

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Wytwarzanie systemów internetowych		Kod 1010515321010510527
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność Technologie wytwarzania oprogramowania	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 3 100%
Odpowiedzialny za przedmiot / wykładowca:		
<p>dr inż. Tomasz Pawlak email: Tomasz.Pawlak@cs.put.poznan.pl tel. 61 6653022 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_W1-2, K_W4, K_W6-15, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
2	Umiejętności:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
3	Kompetencje społeczne	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl 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 dotyczącej wykorzystania narzędzi programistycznych, w zakresie wytwarzania systemów internetowych. Rozwijanie u studentów umiejętności rozwiązywania problemów dotyczących analizy i projektowania systemów internetowych w ramach technologii stosowanych w biznesie. Omówienie zagadnień związanych z bezpieczeństwem i wydajnością systemów internetowych stosowanych w biznesie. Kształtowanie u studentów umiejętności pracy zespołowej w ramach ćwiczeń laboratoryjnych w trakcie wykonywania projektów technicznych w grupach. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych i technologii służących do projektowania i implementacji systemów internetowych - [K_W4] ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: zaprojektowanie, wytworzenie i testowanie systemu internetowego. - [K_W5] ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce w zakresie - [K_W6] ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych - [K_W7] zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki - [K_W8] 		
Umiejętności:		

1. potrafi pozyskiwać informacje z literatury, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski - [K_U1]
2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K_U5]
3. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K_U10]
4. potrafi formułować i testować hipotezy związane z problemami inżynierskimi - [K_U12]
5. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]
6. potrafi sformułować wymagania pozafunkcjonalne dla wybranych charakterystyk jakościowych - [K_U23]
7. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K_U24]
8. potrafi wybrać język programowania odpowiedni do danego zadania programistycznego - [K_U26]
9. potrafi - zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne - zaprojektować system informatyczny - co najmniej w części - używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K_U27]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]
2. rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K_K4]
3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]
4. potrafi współdziałać i pracować w grupie, przyjmując w niej różne role - [K_K5]

Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia 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 egzaminie pisemnym w formie testu wielokrotnego wyboru,

Egzamin składa się ze:

- Zbioru od 8 do 12 pytań zamkniętych, z których na każde pytanie można udzielić jedną prawidłową odpowiedź z czterech możliwych. Za każdą poprawną odpowiedź uzyskuje się 1 punkt, a za błędną odejmowana jest 1/3 punktu.
- Zbioru od 3 do 5 pytań otwartych, za które można uzyskać od 2 do 3 punktów.

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

Czas odpowiedzi na pytania 1.5h.

- omówienie wyników egzaminu,
- 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,
- prezentacje autoreferatu powiązanego z tematem zajęć.

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. Przedstawienie preprocesorów CSS: Sass i Less.

- Technologie wspierane przez przeglądarkę: JavaScript, DOM, XSLT(XPath). 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 związane z nim API. Omówiony zostanie język transformacji XSLT w kontekście tworzenia fragmentów stron internetowych z plików XML.

- 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. Omówione zostaną technologia Servlet znajdująca się w kontenerach WWW w technologii JEE. Następnie pokazane zostanie zastosowanie szablonów JSP do generowania stron HTML i innych udostępnianych zasobów.

- Persistencja 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.

- Tworzenie aplikacji w Ruby on Rails i Sinatra - wykład wprowadza do języka Ruby i przedstawia środowiska Rails i Sinatra służące do tworzenia aplikacji internetowych.

- Tworzenie aplikacji internetowych w środowisku .NET - wykład przedstawia środowisko ASP.NET (MVC) w zakresie możliwości wytwarzania aplikacji internetowych.

- 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 czterech 4-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 1-godzinną sesją instruktażową na początku semestru. Ćwiczenia realizowane są samodzielnie lub przez 2-osobowe zespoły studentów. Ćwiczenia realizowane są na podstawie materiałów dydaktycznych dostarczonych przez prowadzącego. Program laboratorium obejmuje następujące zagadnienia:

Realizacja ćwiczeń wprowadzających do technologii HTTP ? 2 godz.

Rozwiązywanie zadań wprowadzających do technologii HTML5, CSS ? 4 godz.

Rozwiązywanie zadań wprowadzających do technologii DOM, XSLT, JavaScript ? 4 godz.

Realizacja ćwiczeń wprowadzających do technologii Servlet, JSP ? 4 godz.

Omówienie i ćwiczenia z technologiami JDBC, JPA i bazami NoSQL ? 2 godz.

Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy i na komputerze
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, wykonywanie eksperymentów, dyskusja, studium przypadków, mini-projekty programistyczne

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

Literatura 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>
4. Dodatkowa literatura uzupełniająca wskazana na stronie <http://etacar.put.poznan.pl/radoslaw.ziembinski>

Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych	16	
2. przygotowanie do ćwiczeń laboratoryjnych	8	
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych	8	
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)	4	
5. udział w wykładach	16	
6. napisanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	10	
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi	7	
8. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym	8	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	77	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	1
Zajęcia o charakterze praktycznym	42	2