



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Technologie internetowe w przetwarzaniu rozproszonym [S2Inf1-SRC>TI]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/1

Studia w zakresie (specjalność)
Systemy rozproszone i chmurowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
15

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Cezary Sobaniec
cezary.sobaniec@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę w zakresie: systemów operacyjnych, technologii sieciowych, przetwarzania rozproszonego, bezpieczeństwa systemów informatycznych oraz baz danych.

Cel przedmiotu

1. Przekazanie wiedzy z zakresu konstrukcji nowoczesnych i skalowalnych systemów rozproszonych w architekturze zorientowanej na usługi (SOA) z wykorzystaniem usług sieciowych. 2. Nauka projektowania i konstrukcji usług sieciowych w modelu REST. 3. Prezentacja wybranych aktualnych technologii webowych przydatnych do budowy aplikacji rozproszonych. 4. Prezentacja asynchronicznego modelu konstrukcji serwerów usług sieciowych.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma wiedzę ogólną związaną projektowaniem systemów rozproszonych w architekturze zorientowanej na usługi (soa).
2. ma szczegółową wiedzę dotyczącą modelowania i implementacji usług sieciowych w modelu rest.

3. ma wiedzę dotyczącą aktualnie dostępnych technologii webowych możliwych do wykorzystania przy budowie systemów rozproszonych.
4. ma wiedzę dotyczącą asynchronicznego modelu realizacji usług sieciowych.

Umiejętności:

1. potrafi projektować i implementować usługi sieciowe w modelu rest.
2. potrafi właściwie dobrać i implementacyjnie wykorzystać aktualne technologie webowe do budowy rozproszonych systemów usługowych.
3. potrafi konstruować serwery usługowe w oparciu o model asynchronicznego przetwarzania.

Kompetencje społeczne:

1. potrafi pracować w grupie.
2. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Treści wykładowe zaliczane są poprzez pisemne kolokwium składające się z 4 pytań o charakterze problemowym. Za każde pytanie można uzyskać 12 pkt, ocena pozytywna wymaga uzyskania co najmniej 24 pkt.

Ćwiczenia laboratoryjne zaliczają 2 projekty realizowane w trakcie semestru.

Treści programowe

1. Technologie HTML5.
2. Nowe webowe interfejsy programistyczne.
3. Architektura zorientowana na usługi (SOA).
4. Usługi sieciowe Web Services.
5. Model usług sieciowych REST.
6. Architektura zorientowana na zasoby (ROA).
7. Asynchroniczny model programowania usług sieciowych.

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

1. Technologie HTML5: ewolucja języków znacznikowych, standard XML, XML Information Set, prezentacja dokumentów XML, standard XSL, XHTML, geneza HTML5.
2. Nowe technologie webowe: Web Worker, Web Storage, Service Worker, WebSocket, HTTP/2.
3. Architektura SOA: definicja usługi, motywacje dla SOA, definicja architektury, założenia SOA, magistrała ESB, język opisu usług.
4. Web Services: motywacja, przegląd standardów WS-*, protokół SOAP, format komunikatów SOAP, standard opisu usług WSDL, profile WS-I.
5. REST: usługi sieciowe WS a architektura Web, problem adresacji usług sieciowych, 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.
6. 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.
7. Asynchroniczny model programowania usług sieciowych: generatory i współprogramy na przykładzie Python coroutines.

Program laboratorium obejmuje następujące zagadnienia:

1. Technologie HTML5: detekcja wsparcia mechanizmów HTML5, WebWorker i SharedWorker, Web Storage, obiekt Canvas, Service Worker, inne interfejsy programistyczne, CSS3.
2. Extensible Stylesheet Language: transformacje dokumentów XML, formatowanie prezentacji z wykorzystaniem Formatting Objects.

3. Asynchroniczna komunikacja z wykorzystaniem WebSocket (protokół, klient, serwer).
4. Usługi sieciowe REST: modelowanie usług REST, hierarchia zasobów, reprezentacja zasobów, metody protokołu HTTP i ich semantyka, problem niezawodnego przetwarzania, powiązania między zasobami, środowiska programistyczne wspierające tworzenie usług REST.
5. Programowanie usług sieciowych w modelu asynchronicznym, środowisko Tornado.
6. Realizacja 2 projektów: asynchroniczna komunikacja z serwerem z wykorzystaniem Web Socket, usługa sieciowa w modelu REST.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, dyskusja.
2. Laboratorium: ćwiczenia praktyczne, polegające na tworzeniu i rozwijaniu prostych przykładów testujących różne technologie internetowe; konsultacje projektów zaliczeniowych.

Literatura

Podstawowa

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

Uzupełniająca

1. Dokumenty standaryzacyjne technologii internetowych. Portal World Wide Web Consortium, <https://www.w3.org>.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	30	1,00