

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Podstawy aplikacji internetowych		Kod 1010511361010510140
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 3 / 6
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 4
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 nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: Odpowiedzialny za przedmiot / wykładowca: hab. inż. Maciej Zakrzewicz, prof. nadzw. PP dr. inż. Maciej Piernik email: Maciej.Zakrzewicz@cs.put.poznan.pl email: maciej.piernik@cs.put.poznan.pl tel. 616652903 tel. (+48 61) 665-30-57 Instytut Informatyki Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu protokołów sieciowych, systemów baz danych i programowania zorientowanego obiektowo.
2	Umiejętności:	Powinien posiadać umiejętność programowania prostych aplikacji z wykorzystaniem zintegrowanych środowisk programistycznych.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu.
Cel przedmiotu:		
1. Uzyskanie wiedzy dotyczącej podstawowych pojęć i koncepcji z zakresu implementacji dokumentów WWW i aplikacji WWW, niezbędnej do rozróżniania podstawowych architektur aplikacji internetowych oraz metod implementacji ich modułów. 2. Uporządkowanie wiedzy z zakresu architektur sieciowych, protokołów sieciowych, bezpieczeństwa systemów rozproszonych. 3. Nabycie umiejętności implementacji aplikacji WWW z wykorzystaniem podstawowych technologii implementacji interfejsu użytkownika, m.in. HTML, CSS, JavaScript, XML+XSL, technologii implementacji logiki prezentacji, m.in. serwlety Java, JavaServer Pages, PHP, Active Server Pages, technologii implementacji logiki biznesowej, m.in. JavaBeans, biblioteki znaczników JSP. 4. Pozyskanie kompetencji społecznych dotyczących zespołowego przygotowania projektu, w tym organizacji pracy zespołowej, a zwłaszcza przywództwa oraz komunikatywności w procesie grupowego rozwiązywania problemu.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektur aplikacji WWW wykorzystujących bazy danych - [K1st_W4] 2. ma uporządkowaną wiedzę z zakresu protokołów sieciowych i bezpieczeństwa systemów rozproszonych - [K1st_W4] 3. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu projektowania, implementacji i wdrażania aplikacji WWW - [K1st_W7]		
Umiejętności:		

1. potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zrealizować aplikację internetową, używając właściwych metod, technik i narzędzi - [K1st_U10]
2. ma umiejętność formułowania algorytmów i ich programowania w językach stosowanych dla aplikacji internetowych z użyciem przynajmniej jednego z popularnych narzędzi - [K1st_U11]
3. ma umiejętność tworzenia prostych aplikacji internetowych, wykorzystujących systemy baz danych oraz interakcyjne interfejsy użytkownika - [K1st_U12]
4. potrafi zaprojektować dobry interfejs użytkownika dla różnych klas systemów informatycznych, w tym aplikacji internetowych - [K1st_U14]
5. potrafi wybrać technologię implementacji aplikacji WWW odpowiednią do zadanego obszaru zastosowań - [K1st_U18]

Kompetencje społeczne:

1. potrafi uczestniczyć w zespołowym przygotowaniu projektu, w tym zaplanować organizację pracy zespołowej oraz komunikować się w procesie grupowego rozwiązywania problemu - [K1st_K1]
2. ma świadomość znaczenia wiedzy inżynierskiej w rozwiązywaniu problemów inżynierskich, zna przykłady wadliwie działających systemów oraz jest w stanie wskazać przyczyny tych wad - [K1st_K2]
3. potrafi myśleć w sposób przedsiębiorczy, znajdując komercyjne zastosowanie dla tworzonego oprogramowania - [K1st_K3]

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 poprzednich wykładach;

b) w zakresie ćwiczeń:

- na podstawie oceny bieżącego postępu realizacji zadań.

