

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu	Nazwa przedmiotu	ALGORYTMY I ZŁOŻONOŚĆ		
IT/P/1/ST/B1-7		ALGORITHMS AND COMPLEXITY		
Język wykładowy	język polski			
Rok akademicki	2024/2025			
Kierunek	Informatyka techniczna			
w zakresie	-			
Poziom studiów	studia pierwszego stopnia			
Profil studiów	praktyczny			
Forma studiów	studia stacjonarne			
Semestr / semestry	3			
Przynależność do grupy zajęć	B 1. Grupa zajęć kierunkowych - obowiązkowych			
Status przedmiotu	obowiązkowy			
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS	Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS	
	Wykład	30 [h]	5 ECTS	
	Laboratorium	30 [h]		
Powiązanie przedmiotu	z profilem studiów	związany z kształtowaniem umiejętności praktycznych		2,5 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		3 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. 1,2 ECTS)			
Wymagania wstępne	Wymagana dobra znajomość przedmiotu analiza matematyczna, bardzo dobra znajomość przedmiotu podstawy programowania oraz programowanie obiektowe.			
Jednostka prowadząca	Katedra Informatyki i Teleinformatyki			
Koordynator	dr Monika Maj			
Adres strony internetowej pjo	www.wteii.uniwersytetradom.pl			
Adres e-mail, telefon koordynatora	m.maj@urad.edu.pl;			

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

Cel kształcenia:	Poznanie i opanowanie skutecznych metod rozwiązywania zadań użytecznych w wykonywaniu zawodu informatyka, w szczególności: biegłego stosowania podstawowych algorytmów i szacowania ich złożoności obliczeniowej, technik projektowania algorytmów, dobierania odpowiednich struktur danych do zadania oraz stosowania statycznych i dynamicznych struktur danych.
Treści programowe:	<p>Wykład [W1, W2]:</p> <p>Pojęcie algorytmu, metody zapisu algorytmów: opis słowny, lista kroków, schematy blokowe, pseudokod programowania, język programowania.</p> <p>Weryfikacja i poprawność algorytmu. Przegląd metod projektowania algorytmów: metoda „dziel i zwyciężaj”, metoda zachłanna, programowanie dynamiczne, metoda prób i błędów. Złożoność obliczeniowa, notacja asymptotyczna. Przeszukiwanie sekwencyjne i binarne. Rekurencja a iteracja. Proste algorytmy sortowania: Bubblesort, Selectionsort i wersja z wartownikiem, Insertionsort i wersja z wyszukiwaniem binarnym. Szybkie algorytmy sortowania: Quicksort i Heapsort, sortowanie Shella. Pojęcie typu danych, proste i złożone typy danych. Podstawowe struktury danych: tablice, rekordy, zbiory, pliki. Złożone struktury danych: listy, stosy, kolejki, kolejki priorytetowe, grafy, drzewa binarne i wielokierunkowe, słowniki, tablice mieszające. Efektywne algorytmy słownikowe, drzewa AVL, B-drzewa.</p> <p>Algorytmy przeszukiwania grafu w głąb i wszerz oraz ich zastosowania. Znajdowanie najkrótszych dróg w grafie: algorytm Dijkstry, algorytm Floyda-Warshalla. Znajdowanie drzewa rozpinającego grafu. Przykłady uzasadnionej rekurencji: wieże Hanoi, kreślenie krzywych Hilberta i Sierpińskiego, problem plecakowy, problem ośmiu hetmanów.</p> <p>Wyszukiwanie wzorców w tekstach. Kryptografia. Problemy obliczeniowo trudne: pojęcie NP zupełności, klasy P i NP. nierozstrzygalność.</p> <p style="text-align: right;">Suma: 30 [h]</p>

