

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Programowanie aplikacji desktopowych w środowisku .NET		Cykl kształcenia rozpoczynający się w roku akademickim 2022/2023	
Nazwa kierunku studiów, poziom i profil kształcenia: informatyka, studia I-go stopnia, profil praktyczny			
Język wykładowy: polski		Rodzaj zajęć: zajęcia specjalistyczne	
Rok studiów: III		Semestr: V	
Liczba punktów ECTS przypisana zajęciom: 2		Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:	
Jednostka organizacyjna: Instytut Inżynierii Technicznej/Zakład Informatyki			
FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN			
Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:			
Studia stacjonarne		Studia niestacjonarne	
Wykład:		Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	30	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	30	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			
Wymagania wstępne i dodatkowe: wymagana jest wiedza i umiejętności zdobyte na zajęciach kształcenia kierunkowego - <i>podstawy programowania, algorytmy i struktury danych, programowanie obiektowe</i>			

Cel (cele) kształcenia dla zajęć: rozwinięcie i uzupełnienie wiedzy w zakresie paradygmatów programowania obiektowego, zdobycie podstawowych umiejętności pozwalających na budowanie aplikacji desktopowych z GUI w środowisku języka C#, z uwzględnieniem zapytań do bazy danych		
EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW		
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się UWAGA: Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.		
Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie		
W_01	koncepcję architektury platformy .NET, zasady budowania i uruchamiania kodu źródłowego w języku C#, zna cechy oraz koncepcje i paradygmaty programowania obiektowego w języku C#	K_W06, K_W08
Umiejętności - potrafi		
U_01	posługiwać się środowiskiem programistycznym w procesie tworzenia i kontroli aplikacji	K_U08, K_U11, K_U12, K_U14
U_02	realizować aplikacje konsolowe oraz aplikacje z interfejsem graficznym, w tym aplikacje wykorzystujące zewnętrzny system bazy danych	K_U08, K_U09, K_U11, K_U12, K_U14, K_U20
U_03	samodzielnie odnaleźć stosowne informacje, zawarte w dokumentacji technicznej platformy .NET i języka C#	K_U01, K_U05, K_U06
Kompetencji społecznych - jest gotów do		
K_01	podnoszenia swoich kwalifikacji poprzez samokształcenie	K_K01
UWAGA! Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.		
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA		
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):		

Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		zajęcia praktyczne		
TP-01	Ogólna charakterystyka i funkcjonowanie środowiska uruchomieniowego platformy .NET. Struktura kodu w języku C#, charakterystyka typów wartościowych i referencyjnych, realizacja prostych aplikacji konsolowych - analiza elementów zestawu. Deasemblacja kodu pośredniego z wykorzystaniem <i>ildasm</i> . Dokumentacja języka.		pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych, realizacja ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych

TP-02	Realizacja aplikacji konsolowych, implementujących model obiektowy języka C#: definiowanie własnych klas i ich instancji, wdrażanie konstruktorów, przeciążanie konstruktorów, wdrażanie właściwości. Tworzenie własnych bibliotek klas DLL. Instalacja bibliotek zewnętrznych bibliotek klas przy pomocy menadżera pakietów NuGet. Kontrola działania aplikacji - wykorzystanie mechanizmów debuggera w Visual Studio.		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych
-------	---	--	---	---

TP-03	Realizacja aplikacji uwzględniających tworzenie metod wraz z mechanizmami przekazywania i zwracania parametrów, a także przeciążania metod. <i>Implementation of applications involving the creation of methods with mechanisms of passing and returning parameters, as well as method overloading.</i>		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych
TP-04	Realizacja aplikacji z wykorzystaniem mechanizmów dziedziczenia, znaczenie klasy Object. Realizacja aplikacji konsolowych implementujących interfejsy wewnątrz klas. Obsługa błędów i wyjątków.		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych

TP-05	Konceptcja klas generycznych - tworzenie własnych rozwiązań. Wykorzystanie klas kolekcji: stosy, listy, kolejki. Realizacja aplikacji operujących na danych w kolekcjach. / <i>The concept of generic classes - creating your own solutions. Use of collection classes: stacks, lists, queues. Implementation of applications operating on data in collections.</i>		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych
TP-06	Delegacje i zdarzenia - realizacja aplikacji konsolowych z uwzględnieniem mechanizmów delegacji i zdarzeń.		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych

TP-07	Projektowanie i realizacja aplikacji z wykorzystaniem GUI opartego o kontrolki WPF.		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych
TP-08	Połączenie Visual Studio i SQL Server - wykorzystanie budowa aplikacji bazodanowych. Odpytywanie bazy danych przy użyciu wyrażeń LINQ.		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych
TP-09	Koncepcja izolacji aplikacji w środowisku .NET. Uruchamianie aplikacji we własnych domenach aplikacji - badanie wpływu takiego podejścia na procesy. Zajęcia zaliczeniowe		realizacja aplikacji, pogadanka związana z aspektami teoretycznymi w odniesieniu do realizowanych ćwiczeń praktycznych	ocena zrealizowanych aplikacji ocena umiejętności posługiwania się środowiskiem programistycznym obserwacja pracy studenta w trakcie realizacji zadań praktycznych

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Michaelis M.: C# 70. *Kompletny przewodnik dla praktyków*, Wyd. Helion 2019r.
2. Posadas M.: *Tajniki C# i .NET Framework*, Wyd. Promise 2017r.
3. Troelsen A, Philip J: *Język C# 6.0 i platforma .NET 4.6*, Wyd. PWN 2017r.
4. Sharp J.: *Microsoft Visual C# 2017 krok po kroku*, wyd. Promise 2017r.

Literatura uzupełniająca:

1. Dokumentacja elektroniczna języka C# <https://docs.microsoft.com/pl-pl/dotnet/csharp/>
2. Dokumentacja elektroniczna platformy Windows Presentation Foundation <https://learn.microsoft.com/pl-pl/dotnet/desktop/wpf/?view=netdesktop-7.0>

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)

Forma aktywności	Liczba godzin *
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	30
Praca własna studenta	25
SUMA GODZIN:	55

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 2	1
	Praca własna studenta		1

* godziny lekcyjne, czyli 1 godz. oznacza 45 min;

OPIS PRACY WŁASNEJ STUDENTA:

Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbole efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej.

Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.

Czytanie wskazanej literatury/dokumentacji (W_01, U_03, K_01), przygotowanie do zajęć (U_01-U_03, K_01), wykonanie aplikacji zaliczeniowych (W_01, U_01-U_03, K_01),

KRYTERIA OCENIANIA

Ocena kształtująca:

ocena przygotowania do zajęć

ocena umiejętności stosowania zdobytej wiedzy podczas realizowanych podczas zajęć aplikacji

ocena aktywności podczas zajęć

Ocena podsumowująca:

ocena umiejętności rozwiązywania postawionych problemów

ocena zadań implementacyjnych do samodzielnego wykonania

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