

# KARTA PRZEDMIOTU (SYLABUS)

## Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	WIZUALNE SYSTEMY PROGRAMOWANIA	
IT/P/I/NST/B <sub>1</sub> -17			VISUAL PROGRAMMING	
Język wykładowy		język polski		
Rok akademicki		2019/2020		
Kierunek		Informatyka techniczna		
w zakresie				
Poziom studiów		studia pierwszego stopnia		
Profil studiów		praktyczny		
Forma studiów		studia niestacjonarne		
Semestr / semestry		semestr trzeci zimowy		
Przynależność do grupy zajęć		B – grupa przedmiotów kierunkowych		
Status przedmiotu		obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	15 [h]	4 ECTS
		Ćwiczenia	15 [h]	
		...	...	
Powiązanie przedmiotu	z profilem studiów	kształtuje umiejętności praktyczne		2 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		4 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja		4 ECTS
Forma nauczania		tradycyjna - zajęcia zorganizowane w Uczelni		
Wymagania wstępne		Wymagana znajomość: Podstaw programowania i Programowania Obiektowego		
Jednostka prowadząca		Katedra Informatyki		
Koordynator		dr Artur Bartoszewski		
Osoby prowadzące		dr Artur Bartoszewski		
Adres strony internetowej pjo		www.wim.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		artur.bartoszewski@uthrad.pl		

**EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ**

Cel kształcenia:	<p>1. Praktyczne zapoznanie studentów z nowoczesnymi technikami programowania.</p> <p>2. Praktyczne zapoznanie studentów z nowoczesnymi środowiskami programistycznymi na przykładzie Microsoft Visual Studio 2018.</p> <p>3. Zapoznanie studentów z podstawami programowania w języku C#.</p> <p>4. Zachęcenie studentów do dalszego, samodzielnego poszerzania swojej wiedzy i umiejętności w dziedzinie programowania</p>
Treści programowe:	<p><b>Wykład</b></p> <p>Windows Forms i WPF [1h] U2</p> <p>Struktura kodu programu C#, podobieństwa i różnice względem C++, Podstawowe struktury języka C#. [1h]</p> <p>Obiektywność w języku C#.[1h] W1</p> <p>Operacje na tekstach – łańcuchy i klasa String. Kolekcja (tablica i kolekcja list) [1h] W1</p> <p>Budowa projektu aplikacji okienkowej. Podstawowe kontrolki systemu Windows i ich właściwości. [1h] W1 U2</p> <p>Oprogramowanie przycisków. Komunikacja z użytkownikiem za pomocą kontrolki Label i TextBox [2h] U1 U2</p> <p>System zdarzeń i metody obsługi zdarzeń związane z kontrolkami. [2h] U3</p> <p>System menu i okien dialogowych.[2h] U2</p> <p>Zdarzenia myszy i klawiatury oraz ich obsługa. [1h] U3</p> <p>Grafika GDI+, Klasa Graphics i jej możliwości. [2h] U4</p> <p>Klasa Timer. Animacja z wykorzystaniem timer-a.[1h] U4</p> <p><b>Ćwiczenia laboratoryjne</b></p> <p>Posługiwanie się środowiskiem programistycznym MS Visual Studio. Budowa i własności projektu. Uruchamianie i debatowanie programu. [1h] U1</p> <p>Tworzenie aplikacji konsolowych w języku C#. Składnia języka C#, [2h] W1 U1</p> <p>Programowanie zorientowane obiektowo - tworzenie własnych klas i wykorzystanie ich w programach konsolowych. [2h] W1 U1</p> <p>Tworzenie aplikacji Windows Forms. [4h] W1 U1 U2</p> <p>Kolekcje, generator liczb pseudolosowych - przykłady: „Losowanie Totolotka” „Liczby pierwsze - sito Erastotenesa”. [2h] W1 U2</p> <p>Obsługa zdarzeń myszy i klawiatury. [1h] U3</p> <p>Programowa grafika 2D - klasy Graphics, Brush i Pen, rysowanie linii, prostokątów elips itp. [2h] U4</p> <p>Generowanie i przetwarzanie bitmapy (przykład fraktal Sierpińskiego) [1h] U4</p>
Metody dydaktyczne (kształcenia):	<p>Metody podające - wykład informacyjny – <b>W1</b></p> <p>Metody programowane z wykorzystaniem komputera – ćwiczenia laboratoryjne- <b>U1,U2, U3, U4, K1</b></p> <p>Wszystkie zastosowane metody umożliwiają rozpoznawanie i zaspokajanie indywidualnych potrzeb studentów, w tym studentów niepełnosprawnych oraz indywidualizację toku studiów.</p>
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	<p>Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów kształcenia określonych dla danego przedmiotu</p> <p><b>Ćwiczenia laboratoryjne:</b></p> <p>dwa sprawdziany praktyczne oceniające umiejętności programistyczne</p> <p><b>Wykład:</b></p> <p>zaliczenie w postaci testu praktycznego wraz z odpowiedzią ustną.</p>