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- w zakresie laboratorium - ocenę i obronę projektów wykonanych przez studenta. Za projekty przyznawane są punkty. Ocena końcowa jest obliczana na podstawie sumy zdobytych punktów). Przyjmuje się następującą skalę ocen i punktów: <0;50%>: ndst., (50%;60%>: dst, (60%;70%>: dst+, (70%;80%>: db, (80%;90%>: db+, (90%;100%>: bdb.

- w zakresie wykładu - ocenę wiedzy i umiejętności wykazanych na zaliczeniu pisemnym o formie testu jednokrotnego wyboru składającego się z 15 pytań, łączna liczba punktów za prawidłowe odpowiedzi: 15, minimalna liczba punktów umożliwiających zaliczenie: 8

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

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

1. Wprowadzenie do architektury WWW i problematyki aplikacji WWW: historia rozwoju technologii WWW, składniki podstawowej architektury WWW: klient HTTP, serwer HTTP, protokół http, pojęcia aplikacji WWW, serwera aplikacji oraz komponentowe modele aplikacji WWW, podstawowe własności języka HTML.
2. Interfejs użytkownika aplikacji WWW: arkusze stylów CSS i ich wykorzystanie do formatowania dokumentów HTML; język XML z uwzględnieniem ogólnej struktury dokumentu, opisu struktury za pomocą DTD i przestrzeni nazw, język XHTML, będący wersją HTML dostosowaną do reguł języka XML; język XSL służący do transformowania i formatowania dokumentów XML. Omówione zostaną zarówno część XSL dotycząca transformacji dokumentów (XSLT), jak i część dotycząca opisu sposobu formatowania (XSL-FO); znaczniki języka HTML służące do konstruowania formularzy; język skryptowy JavaScript, a w szczególności podstawy języka (deklaracje zmiennych, wyrażenia, operatory języka, struktury kontrolne i instrukcje, w tym pętle i instrukcje warunkowe) oraz model DOM (ang. Document Object Model) definiujący interakcję między skryptem JavaScript i obiektami zawartymi w dokumencie HTML; metody walidacji danych wprowadzanych przez użytkownika do formularzy HTML oraz metody manipulowania zawartością okna i dokumentu; omówienie metod pracy z łańcuchami znaków i datami oraz biblioteką oferującą bogaty zestaw funkcji matematycznych; prezentacja możliwości tworzenia interfejsu użytkownika za pomocą apletów języka Java.
3. Przegląd własności funkcjonalnych serwerów HTTP na przykładzie serwera Apache: wewnętrzna architektura serwera Apache, jego podstawowe parametry konfiguracyjne, strukturę i zastosowania dziennika serwera, mechanizmy odwzorowania logicznych ścieżek dostępu w ścieżki fizyczne, dyrektywy blokowe oraz funkcje ochrony dostępu do dokumentów znajdujących się po stronie serwera Apache.
4. Technologie implementacji logiki prezentacji: klasyfikacja metod konstrukcji logiki prezentacji; architektura i sposoby implementacji serwetów Java, obsługa nagłówków HTTP, zmiennych Cookies i sesji HttpSession; najważniejsze technologie szablonów (server pages) wykorzystywanych w aplikacjach WWW: ASP.NET oraz PHP; metody tworzenia logiki prezentacji aplikacji internetowej przy wykorzystaniu technologii szablonów rozwijanych dla języka Java; podstawowa technologia szablonów dla języka Java - technologia JSP (ang. Java Server Pages); cykl życia aplikacji JSP oraz podstawowe składowe technologie: deklaracje, dyrektywy i skrypty; język EL (ang. Expression Language), środowisko szkieletowe JavaServer Faces; przegląd rozwiązań szkieletowych typu Single Page Application, np. Angular, React.