	<p>LABORATORIUM [PP, U1, U2, K1, K2]:</p> <p>Podczas laboratorium studenci tworzą i uruchamiają aplikacje konsolowe w języku programowania wysokiego poziomu wykorzystując poznane na wykładach algorytmy, które dotyczą następujących zagadnień:</p> <ol style="list-style-type: none"> 1. Przeszukiwanie sekwencyjne i binarne, 2. Proste algorytmy sortowania: Bubblesort, Selectionsort i wersja z wartownikiem, Insertionsort i wersja z wyszukiwaniem binarnym, 3. Szybkie algorytmy sortowania: Quicksort i Heapsort, sortowanie Shella, 4. Proste i złożone typy danych, 5. Podstawowe struktury danych: tablice, rekordy, zbiory, pliki, 6. Złożone struktury danych: listy, stosy, kolejki, kolejki priorytetowe, grafy, drzewa binarne i wielokierunkowe, słowniki, tablice mieszające, 7. Efektywne algorytmy słownikowe, drzewa AVL, B-drzewa, 8. Algorytmy przeszukiwania grafu w głąb i wszerz oraz ich zastosowania, 9. Znajdowanie najkrótszych dróg w grafie: algorytm Dijkstry; znajdowanie drzewa rozpinającego grafu, 10. Przykład uzasadnionej rekurencji np.: wieże Hanoi, kreślenie krzywych Hilberta i Sierpińskiego, problem plecakowy, problem ośmiu hetmanów, 11. Wyszukiwanie wzorców w tekstach; kryptografia, 12. Problemy obliczeniowo trudne: pojęcie NP zupełności, klasy P i NP; nierozstrzygalność. <p style="text-align: right;">Suma: 30 [h]</p>
Metody dydaktyczne (kształcenia):	<ul style="list-style-type: none"> – metody podające (wykład informacyjny), – metody problemowe (wykład problemowy, wykład konwersatoryjny), – metody aktywizujące (metoda przypadków, dyskusja dydaktyczna,), – metody praktyczne (ćwiczenia laboratoryjne) – metody programowane z wykorzystaniem komputera
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	<p>Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów uczenia się 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 obliczania oceny z poszczególnych form zajęć przedstawia się następująco:</p> <p>Wykład: Ocena końcowa z przedmiotu stanowi 100% oceny z egzaminu. Egzamin przeprowadzany jest w formie pisemnej, polega na samodzielnym rozwiązaniu przez studenta testu oraz zadań algorytmicznych z zakresu tematyki omawianej na wykładach.</p> <p>Laboratorium: w czasie laboratorium student zalicza dwa kolokwia polegające na samodzielnym wykonaniu i uruchomieniu programu komputerowego z zakresu tematyki przedmiotu.</p> <p>Ocena końcowa z laboratorium stanowi średnia ocen z kolokwiów.</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	algorytmikę oraz teorię złożoności obliczeniowej dotyczącą stosowania podstawowych znanych algorytmów oraz szacowania ich złożoności	K_WG05 K_WG06 K_WG08	wykład	egzamin pisemny	egzamin pisemny
W2	metody i techniki projektowania algorytmów, dobierania odpowiednich struktur danych do zadania oraz zna metody stosowania statycznych i dynamicznych struktur danych	K_WG05 K_WG06 K_WG08	wykład	egzamin pisemny	egzamin pisemny
U1	formułować algorytmy i je programować, posługując się językami wysokiego poziomu oraz potrafi ocenić ich złożoność obliczeniową	K_UW12 K_UW18	laboratorium	zaliczenie na ocenę	wykonanie ćwiczeń laboratoryjnych, kolokwia
U2	wykorzystać nabytą wiedzę matematyczną do zapisu algorytmów, a także zastosować zaawansowane struktury danych do ich realizacji	K_UW05	laboratorium	zaliczenie na ocenę	wykonanie ćwiczeń laboratoryjnych, kolokwia
K1	wykonywania zawodu programisty wykazując się rzetelnością, bezstronnością, profesjonalizmem i etyczną postawą	K_KR08	laboratorium	obserwacja	aktywność na zajęciach, dyskusja

K2	planowania pracy pod kątem zakładanych rezultatów oraz określania priorytetów podczas ich wykonywania.	K_KO05	laboratorium	obserwacja	aktywność na zajęciach, dyskusja
----	--	--------	--------------	------------	----------------------------------

Literatura i pomoce naukowe					
1. Banachowski L., Diks K., Rytter W.: Algorytmy i struktury danych, WNT, Warszawa 2006 2. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C.: Wprowadzenie do algorytmów, WNT, Warszawa 2022 3. Jakubczyk K.: Wprowadzenie do algorytmów i struktur danych, Wyd. Politechniki Radomskiej, Radom 2005, 2007 4. Wirth N.: Algorytmy + struktury danych = programy, WNT, Warszawa 2004 5. Drozdek A.: C++. Algorytmy i struktury danych, Helion, Gliwice 2004 6. Harel D., Feldman Y.: Rzecz o istocie informatyki. Algorytmika, WNT, Warszawa 2008 7. Knuth D.E.: Sztuka programowania, t. 1 – Algorytmy podstawowe, t. 2 – Algorytmy seminumeryczne, t. 3 – Sortowanie i wyszukiwanie, WNT, Warszawa 2002 8. Wróblewski P.: Algorytmy, struktury danych i techniki programowania, Helion, Gliwice 2003 9. Lipski W.: Kombinatoryka dla programistów, WNT, Warszawa 2004 10. Pomoce naukowe: Materiały dydaktyczne opracowane przez wykładowcę					

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	30 [h]
Udział w ćwiczeniach / laboratoriach / projektach / seminariach	X	X	30 [h]
Udział w konsultacjach	6 [h]	X	X
Przygotowanie do wykładów / ćwiczeń / laboratoriów / projektów / seminariów	X	59 [h]	X
Przygotowanie do zaliczenia/egzaminu			
Sumaryczne obciążenie pracą studenta	6[h]/ 0,2 ECTS	59 [h]/ 2,4ECTS	60[h]/ 2,4 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>