

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Technologie programistyczne - aplikacje lokalne		Kod 1010515311010510648
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność Technologie wytwarzania oprogramowania	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -	Liczba punktów 4	
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 nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Piotr Zielniewicz email: Piotr.Zielniewicz@cs.put.poznan.pl tel. (48)(61) 665-2935 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty: K1st_W1, K1st_W5, K1st_W6 oraz K1st_W7, 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 K1st_U4, K1st_U9, K1st_U10, K1st_U11 oraz K1st_U14, 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 K1st_K1, K1st_K2 oraz K1st_K3, 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: 1. Przekazanie studentom podstawowej wiedzy z technologii programistycznych, w zakresie budowy, projektowania, testowania i oceny graficznych interfejsów użytkownika, technologii budowy graficznego interfejsu użytkownika, przetwarzania wielowątkowego na CPU i GPU, wykorzystania obiektowego modelu komponentów COM oraz technologii OLE Automation do rozbudowy aplikacji rodziny MS Office, wizualizacji w oparciu o technologię DirectX, składowania/serializacji danych aplikacji z wykorzystaniem technologii ODBC, JDBC, ADO.NET, DAO i Hibernate, tworzenia efektywnych programów na platformie .NET, korzystania z kodu natywnego w aplikacjach Javy i na platformie .NET (JNI, P/Invoke), aspektów bezpieczeństwa w wytwarzaniu oprogramowania 2. Rozwijanie u studentów umiejętności rozwiązywania problemów dotyczących projektowania interfejsów użytkownika, doboru technologii programistycznych do realizowanego zadania, zwiększania jakości i efektywności tworzonego oprogramowania		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie języków i paradygmatów programowania, grafiki i komunikacji człowiek-komputer, oraz inżynierii oprogramowania - [K2st_W2]</p> <p>2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: tworzenie graficznych interfejsów użytkownika, przetwarzanie wielowątkowe, interfejsy i mechanizmy do komunikacji aplikacji ze źródłem danych - [K2st_W3]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, w szczególności związanych z projektowaniem interfejsu człowiek-komputer (kognitywistyka, ergonomia, biocybernetyka) - [K2st_W4]</p> <p>4. ma podstawową wiedzę o cyklu życia systemów informatycznych, zna aspekty bezpieczeństwa związane z projektowaniem, tworzeniem i pielęgnacją dystrybucją oprogramowania - [K2st_W5]</p> <p>5. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich, w tym tworzenia aplikacji wielowątkowych, projektowania graficznych interfejsów użytkownika - [K2st_W6]</p>
<p>Umiejętności:</p>
<p>1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K2st_U1]</p> <p>2. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody symulacyjne oraz eksperymentalne - [K2st_U4]</p> <p>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]</p> <p>4. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]</p> <p>5. potrafi współdziałać w zespole podczas realizacji projektów na zajęciach - [K2st_U15]</p> <p>6. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K2st_U16]</p>
<p>Kompetencje społeczne:</p>
<p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]</p> <p>2. rozumie znaczenie wykorzystania najnowszej wiedzy z zakresu technologii programistycznych - [K2st_K2]</p>

<p style="text-align: center;">Sposoby sprawdzenia efektów kształcenia</p>
<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:</p> <p>Ocena formująca:</p> <p>a) w zakresie wykładów:</p> <ul style="list-style-type: none">- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach <p>b) w zakresie laboratoriów:</p> <ul style="list-style-type: none">- na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych <p>Ocena podsumowująca:</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- ocenę wiedzy i umiejętności wykazanych na kolokwium pisemnym o charakterze problemowym <p>b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,- ocenianie ciągle, na każdym zajęciu (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznаныmi zasadami i metodami,- ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez 2 projekty programistyczne w semestrze, <p>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:</p> <ul style="list-style-type: none">- 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
<p style="text-align: center;">Treści programowe</p>

Program wykładu obejmuje następujące zagadnienia:

Wprowadzenie do graficznych interfejsów użytkownika (zadania, koncepcja budowy, historia i trendy rozwoju interfejsów użytkownika). Projektowanie interfejsów użytkownika (ergonomia, czynniki psychologiczne, konsekwencje błędów projektowo-implémentacyjnych, model programu a model użytkownika, podstawowe cele projektowe: intuicyjność, prostota, kompletność, elegancja). Prototypowanie i testowanie interfejsu użytkownika (ocena spełnialności wymagań, ocena użyteczności interfejsu). Warunki ekstremalne w projektowaniu interfejsu użytkownika. Przegląd zasad i dobrych praktyk dotyczących projektowania interfejsu użytkownika. Technologie budowy interfejsu użytkownika w Javie (AWT, Swing, SWT) oraz na platformie .NET: (WinForms, WPF): założenia, przegląd technologii, modele obiektowe bibliotek graficznych, delegacyjny model obsługi zdarzeń, architektura MVC, przegląd i zadania menadżerów rozkładu, przegląd wybranych komponentów, dostosowywanie komponentów do własnych potrzeb, mechanizm look&feel, wielowątkowość a obsługa interfejsu użytkownika, tworzenie aplikacji wielojęzycznych. Omówienie obiektowego modelu komponentów COM (architektura, definiowanie interfejsów, zarządzanie czasem życia, techniki tworzenia komponentów). Podstawy technologii OLE Automation (założenia, typy danych, obiektowy model MS Office) i DirectX (wprowadzenie, porównanie z OpenGL, modele obiektów, transformacje 3D, renderowanie grafiki, przegląd interfejsów Managed DirectX). Podstawy platformy .NET (środowisko uruchomieniowe CLR, wspólna biblioteka typów CTS, mechanizmy zarządzania zasobami, usługi wykonawcze, przetwarzanie wielowątkowe, sposoby zwiększania efektywności działania kodu). Technologie składowania i dostępu do baz danych (ODBC, JDBC, ADO.NET, DAO, Hibernate). Wprowadzenie do programowania na GPU (CUDA, OpenCL, DirectCompute). Aspekty bezpieczeństwa w wytwarzaniu oprogramowania (przyczyny powstawania błędów zabezpieczeń, przegląd i typy zagrożeń, analizowanie i modelowanie zagrożeń, przegląd błędów implementacyjnych, sposoby przeciwdziałania).

Zajęcia laboratoryjne prowadzone są w formie ośmiu 2-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych sesją instruktazową na początku semestru. Ćwiczenia realizowane są przez studentów indywidualnie. Program laboratorium obejmuje następujące zagadnienia:

Projektowanie interfejsów użytkownika: podstawowe założenia i cele projektowe, wybrane problemy i sposoby ich rozwiązywania. Przegląd zasad i dobrych praktyk dotyczących projektowania interfejsu użytkownika. Prototypowanie i ocena użyteczności interfejsu użytkownika. Technologia Swing: przegląd i zadania menadżerów rozkładu, dostosowywanie wybranych komponentów do własnych potrzeb, obsługa mechanizmu look&feel, wielowątkowa realizacja zadań długotrwałych, dynamiczna zmiana języka interfejsu użytkownika. Platforma .NET: tworzenie aplikacji wielowątkowych, zarządzanie procesami, wielowątkowa realizacja zadań długotrwałych (WinForms, WPF), dynamiczna zmiana języka interfejsu użytkownika (WinForms, WPF). Technologia DirectX: wprowadzenie, budowa przykładowej aplikacji. Mechanizmy dostępu do baz danych: obsługa składowania danych/serializacji obiektów w technologiach ODBC, JDBC, ADO.NET, Hibernate. Programowanie na GPU: przegląd technologii, budowa przykładowej aplikacji w technologii CUDA.

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. B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition), Prentice Hall, 2009
2. B. Eckel, Thinking in Java, Edycja polska, Wydanie IV, Helion, 2006
3. K. Richlicki-Kicior, Tworzenie aplikacji graficznych w .NET 3.0, Helion, 2007
4. C. Campbell, R. Johnson, A. Miller, S. Toub, Parallel Programming with Microsoft .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures, Microsoft Patterns & Practices, 2010

Literatura uzupełniająca:

1. W. O. Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Wiley Computer Publishing, 2007
2. J. Elliott, R. Eckstein, M. Loy, D. Wood, B. Cole, Java Swing, 2nd Edition, O'Reilly, 2002

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych	16
2. przygotowanie do ćwiczeń laboratoryjnych	16
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)	4
4. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	16
1	20
5. udział w wykładach	16
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 200 stron	
7. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym	

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	104	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	1
Zajęcia o charakterze praktycznym	48	2