

## KARTA PRZEDMIOTU (SYLABUS)

## Opis przedmiotu

Kod przedmiotu	Nazwa przedmiotu	JĘZYKI I PARADYGMATY PROGRAMOWANIA		
I/O/1/NST/B1-13		PROGRAMMING LANGUAGES AND PARADIGMS		
Język wykładowy	język polski			
Rok akademicki	2024/2025			
Kierunek	Informatyka			
w zakresie	-			
Poziom studiów	studia pierwszego stopnia			
Profil studiów	ogólnoakademicki			
Forma studiów	studia niestacjonarne			
Semestr / semestry	5			
Przynależność do grupy zajęć	B1. Grupa zajęć kierunkowych - obowiązkowych			
Status przedmiotu	obowiązkowy			
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	18 [h]	5 ECTS
		Laboratorium	24 [h]	
Powiązanie przedmiotu	z profilem studiów	związany z prowadzoną działalnością naukową w dyscyplinach, do których przyporządkowany jest kierunek studiów		3 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		4 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja		5 ECTS
Forma nauczania		tradycyjna – zajęcia zorganizowane w Uczelni i/lub zajęcia z wykorzystaniem metod i technik kształcenia na odległość (max. 0,7 ECTS)		
Wymagania wstępne		Wymagana znajomość przedmiotów: podstawy programowania, programowanie obiektowe, programowanie niskopoziomowe		
Jednostka prowadząca		Katedra Informatyki i Teleinformatyki		
Koordynator		dr Artur Hermanowicz		
Adres strony internetowej pjo		<a href="http://www.wteii.uniwersytetradom.pl">www.wteii.uniwersytetradom.pl</a>		
Adres e-mail, telefon koordynatora		artur.hermanowicz@urad.edu.pl, +48 48 361 78 21		

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

Cel kształcenia:	Zapoznanie z najważniejszymi paradygmatami występującymi we współczesnym programowaniu: programowanie proceduralne i obiektowe, imperatywne i funkcyjne. Na wykładzie przedstawiane są również skrótowo paradygmaty programowania w logice. Ćwiczenia o charakterze laboratoryjnym pozwalają pogłębić znajomość języków programowania i zrozumieć podstawowe zagadnienia implementacyjne.
Treści programowe:	<p>Wykład [BN, W1]:</p> <ol style="list-style-type: none"> <li>1. Programowanie proceduralne: składnia i semantyka języków programowania, metody opisu składni, typy, podprogramy (procedury i funkcje). Metody przekazywania parametrów do podprogramów: przez wartość, zmienną, referencję.</li> <li>2. Programowanie imperatywne: zmienne, struktura programu, organizacja wywołań podprogramów, przydział pamięci na stosie i na stercie.</li> <li>3. Programowanie obiektowe: pojęcie klasy i obiektu, zagadnienia związane z kapsułkowaniem, dziedziczeniem i polimorfizmem. Późne (dynamiczne) wiązanie wywołań. Wyjątki. Szablony. Środowisko Eclipse i programowanie w języku Java – tworzenie aplikacji konsolowych, appletów i aplikacji okienkowych. Zarządzanie pamięcią i odśmiecanie. Programowanie w języku C# w środowisku MS Visual Studio. Ocena przydatności różnych środowisk programistycznych z uwzględnieniem projektowania, implementowania, testowania i debugowania programów obiektowych.</li> <li>4. Programowanie funkcyjne: funkcje jako model programowania, rachunek lambda, działania na listach, dopasowywanie wzorca, nadawanie typów, rekursja. Procedury jako argumenty i wyniki.</li> <li>5. Programowanie w logice: rachunek predykatów w języku Prolog, rezolucja, listy.</li> </ol> <p style="text-align: right;">Suma: 18 [h]</p> <p>Laboratorium [BN, U1, U2, K1]:</p> <ol style="list-style-type: none"> <li>1. Zapoznanie się ze środowiskiem programistycznym Eclipse. Tworzenie prostych aplikacji konsolowych w Javie.</li> </ol>

