

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie wizualne		Kod 1010514371010510452
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 4 / 7
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: I 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) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 4 100%
Odpowiedzialny za przedmiot / wykładowca:		
Dr inż. Jan Kniat email: Jan.Kniat@cs.put.poznan.pl tel. 61 6652981 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		Dr inż. Paweł Wojciechowski email: Pawel.Wojciechowski@cs.put.poznan.pl tel. 61 6653031 Instytut Informatyki 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 metodologii programowania, znać metodologię programowania obiektowego i omawiane uprzednio języki programowania obiektowego, znać podstawowe wzorce projektowe oraz architekturę aplikacji modułowych, których składniki są wykorzystywane. Powinien również znać podstawowe pojęcia z zakresu algorytmiki, złożoności obliczeniowej, programowania systemów współbieżnych i współpracy z bazami danych.
2	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych, korzystania z zaawansowanych systemów programowania, budowania kodu wysokiej jakości oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i mieć gotowość do podjęcia współpracy w ramach zespołu. 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 dotyczącej zaawansowanego środowiska programowania wizualnego .NET, w zakresie: szczególowej znajomości obiektowego języka programowania C#, zestawu klas i funkcji udostępnianych przez biblioteki Windows Forms i Windows Presentation Foundation, technologii ADO.NET, NET.Remoting/Networking. 2. Rozwijanie u studentów umiejętności rozwiązywania podstawowych problemów algorytmicznych, umiejętności projektowania oraz implementowania aplikacji okienkowych. Celem przedmiotu jest również pogłębienie umiejętności stosowania wzorców projektowych, tworzenia bibliotek dynamicznych DLL, testowania tworzonych aplikacji, tworzenia aplikacji współbieżnych i rozproszonych oraz aplikacji współpracujących z bazami danych. 3. Kształtowanie u studentów umiejętności pracy zespołowej poprzez realizowanie przez kilkusobową grupę studentów nieco większych projektów programistycznych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności obliczeniowej, architektury systemów informatycznych, technologii sieciowych, języków i paradygmatów programowania, grafiki i komunikacji człowiek-komputer, baz danych, inżynierii oprogramowania. - [K_W4]</p> <p>2. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów, budowy systemów komputerowych, technologii sieciowych, wykorzystywania języków programowania, grafiki i komunikacji człowiek-komputer, baz danych, inżynierii oprogramowania. - [K_W8]</p> <p>3. zna architekturę środowiska programowego .NET i posiada wiedzę o podstawowych technologiach udostępnianych przez to środowisko i sposobach tworzenia aplikacji konsolowych i okienkowych oraz przygotowywania bibliotek dynamicznych DLL - [K_W8]</p> <p>4. zna szczegółowo język programowania obiektowego C# i jego zaawansowane konstrukcje (m. in. programowanie współbieżne, synchronizacja dostępu do obiektów współużytkowanych, wyrażenia lambda, funkcje rozszerzające, wyrażenia LINQ) - [K_W5,K_W8]</p> <p>5. zna podstawowe wzorce projektowe stosowane przy tworzeniu aplikacji ze szczególnym uwzględnieniem wzorca Model-View-ViewModel oraz metodologię testowania tworzonych oprogramowania (testy jednostkowe, testy akceptacyjne) - [K_W8]</p> <p>6. zna biblioteki klas i funkcji środowiska programowego .NET umożliwiające tworzenie aplikacji okienkowych Windows Forms, Windows Presentation Foundation - [K_W8]</p> <p>7. zna technologie środowiska programowego .NET umożliwiające tworzenie aplikacji współpracujących z bazami danych ADO.NET - [K_W8]</p> <p>8. zna technologie środowiska programowego .NET umożliwiające tworzenie aplikacji współbieżnych i rozproszonych NET.Remoting/Networking - [K_W8]</p>
<p>Umiejętności:</p> <p>1. 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 - [K_U1]</p> <p>2. ocenić złożoność obliczeniową algorytmów i problemów - [K_U13]</p> <p>3. systematycznie przeprowadzać testy jednostkowe i funkcjonalne - [K_U17]</p> <p>4. wybrać język programowania odpowiedni do danego zadania programistycznego - [K_U20]</p> <p>5. potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K_U21]</p> <p>6. formułować algorytmy i zapisywać je z użyciem języka C# i środowiska programowego .NET - [K_U21]</p> <p>7. zaprojektować, zrealizować i przetestować aplikację okienkową wykorzystującą biblioteki Windows Forms lub Windows Presentation Foundation - [K_U21]</p> <p>8. przygotować bibliotekę dynamiczną DLL będącą modułem przeznaczonym do wielokrotnego wykorzystania 9. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi, - [K_U22]</p>
<p>Kompetencje społeczne:</p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]</p> <p>2. potrafi inspirować i organizować proces uczenia się innych osób - [K_K2]</p> <p>3. potrafi współdziałać i pracować w grupie, przyjmując w niej różne role - [K_K5]</p>

