2st_U5]

4. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]

5. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, w tym innych osób - [K2st_U16]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]

2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i proktycznych - [K2st_K2]

Sposoby sprawdzenia efektów kształcenia

Ocena podsumowująca:

- wykład: zaliczenie pisemne o charakterze problemowym (5 pytań otwartych, każde dające 1 pkt., zaliczenie od 50%),

- (opcjonalnie:) przygotowanie i poprowadzenie na forum grupy warsztatów prezentujących uzgodnioną wcześniej usługę lub środowisko,

- laboratorium: realizacja dwóch projektów programistycznych, ocenie podlega funkcjonalność, poprawność, zgodność z wymaganiami i rozumienie użytych technologii

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

1. Technologie HTML5: ewolucja języków znacznikowych, standard XML, aplikacje języka XML, XML Information Set, prezentacja dokumentów XML, standard XSLT i XSL Formatting Objects, XHTML, geneza HTML5.

2. Nowe technologie webowe: Web Worker, Web Storage, Service Worker, WebWebSocket, 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, profile WS-I.

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, 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: HTML5, CSS3, WebFonts.

2. Technologie HTML5: detekcja wsparcia mechanizmów HTML5, WebWorker i SharedWorker, Web Storage, obiekt Canvas, Service Worker, inne interfejsy programistyczne.

3. Powiadomienia WebPush.

4. Extensible Stylesheet Language: XSLT ? transformacje dokumentów XML, formatowanie prezentacji z wykorzystaniem Formatting Objects.

5. 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.

6. Programowanie usług sieciowych w modelu asynchronicznym, środowisko Tornado.

7. Realizacja 2 projektów: asynchroniczna komunikacja z serwerem z wykorzystaniem Web Socket, usługa sieciowa w modelu REST.

8. Konfiguracja i strojenie serwerów WWW, produkcyjne uruchamianie usług.

9. Prezentacje studentów

Literatura podstawowa:

Literatura uzupełniająca:

1. Leonard Richardson, Sam Ruby, RESTful Web Services, O'Reilly Media, 2008.

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
4. udział w wykładach	15
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	12
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 80 stron	8
7. przygotowanie do zaliczenia wykładu i obecność zaliczeniu: 3 godz. + 1 godz.	4

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