

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie wizualne		Kod 1010511371010510452
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) 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 z kodem, którego jakość pozwala na ponowne wykorzystanie części kodu w innych projektach programistycznych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. 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>2. zna szczegółowo język programowania obiektowego C# i jego zaawansowane konstrukcje (m. in. wyrażenia lambda, funkcje rozszerzające, wyrażenia LINQ) - [K1st_W4]</p> <p>3. zna podstawowe wzorce projektowe stosowane przy tworzeniu aplikacji ze szczególnym uwzględnieniem wzorca Model-View-ViewModel - [K1st_W4]</p> <p>4. zna biblioteki klas i funkcji środowiska programowego .NET umożliwiające tworzenie aplikacji okienkowych Windows Presentation Foundation - [K1st_W5]</p> <p>5. 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. potrafi wybrać sposób zabezpieczenia interfejsu użytkownika aplikacji okienkowej wykorzystującej biblioteki Windows Presentation Foundation przed wprowadzaniem błędnych danych, krytycznie ocenić wady i zalety zastosowanego rozwiązania - [K1st_U1]</p> <p>2. 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_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 - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prostą bazę danych opartą o relację Producent - Produkt używając właściwych metod, technik i narzędzi - [K1st_U10]</p> <p>5. Student potrafi zaimplementować algorytm filtrowania i sortowania danych w aplikacji z interfejsem użytkownika - [K1st_U11]</p> <p>6. potrafi znaleźć rozwiązanie wybranych problemów programistycznych w ramach technologii .NET i WPF - [K1st_U19]</p>
<p>Kompetencje społeczne:</p> <p>1. Student rozumie, że platforma .NET ciągle się rozwija i w kolejnych wersjach wprowadzane są nowe elementy języka C# - [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>

<p>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 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 realizacji kolejnych etapów projektu zaliczeniowego. Projekt zaliczeniowy składa się z czterech obligatoryjnych i jednego opcjonalnego etapu. Każdy z nich jest sprawdzany i omawiany. - ocenę wiedzy zdobytej na wykładzie w formie albo krótkich, 10 minutowych testów przeprowadzanych na wykładzie i obejmujących 1-2 wykłady wstecz z wykorzystaniem platformy moodle albo w formie kolokwium po zakończeniu wykładów. Każdy z testów przeprowadzanych na wykładach oceniany jest tak samo, przy czym maksymalną liczbę punktów wyznacza najlepszy uzyskany wynik. Względem niego wyznaczana jest ocena procentowa. Aby uzyskać zaliczenie w tej formie, należy uzyskać średnią z testów min. 50%, przy czym dwa najniższe wyniki testów nie są brane pod uwagę. Student, który nie uzyskał wymaganych punktów ma obowiązek przyjść kolokwium zaliczeniowe po skończeniu wykładów.
<p>Treści programowe</p>
<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

<p>danych. - Wprowadzenie do WCF.</p> <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 jako aplikacje konsolowe w celu pokazania 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. Aplikacja ta, realizowana jest w pięciu etapach (4 obowiązkowych i jednym opcjonalnym), przy czym każdy jest oceniany przez prowadzącego. Dodatkowo wpływ na ocenę ma terminowość realizacji poszczególnych etapów.</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 		
<p>Bilans nakładu pracy przeciętnego studenta</p>		
<p>Czynność</p>		<p>Czas (godz.)</p>
<p>1. udział w zajęciach laboratoryjnych:</p>		<p>30</p>
<p>2. przygotowanie do ćwiczeń laboratoryjnych:</p>		<p>5</p>
<p>3. dokończenie (w ramach pracy własnej) ćwiczeń laboratoryjnych:</p>		<p>5</p>
<p>4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu</p>		<p>2 10</p>
<p>5. napisanie programu zaliczeniowego, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)</p>		<p>5</p>
<p>6. przygotowanie do sprawdzianów / kolokwium</p>		<p>30</p>
<p>7. udział w wykładach</p>		<p>10</p>
<p>8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron</p>		<p>8</p>
<p>9. przygotowanie do zaliczenia wykładów w wybranej formie</p>		
<p>Obciążenie pracą studenta</p>		
<p>forma aktywności</p>	<p>godzin</p>	<p>ECTS</p>
Łączny nakład pracy	105	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	50	2