



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy Internetowe

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Inteligentne Technologie Informatyczne

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratoria

30

Inne (np. online)

Ćwiczenia

Projekty/seminaria

Liczba punktów ECTS

5

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:
dr inż. Tomasz Pawlak

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu wytrarzania systemów komputerowych, programowania i architektury systemów komputerowych. W szczególności konieczna jest znajomość języka programowania Java.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy dotyczącej wykorzystania narzędzi programistycznych, w zakresie wytwarzania systemów internetowych.
2. Rozwijanie u studentów umiejętności rozwiązywania problemów dotyczących analizy i projektowania systemów internetowych w ramach technologii stosowanych w biznesie.
3. Omówienie zagadnień związanych z bezpieczeństwem i wydajnością systemów internetowych stosowanych w biznesie.
4. Kształtowanie u studentów umiejętności pracy zespołowej w ramach ćwiczeń laboratoryjnych w trakcie wykonywania projektów technicznych w grupach.

Przedmiotowe efekty uczenia się

Wiedza



1. Student ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów internetowych i technologii służących do projektowania i implementacji systemów internetowych - [K2st_W1]
2. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu systemów internetowych - [K2st_W2]
3. Student ma podbudowaną teoretycznie szczegółową wiedzę związaną z zagadnieniami z zakresu systemów internetowych, takimi jak: zaprojektowanie, wytworzenie i testowanie systemu internetowego - [K2st_W3]
4. Student ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki w zakresie systemów internetowych - [K2st_W4]
5. Student ma podstawową wiedzę o cyklu życia systemów internetowych - [K2st_W5]

Umiejętności

1. Student potrafi formułować i testować hipotezy związane z problemami inżynierskimi - [K2st_U4]
2. Student 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]
3. Student potrafi ocenić przydatność i możliwość wykorzystania nowych rozwiązań informatycznych w zakresie systemów internetowych - [K2st_U6]
4. Student potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie systemu internetowego, w tym dostrzec ograniczenia technologiczne - [K2st_U9]
5. Student potrafi wybrać język programowania odpowiedni do danego zadania programistycznego - [K2st_U11]

Kompetencje społeczne

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

Ocena formująca:

a) w zakresie wykładów:

- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach,

b) w zakresie laboratoriów:

- na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych,

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę wiedzy i umiejętności wykazanych na kolokwium pisemnym składającym się ze:

- zbioru pytań zamkniętych, z których na każde pytanie można udzielić jedną prawidłową odpowiedź z czterech możliwości. Za każdą poprawną odpowiedź uzyskuje się 1 punkt, a za błędną odejmowana jest



1/3 punktu.

- zbioru pytań otwartych, za które za każde można uzyskać od 2 do 4 punktów.

Aby uzyskać ocenę 3,0 należy uzyskać minimum 51% punktów, 3,5 - 61%, 4,0 - 71%, 4,5 - 81%, 5,0 - 91%.

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych obejmujących programowanie,
- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi związane z udoskonaleniem materiałów dydaktycznych.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

