

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie wizualne		Kod 1010511361010510452
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: 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: szczegółowej znajomości obiektowego języka programowania C#, zestawu klas i funkcji udostępnianych przez biblioteki Windows Presentation Foundation, technologii ADO.NET, Windows Communication Foundation (WCF). 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 nieco większych projektów programistycznych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów informatycznych - [K1st_W4]</p> <p>2. 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 - [K1st_W4]</p> <p>3. 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) - [K1st_W4]</p> <p>4. zna podstawowe wzorce projektowe stosowane przy tworzeniu aplikacji ze szczególnym uwzględnieniem wzorca Model-View-ViewModel oraz metodologię testowania tworzonych oprogramowania za pomocą testów jednostkowych - [K1st_W4]</p> <p>5. zna biblioteki klas i funkcji środowiska programowego .NET umożliwiające tworzenie aplikacji okienkowych Windows Presentation Foundation - [K1st_W4]</p> <p>6. zna technologie środowiska programowego .NET umożliwiające tworzenie aplikacji współpracujących z bazami danych ADO.NET - [K1st_W4]</p> <p>7. ma podstawową wiedzę o cyklu życia systemów informatycznych w ramach realizacji projektu od założeń systemu, projektu i implementacji architektury, przygotowania testów jednostkowych, realizacji dostępu do różnych źródeł danych i interfejsu użytkownika - [K1st_W6]</p> <p>8. zna techniki programowania obiektowego, wykorzystania wybranych wzorców projektowych, budowy elastycznej architektury aplikacji i interfejsu użytkownika. Zna środowisko Visual Studio - [K1st_W7]</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 - [K1st_U1]</p> <p>2. potrafi odpowiednio posługiwać się technikami informacyjno-komunikacyjnymi, znajdującymi zastosowanie na różnych etapach realizacji przedsięwzięć informatycznych - [K1st_U2]</p> <p>3. potrafi, formułując i rozwiązując zadania informatyczne, zastosować odpowiednio dobrane metody dostępu do danych, zaprojektować architekturę aplikacji - [K1st_U4]</p> <p>4. potrafi dokonać analizy sposobu funkcjonowania systemu informatycznego z interfejsem użytkownika, ocenić architekturę aplikacji w zakresie łatwości dokonywania w niej zmian. Potrafi ocenić architekturę oprogramowania z punktu możliwości zmiany źródła danych (aplikacja wielowarstwowa), niezależności warstwy prezentacji aplikacji od jej logiki funkcjonalnej (model MVVM) - [K1st_U9]</p> <p>5. wybrać język programowania odpowiedni do danego zadania programistycznego - [K1st_U10]</p> <p>6. potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K1st_U10]</p> <p>7. potrafi zaprojektować, zrealizować i przetestować aplikację okienkową wykorzystującą biblioteki Windows Presentation Foundation - [K1st_U14]</p> <p>8. potrafi przygotować bibliotekę dynamiczną DLL będącą modułem przeznaczonym do wielokrotnego wykorzystania - [K1st_U14]</p> <p>9. potrafi planować i realizować proces własnego permanentnego uczenia się oraz zna możliwości dalszego rozwijania wiedzy w zakresie zaprezentowanych technologii - [K1st_U19]</p>
<p>Kompetencje społeczne:</p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K1st_K1]</p> <p>2. ma świadomość znaczenia wiedzy i zaawansowanych technik udostępnianych w ramach języka C# i bibliotek WCF i WPF w rozwiązywaniu problemów inżynierskich - [K1st_K2]</p>

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:

- ocenę wiedzy i umiejętności związanych z realizacją kolejnych zadań projektowych / laboratoryjnych,
- ocenę i "obronę" przez studenta/studentów projektu końcowego,
- przeglądowy test sprawdzający wiedzę przekazywana podczas wykładu.

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.

Treści programowe		
<p>Wykład.</p> <p>W ramach wykładu omawiane są następujące zagadnienia:</p> <ul style="list-style-type: none"> - Wprowadzenie do środowiska .NET i języka C# - Podstawowe elementy języka C#: rodzaje danych, wbudowane typy zmiennych, funkcje i właściwości dostępne dla typów danych, łańcuchy znaków, tablice, struktury i klasy. - Zaawansowane elementy języka: indeksatory, konwersja typów, typy parametryczne (generics), interfejsy, iteratory, typy i metody anonimowe, delegacje, wyrażenia lambda, zdarzenia, wyjątki, zapytania LINQ, współbieżność, - Architektura ADO.NET - Biblioteka WPF: architektura aplikacji, XAML, pojemniki, zdarzenia, Dependency Objects, wiązanie danych, style, tworzenie komponentów, zasoby, animacje, elementy graficzne, system komend, model MVVM, weryfikacja danych formularza, widoki danych. - Wprowadzenie do WCF. <p>Laboratorium.</p> <p>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.</p> <p>W pierwszym etapie konsolowe aplikacje pozwalają na pokazanie zaawansowanych konstrukcji języka C#. 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)). Równolegle, na bazie zdobytej wiedzy studenci rozwijają indywidualnie lub w niewielkich zespołach bardziej rozbudowaną, warstwową aplikację z interfejsem użytkownika (WPF), realizowaną zgodnie z modelem MVVM.</p>		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Troelsen, Andrew W., Japikse, Philip F, Język C# 6.0 i platforma .NET 4.6, PWN, 2017. 2. Michaelis, Mark., Lippert, Eric. , C# 6.0 : kompletny przewodnik dla praktyków, Helion, 2016. 3. Raffaele Garofalo, Budowanie aplikacji biznesowych za pomocą Windows Presentation Foundation i wzorca Model View ViewM, PROMISE, 2011 4. Dokumentacja elektroniczna systemu programowania wizualnego Visual Studio.NET 5. Dokumentacja języka C# i bibliotek WPF i WCF 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Matthew MacDonald, Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5, Apress, 2012. 2. Magennis, Troy, LINQ to Objects w C# 4.0 : wygodne operacje na danych!, Helion, 2012. 3. Metsker, Steven John, C# - wzorce projektowe, Helion, 2005. 4. WCF ? Getting started tutorial https://docs.microsoft.com/en-us/dotnet/framework/wcf/getting-started-tutorial 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych:	30	
2. przygotowanie do ćwiczeń laboratoryjnych:	5	
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych:	5	
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2 10	
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	5	
6. przygotowanie do sprawdzianów / kolokwium	30	
7. udział w wykładach	10	
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	8	
9. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	105	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	50	2