5. Zagadnienia tworzenia logiki biznesowej przy wykorzystaniu komponentów JavaBean i bibliotek znaczników JSF: koncepcja komponentów JavaBean wraz z ich specyfikacją; zagadnienia związane z wykorzystaniem komponentów JavaBean na stronach JSF; wzorce projektowe wykorzystujące komponenty JavaBean do komunikacji między poszczególnymi warstwami aplikacji internetowej; technologia bibliotek znaczników JSF ? podstawowemu narzędziu służącemu do tworzenia dużych i złożonych aplikacji JSF; koncepcja i implementacja mechanizmu bibliotek znaczników na podstawie JSTL (ang. Java Standard Tag Library), standardowej biblioteki znaczników; przykłady ilustrujące sposób wykorzystania kilku rodzajów znaczników, m.in. znaczników podstawowych, formatujących, przetwarzających XML oraz obsługujących połączenie z bazą danych; metoda tworzenia własnych znaczników JSP.
6. Mechanizmy dostępu do baz danych w najpopularniejszych technologiach do tworzenia aplikacji WWW: Java EE, ASP.NET i PHP. W kontekście technologii Java EE przedstawione będą podstawy JDBC, charakterystyczny dla aplikacji Java EE mechanizm uzyskiwania połączeń z bazą danych poprzez źródła danych oraz wprowadzenie do technologii odwzorowania obiektowo-relacyjnego (O/RM) i standardu Java Persistence.
7. Funkcjonalność wymagana w aplikacjach internetowych, powtarzająca się niezależnie od konkretnych zastosowań i określana mianem infrastruktury. Po ogólnym wprowadzeniu przedstawiona będzie architektura Web Forms dla ASP.NET oraz architektura Model-View-Controller i jej popularna implementacja dla platformy Java EE ? Struts. Następnie omówiona zostanie technologia JavaServer Faces (JSF), ułatwiająca tworzenie aplikacji Java EE poprzez dostarczenie infrastruktury dla stanowego, komponentowego interfejsu użytkownika.
8. Przegląd własności funkcjonalnych serwerów HTTP na przykładzie serwera Apache. W ramach wykładu omówimy wewnętrzną architekturę serwera Apache, jego podstawowe parametry konfiguracyjne, strukturę i zastosowania dziennika serwera, mechanizmy odwzorowania logicznych ścieżek dostępu w ścieżki fizyczne, dyrektywy blokowe oraz funkcje ochrony dostępu do dokumentów znajdujących się po stronie serwera Apache.
9. Najważniejsze metody atakowania aplikacji WWW oraz mechanizmy ochrony przed nimi. Poruszone zostaną następujące rodzaje zagrożeń: kradzież kodu źródłowego aplikacji JSP, atak na pola ukryte HTML, atak na zmienne Cookies, atak typu Path Traversal, atak typu SQL Injection, przejęcie sesji, atak typu Cross-Site Scripting i atak typu Denial of Service.

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

Metody dydaktyczne:

- wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, demonstracja.
- ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, warsztaty.

Literatura podstawowa:

- Praca zbiorowa, CSS i Ajax. Strony www zgodne ze standardami sieciowymi W3C, Helion, 2008.
- A. Marciniak, JavaServer Faces i Eclipse Galileo. Tworzenie aplikacji www, Helion, 2010.
- L. Bruce, S. Remy, Wprowadzenie do HTML 5. Autorytety informatyki, Helion, 2011.
- L. Jesse, H. Dan, M. Brian, ASP.NET 2.0 i Ajax. Wprowadzenie, O'Reilly, 2008.

Literatura uzupełniająca:

- P. Kazienko, K. Gwiazda, XML na poważnie, Helion, 2002.
- L. Jesse, H. Dan, M. Brian, ASP.NET 2.0 i Ajax. Wprowadzenie, O'Reilly, 2008.
- E. Jendrock, I. Evans, D. Gollapudi, K. Haase, C. Srivathsa, The Java EE 6 Tutorial, Oracle, 2010.

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych / ćwiczeniach:	30
2. przygotowanie do ćwiczeń laboratoryjnych:	10
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2 8
4. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	8
5. przygotowanie do sprawdzianów / kolokwium	30
6. udział w wykładach	15
7. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym	

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	103	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	48	2