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi /(K) jest gotów do:	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny
W1	Student zna składnię języka C# oraz metody programowania obiektowego i wizualnego	K_WG05	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium

U1	Potrafi praktycznie wykorzystać środowisko MS Visual Studio do tworzenia interfejsu użytkownika programu z użyciem typowych komponentów Windows Forms i WPF	K_UW04	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium
U2	Potrafi zbudować program z wykorzystaniem kontrolek ekranowych, systemu menu i okien dialogowych (do sterowania aplikacją oraz przyjmowania i wyprowadzania danych)	K_UW04	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium
U3	Potrafi oprogramować system zdarzeń związanych z urządzeniami peryferyjnymi (mysz, klawiatur itp.)	K_UW04	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium
U4	Potrafi wykorzystać GDI+ do generowania grafiki 2D oraz prostych animacji.	K_UW04	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	
K1	Jest gotowy poszerzać swoją wiedzę i umiejętności podążając za rozwojem narzędzi i technik programistycznych	K_KK01	wykład, ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium
Stopień osiągnięcia kierunkowych efektów uczenia się: K_WG05+++ K_UW04+++ K_UW05++ K_KK01+++					

Literatura podstawowa, literatura uzupełniająca, pomoce naukowe

**Literatura podstawowa:**

1. Greene J., Stellman A., C#. Rusz głową! Wydanie III, Helion 2017,
2. Matulewski J., Visual Studio 2017 tworzenie aplikacji Windows w języku C#, Helion 2018,
3. Sharp J., Microsoft Visual C# 2017 Krok po kroku, Helion 2018

**Literatura uzupełniająca:**

1. Autor: Sharp J., Microsoft Visual C# 2017. Krok po kroku, Wydawnictwo APN Promise, Warszawa 2018
2. Liberty J., Programming C# 3.0, Wydawnictwo O'Reilly, 2002

Autorskie materiały zamieszczone na stronie [www.bartoszewski.uniwersytetradom.pl](http://www.bartoszewski.uniwersytetradom.pl)

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS

Udział w zajęciach, aktywność	Obciążenie studenta [h]		
	Inne godz. kontaktowe (IGK)	Zajęcia bez nauczyciela-praca własna studenta (ZBN)	Zajęcia dydaktyczne
Udział w wykładach	X	X	15 [h]
Samodzielne studiowanie tematyki wykładów	X	30 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	15 [h]
Samodzielne przygotowanie się do ćwiczeń	X	30 [h]	X
Udział w konsultacjach	3 [h]	X	X
Przygotowanie do zaliczenia	X	10 [h]	X
Udział w zaliczeniu	2 [h]	X	X
Sumaryczne obciążenie pracą studenta	5 [h]/ 0,2 ECTS	70 [h]/ 2,6 ECTS	30 [h]/ 1,2 ECTS
Punkty ECTS za przedmiot	4 ECTS		

Informacje dodatkowe, uwagi

W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.

Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekłe chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych.

Studentowi przysługuje jeden termin podstawowy i jeden termin poprawkowy zaliczenia dla każdej formy zajęć. Obecność na zajęciach laboratoryjnych jest obowiązkowa. Obecność na wykładach jest zalecana i może być premiowana. W przypadku zaległości powstałych

wskutek nieobecności studenta na zajęciach projektowych student jest zobowiązany do uczestnictwa w zajęciach innej grupy (tzw. odrobienie zajęć) lub wykonania (w przypadku braku możliwości odrobienia) i zaliczenia dodatkowego projektu.

Zgodnie z Regulaminem Studiów UTHRad podstawowym terminem uzyskania zaliczenia jest ostatni dzień zajęć w danym semestrze. Termin zaliczenia poprawkowego (tryb i warunki ustala prowadzący moduł na zajęciach początkowych) nie może być późniejszy niż ostatni termin egzaminu w sesji poprawkowej (dla przedmiotów kończących się egzaminem) lub ostatni dzień trwania semestru (dla przedmiotów niekończących się egzaminem).

Terminy odbywania zajęć: semestr zimowy oraz semestr letni, zgodnie z rozkładem zajęć

Miejsce odbywania zajęć: ul. Malczewskiego 20A, Radom;