- Komunikacja w systemach internetowych. Wykład wprowadza do zagadnień związanych z protokołami HTTP, XML-RPC oraz usługami Web-services. Wykład wprowadza do protokołu XML-RPC służącego do wymiany danych przy pomocy ramek XML. Następna część wykładu poświęcona jest wprowadzeniu do technologii Web Services korzystającej z protokołu HTTP. Omówiony zostanie także wzorzec usług sieciowych REST.
- Technologie wspierane przez przeglądarkę: HTML, CSS. Omówiona zostanie funkcjonalność języka oraz mniej popularne funkcje wpływające na ergonomię korzystania z aplikacji internetowej. Przedstawione zostaną rozszerzenia dostępne w HTML 5 i CSS 3.
- Technologie wspierane przez przeglądarkę: JavaScript, DOM. Przedstawione zostaną zaawansowane metody wykorzystania języka np. budowanie wyrażeń lambda, pseudoobiektowość oraz narzędzia wspomagające programowanie w języku JavaScript. Przedstawiony zostanie model DOM dokumentów HTML i XML oraz związek z nim API.
- Mechanizm Ajax i JSON. Omówione zostanie działanie mechanizmu dostępu do danych pochodzących z innych witryn (Cross-origin resource sharing). Na wykładzie zostanie dokładnie omówiony mechanizm Ajax. Omówione zostaną nowe rozwiązania służące do komunikacji z serwerem wprowadzone w ramach projektu HTML 5. Następnie przedstawiony zostanie format JSON przesyłania danych oraz biblioteka jQuery ułatwiająca manipulację drzewem DOM oraz wykonywanie zapytań AJAX.
- Dynamiczne generowanie stron i zasobów WWW: Servlet'y i JSP. Wykład wprowadza do mechanizmów dynamicznego generowania stron WWW oraz struktury aplikacji internetowej w technologii JEE. Omówiona zostanie technologia Servlet znajdująca się w kontenerach WWW w technologii JEE. Następnie pokazane zostanie zastosowanie szablonów JSP do generowania stron HTML i innych udostępnianych zasobów.
- Trwałość danych w aplikacji internetowej na przykładzie JDBC i JPA. Wykład przedstawia sposoby przechowywania i pobierania danych przy pomocy Java Database Connectivity (JDBC). Omówiony zostanie interfejs JPA służący do utrwalania obiektów Java w bazie danych. Omówienie relacyjnych baz danych oraz baz typu NoSQL na przykładzie systemów MongoDB, Redis oraz Apache Cassandra.



- Wzorzec MVC. Przegląd środowisk zgodnych z wzorcem MVC i jego pochodnymi.
- Różne sposoby tworzenia aplikacji WWW na przykładach języka Dart i stosu MEAN: MongoDB + Express + AngularJS + Node.js. Dwa wykłady obejmują omówienie popularnych środowisk programistycznych pozwalających na tworzenie aplikacji zgodnych z modelem MVC. Środowiska prezentują różne paradygmaty stosowane w programowaniu aplikacji internetowych.
- Bezpieczeństwo, wydajność i poprawność aplikacji internetowych. Dwa wykłady wprowadzają w zagadnienia związane z tworzeniem bezpiecznych i wydajnych aplikacji internetowych. Omówione zostaną podstawowe zagrożenia, jakie można napotkać podczas eksploatacji aplikacji internetowych. Przedstawione zostaną metody pomiaru wydajności aplikacji internetowej oraz wpływające na nią czynniki.

Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych sesją instruktorską na początku semestru. Ćwiczenia realizowane są samodzielnie. Ćwiczenia realizowane są na podstawie materiałów dydaktycznych dostarczonych przez prowadzącego. Program laboratorium obejmuje następujące zagadnienia:

- Realizacja ćwiczeń wprowadzających do technologii HTTP - 8 godz.
- Technologia Representational State Transfer - 4 godz.
- Bazy danych NoSQL - 4 godz.
- Hypertext Markup Language - 2 godz.
- Cascading Style Sheets - 2 godz.
- JavaScript - 4 godz.
- Omówienie trudności w realizacji zagadnień i konsultacje 12 godz.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna.
2. Ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, praca w zespole, demonstracja.

Literatura

Podstawowa

1. Krzysztof Rychlicki-Kicior: Java EE 6 : programowanie aplikacji www, Helion, 2010
2. Steven Sanderson, Adam Freeman: ASP.NET MVC 3 framework : zaawansowane programowanie, Helion, 2012
3. Semmy Peruwal: Learning Web App Development, O'Reilly Media, 2014
4. Ethan Brown: Web Development with Node and Express, O'Reilly Media, 2014

Uzupełniająca

1. Dokumentacja standardów technicznych WWW: <http://www.w3.org>
2. Dokumentacja bibliotek Rails i Dart: <http://rubyonrails.org/>, <https://www.dartlang.org/>
3. Standardy związane z technologią J2EE: <http://www.jcp.org>



Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,5
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, realizacja zadań laboratoryjnych w ramach pracy domowej, przygotowanie do kolokwium) ¹	65	2,5

¹ niepotrzebne skreślić lub dopisać inne czynności