	2. Programowanie obiektowe w Javie. Działania na łańcuchach i tablicach. Obsługa plików tekstowych i binarnych. 3. Budowa aplikacji okienkowej w Javie, programowanie kontrolek, projektowanie interfejsu użytkownika, obsługa zdarzeń. 4. Wyświetlanie grafiki w oknie, animacja. 5. Programowanie funkcyjne w Haskellu. 6. Programowanie w logice z wykorzystaniem języka Prolog.
	Suma: 24 [h]
Metody dydaktyczne (kształcenia):	– metody podające (wykład informacyjny), – metody problemowe (wykład problemowy, wykład konwersatoryjny), – metody aktywizujące (dyskusja dydaktyczna), – metody eksponujące (film, ekspozycja, pokaz), – metody programowane (z wykorzystaniem komputera), – metody praktyczne (pokaz, ćwiczenia laboratoryjne, rachunkowe, symulacja)
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów kształcenia określonych dla danego przedmiotu. Uzyskanie pozytywnych ocen ze wszystkich form zajęć wchodzących w skład danego przedmiotu jest równoznaczne z jego zaliczeniem i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi. Sposób obliczenia oceny końcowej z przedmiotu określony został w regulaminie studiów. Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco: Na ocenę z laboratorium składa się: punktowa ocena sprawdzianów praktycznych przy komputerze (90%), punktowa ocena aktywności na zajęciach (10%). Ocena z egzaminu – wynik otwartego testu pisemnego. Zdobyte w poszczególnych formach zajęć punkty przeliczane zostają na ocenę wg skali: Ocena 2 poniżej 51% Ocena 3 od 51% Ocena 3,5 od 61% Ocena 4 od 71% Ocena 4,5 od 81% Ocena 5 od 91%

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	najważniejsze paradygmaty występujące we współczesnym programowaniu: imperatywne, proceduralne, obiektowe, funkcyjne i logiczne, możliwości i różnice w implementacji algorytmów oraz wykorzystania struktur danych w zależności od wybranego paradygmatu programowania	K_WG06 K_WG08	wykład	egzamin pisemny	pisemny test otwarty
U1	zaimplementować aplikację konsolową oraz wyposażoną w graficzny interfejs użytkownika, wykorzystać w programie struktury danych potrafiąc ocenić ich przydatność uwzględniając złożoność pamięciową i obliczeniową,	K_UW12 K_UW18	laboratorium	zaliczenie	sprawdzian praktyczny przy komputerze
U2	zaimplementować algorytm, stosując wybrany język programowania i paradygmat, w celu przeprowadzenia eksperymentu oraz przedstawić jego wyniki	K_UK20	laboratorium	zaliczenie	sprawdzian praktyczny przy komputerze
K1	obserwacji zmian zachodzących w podejściu do programowania oraz powstawania nowych możliwości będąc świadomym konieczności aktualizowania i poszerzania wiedzy w zakresie nauk informatycznych	K_KK01 K_KK02	wykład / laboratorium	obserwacja	dyskusja, aktywność na zajęciach, prezentacja wyników prac

Literatura i pomoce naukowe	
1.	Abelson H., Sussman G.J., Sussman J.: Struktura i interpretacja programów komputerowych, WNT, Warszawa 2002
2.	Clocksinn W.F., Mellish C.S.: Prolog. Programowanie, Helion, Gliwice 2002
3.	Eckel B.: Thinking in Java. Edycja polska, wyd. 4, Helion, Gliwice 2006
4.	Horstmann C.S.: Java. Podstawy, wyd. XII, Helion, Gliwice 2022
5.	Skinner R.: Effective Haskell, Pragmatic Bookshelf 2023
6.	Van Roy P., Haridi S.: Programowanie. Koncepcje, techniki i modele, Helion, Gliwice 2005
7.	Weidig B.: Java. Podejście funkcyjne, Helion, Gliwice 2024

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	18 [h]
Udział w ćwiczeniach / laboratoriach / projektach / seminariach	X	X	24 [h]
Udział w konsultacjach	4 [h]	X	X
Przygotowanie do wykładów / ćwiczeń / laboratoriów / projektów / seminariów	X	79 [h]	X
Przygotowanie do zaliczenia/egzaminu			
Sumaryczne obciążenie pracą studenta	4 [h] / 0,2 ECTS	79 [h] / 3,1 ECTS	42 [h] / 1,7 ECTS
Punkty ECTS za przedmiot	5 ECTS		

Informacje dodatkowe, uwagi
<p>W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.</p> <p>Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekle 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, przewlekle chorych.</p>