Sposoby sprawdzenia efektów kształcenia
<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 poprzednich wykładach; <p>b) w zakresie ćwiczeń:</p> <ul style="list-style-type: none">- na podstawie oceny bieżącego postępu realizacji zadań, <p>Ocena podsumowująca:</p> <p>Sprawdzanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- ocenę wiedzy i umiejętności związanych z realizacją kolejnych zadań projektowych / laboratoryjnych,- ocenę i obronę przez studenta/studentów sprawozdania z realizacji projektu,- przeglądowy test sprawdzający wiedzę przekazywana podczas wykładu. <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,- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,- uwagi związane z udoskonaleniem materiałów dydaktycznych,- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Wykład.

Definicja programowania wizualnego. Środowisko programowe .NET. Wspólne środowisko uruchomieniowe: kompilator JITC, biblioteki klas. Podstawowe cechy .NET. Wzorce projektowe, a szczególnie wzorzec Model-View-ViewModel.

Język C#. Struktura programu. Rodzaje danych, wbudowane typy zmiennych, deklarowanie i definiowanie zmiennych, wiązania dynamiczne dla zmiennych. Typy implikowane, rozszerzone, rzutowanie wartości zmiennych, opakowanie i rozpakowanie danych. Funkcje i właściwości dostępne dla typów danych. Łańcuchy znaków i metody ich przekształcania. Wprowadzanie i wyprowadzanie danych za pośrednictwem konsoli. Konwersja znakowo - binarna i binarno - znakowa. Ustawienia narodowe. Wyrażenia arytmetyczne i logiczne. Instrukcje warunkowe i tworzenia pętli. Struktury danych. Tablice jedno i wielowymiarowe, tablice nierównomierne, funkcje i metody dostępne dla tablic. Struktury i ich egzemplarze. Modularyzacja programu - definiowanie i wywoływanie funkcji. Definicja funkcji, wywołanie funkcji, sposoby przekazywania argumentów funkcjom, wartości domyślne argumentów i argumenty nazwane, wzorce funkcji. Klasy. Definiowanie klas, atrybuty klas, definiowanie danych składowych (atrybutów) i funkcji składowych (metod). Hermetyzacja danych zawartych w klasie. Składowe statyczne klas. Klasy wieloplikowe, klasy statyczne.

Tworzenie obiektów klas, inicjowanie wartości składowych klasy. Funkcje rozszerzające dla klas. Klasy anonimowe. Wiązania dynamiczne dla klas. Polimorfizm w zakresie pojedynczej klasy. Indeksatory. Definiowanie konwersji obiektów klas. Modelowanie relacji całość - część za pomocą kolekcji klas, klasy zagnieżdżone. Dziedziczenie. Polimorfizm w hierarchii klas dziedziczących.

Klasy abstrakcyjne i abstrakcyjne funkcje wirtualne. Finalizacja i zbieranie śmieci. Pliki dyskowe i serializacja. Klasy kolekcji i udostępniane przez nie funkcje. Parametryczne klasy kolekcji (generics). Interfejsy. Interfejsy standardowe. Iteratory. Interfejsy parametryczne. Delegacje. Tablice delegacji, delegacje wielokrotne, delegacje anonimowe, wzorce delegacji. Wyrażenia lambda. Zdarzenia. Obsługa wyjątków. Wyrażenia zapytaniowe LINQ dla kolekcji danych. Współbieżność. Asynchroniczne wykonywanie delegacji. Funkcje zwrotne (callback), wykonywanie cykliczne. Synchronizacja dostępu do obiektów współużytkowanych. Synchronizacja ścieżek za pomocą zdarzeń. Pula ścieżek CLR. Współbieżne wykonywanie zadań, współbieżne wykonywanie obliczeń. Współbieżne wyrażenia zapytaniowe PLINQ. Przestrzenie nazw, pakiety i refleksja typów. Tworzenie bibliotek dynamicznych DLL.

Biblioteka Windows Forms. Wykorzystywane przestrzenie nazw, formularz początkowy i jego właściwości. Dodawanie elementów sterujących, określanie ich właściwości i definiowanie funkcji obsługi zdarzeń. Obsługa myszki i klawiatury, menu, pasek statusu, pasek narzędzi. Podstawowe elementy sterujące. Tworzenie wykresów. Realizacja funkcji przeciągnij i upuść. Obsługa schowka, formularze wielojęzyczne. Okna dialogowe. Zakładki.

Środowisko graficzne GDI+. Pozyskiwanie obiektu graphics. Narzędzia graficzne. Funkcje rysujące i wypełniające. Przetwarzanie obrazów. Przetwarzanie pikseli, przetwarzanie obrazów GDI+.

Biblioteka Windows Presentation Foundation. Architektura aplikacji - rola języków XAML, C#. Panele i ich rodzaje. Menu. paski narzędzi, pasek statusu. Podstawowe elementy sterujące. Wiązanie elementów sterujących ze zmiennymi i strukturami danych. Definiowanie stylów. Współpraca z systemem operacyjnym - obsługa poleceń standardowych. Generowanie obiektów graficznych. Transformacje, grupowanie i łączenie geometrii. Efekty wizualne. Animacje.

Technologia ADO.NET. Tworzenie i przetwarzanie lokalnych relacyjnych baz danych. Klasy wiersza, kolumny, tabeli, bazy danych. Dołączanie, usuwanie, filtrowanie, sortowanie danych. Definiowanie związków pomiędzy tabelami. Przeszukiwanie lokalnej bazy danych: język TSQL. Wyrażenia zapytaniowe LINQ to DataSet. Współpraca z istniejącymi bazami danych. Wyrażenia zapytaniowe LINQ to SQL.

Przetwarzanie plików XML. Wczytywanie i zapis plików XML. Przeszukiwanie pliku XML. Analiza poprawności plików XML. Wykorzystanie plików XML we współpracy z serwisami internetowymi.

Technologia NET.Remoting. Domena aplikacji, tworzenie nowych domen i uruchamianie programów. Elementy współpracujące: klient, kanał, serwer, obiekt zdalny. Rodzaje kanałów i tryby współpracy. Ustalanie reguł współpracy za pomocą plików konfiguracyjnych lub za pomocą funkcji odpowiednich klas. Pobieranie pliku konfiguracyjnego ze zdalnego serwera. Asynchroniczne wywoływanie obiektu zdalnego, ustalanie czasu życia obiektu zdalnego.

Technologia NET.Networking. Definiowanie i wykorzystanie gniazd UDP i TCP. Klasy transportowe UDPClient, TCPClient, TCPListener. Klasy protokółów. Programowanie asynchroniczne.

Laboratorium.

W początkowej części zajęć laboratoryjnych studenci implementują proste aplikacje stanowiące przykład omawianych na wykładach mechanizmów. Programy te są realizowane przez wszystkich studentów i podlegają ocenie prowadzącego. Kolejne aplikacje konsolowe zapisywane w języku C# to: programy wykorzystujących podstawowe struktury danych i klasy kolekcji, programy korzystające z interfejsów i delegacji, programy przetwarzające pliki dyskowe, programy współbieżne. Następnie studenci przygotowują aplikacje okienkowe stosując bibliotekę Windows Forms oraz technologie ADO.NET i NET.Remoting/Networking. Kolejnym etapem jest realizacja aplikacji okienkowych korzystających z biblioteki WPF z uwzględnieniem zaawansowanych mechanizmów oferowanych tę bibliotekę (m. in. mechanizmów wiązania (binding) i powiadamiania o zmianach właściwości (property change notification)).

W drugiej części zajęć laboratoryjnych studenci indywidualnie lub w niewielkich zespołach przygotowują bardziej rozbudowaną, warstwową aplikację z interfejsem użytkownika, realizowaną zgodnie z modelem MVVM.

Literatura podstawowa:		
1. Griffiths Ian, Adams Matthew, Liberty Jesse, C#. Programowanie, Helion, 2012		
2. Andrew Troelsen, Język C# 2010 i platforma .NET 4.0, Wydawnictwo Naukowe PWN, 2011		
3. Jacek Matulewski, Dawid Borycki, Grzegorz Krause, Maciej Grabek, Maciej Pakulski, Mateusz Warczak, Jacek Lewandowski, Sławomir Orłowski, Visual Studio 2010 dla programistów C#, Helion, 2011		
4. Raffaele Garofalo, Budowanie aplikacji biznesowych za pomocą Windows Presentation Foundation i wzorca Model View ViewM, PROMISE, 2011		
5. Materiały dydaktyczne dostępne na stronie www.cs.put.poznan.pl/jkniat		
6. Dokumentacja elektroniczna systemu programowania wizualnego Visual Studio.NET		
Literatura uzupełniająca:		
1. Hejlsberg Anders, Torgersen Mads, Wiltamuth Scott, Golde Peter, Język C#. Programowanie, Helion, 2010		
2. Mike Snell, Lars Powers, Microsoft Visual Studio 2010. Księga eksperta, Helion 2011		
3. Matthew MacDonald, Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4.0, Apress, 2010		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych:	16	
2. przygotowanie do ćwiczeń laboratoryjnych:	10	
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych:	10	
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2	
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	10	
6. przygotowanie do sprawdzianów / kolokwium	16	
7. udział w wykładach	20	
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 200 stron	10	
9. 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	34	1
Zajęcia o charakterze praktycznym	46	2