



Karta przedmiotu
Podstawy programowania

1. Informacje podstawowe

Kierunek studiów teleinformatyka	Cykl kształcenia (nabór) 2024/25	
Specjalność -	Kod przedmiotu 05TINN.PI1B.0533.24	
Jednostka zarządzająca kierunkiem studiów Wydział Telekomunikacji, Informatyki i Elektrotechniki	Języki wykładowe polski	
Poziom studiów pierwszego stopnia (inż.)	Obligatoryjność Obowiązkowy	
Profil studiów Profil ogólnoakademicki	Blok zajęciowy Przedmioty podstawowe	
Forma studiów studia niestacjonarne		
Wymagania wstępne	brak	
Przedmioty wprowadzające	brak	
Koordinator	Rafał Boniecki	
Okres Semestr 1	Forma i godziny zajęć • Wykład: 18, Zaliczenie na ocenę • Ćwiczenia laboratoryjne: 18, Zaliczenie na ocenę	Liczba punktów ECTS 5

2. Efekty uczenia się dla przedmiotu

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
Wiedza:			

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
W1	Ma uporządkowaną wiedzę w zakresie metodyki i techniki programowania aplikacji desktopowych, webowych i mobilnych.	TIN_O1_K_W03, TIN_O1_K_W05	P6S_WG, P6S_WG_inż, P6S_WG P6S_WG_inż
W2	Ma szczegółową wiedzę w zakresie architektury oprogramowania systemów informatycznych.	TIN_O1_K_W05, TIN_O1_K_W06	P6S_WG, P6S_WG_inż, P6S_WG P6S_WG_inż
Umiejętności:			
U1	Potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie.	TIN_O1_K_U01	P6S_UW P6S_UW_inż
U2	Potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania.	TIN_O1_K_U05, TIN_O1_K_U12	P6S_UU, P6S_UO
U3	Potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst.	TIN_O1_K_U03	P6S_UO
U4	Potrafi sformułować specyfikację prostych systemów informatycznych na poziomie klas, z wykorzystaniem języków zorientowanych obiektowo. Potrafi sformułować algorytm z wykorzystaniem klas kolekcji, posługuje się językami programowania wysokiego poziomu oraz odpowiednimi narzędziami informatycznymi do opracowania programów komputerowych	TIN_O1_K_U17	P6S_UW P6S_UW_inż
Kompetencje społeczne:			
K1	Rozumie potrzebę i zna możliwości ciągłego dokształcania się (studia drugiego i trzeciego stopnia, studia podyplomowe, kursy) - podnoszenia kompetencji zawodowych, osobistych i społecznych.	TIN_O1_K_K01	P6S_KK
K2	Ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.	TIN_O1_K_K05	P6S_KO

3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	Dane i ich komputerowe reprezentacje. Struktury danych, paradygmaty projektowania algorytmów, modele rozwiązywania problemów. Podstawowe konstrukcje języków algorytmicznych. Rekurencja i typy programów rekurencyjnych. Analiza sprawności algorytmów. Programowanie strukturalne i obiektowe. Algorytmy sortowania i wyszukiwania danych. Przetwarzanie grafów i przetwarzanie łańcuchów znakowych. Dynamiczne struktury danych – listy, tablicowe implementacje list, stos, kolejki, sterty i kolejki priorytetowe, drzewa i ich reprezentacje. Zastosowanie techniki programowania typu „dziel-i-rządź”. Programowanie interakcji z użytkownikiem. Algorytmy i struktury danych są przedstawiane w Javie, ale w stylu przystępnym dla osób znających inne współczesne języki programowania.	Wykład	W1, W2
2.	W ramach ćwiczeń laboratoryjnych studenci uczą się praktycznej implementacji w języku Java z wykorzystaniem środowiska IDE (Eclipse lub NetBeans lub IntelliJ IDEA) zagadnień omawianych na wykładzie.	Ćwiczenia laboratoryjne	U1, U2, U3, U4, K1, K2

4. Metody prowadzenia zajęć, weryfikacji efektów uczenia się i warunki zaliczenia

Forma zajęć		
Wykład	Metody prowadzenia zajęć:	
	Wykład	
	Metody (sposoby) weryfikacji:	Udział:
	Kolokwium	90%
	Test	10%
	Warunki zaliczenia przedmiotu:	
Zdobycie min 55% punktów z kolokwium ,udział w zajęciach oraz zaliczenie testu		
Ćwiczenia laboratoryjne	Metody prowadzenia zajęć:	
	Ćwiczenia laboratoryjne, Projekt	
	Metody (sposoby) weryfikacji:	Udział:
	Sprawozdanie	100%
	Warunki zaliczenia przedmiotu:	
	Udział w zajęciach oraz ocena sprawozdań z zajęć	

Efekt uczenia się dla przedmiotu	Metody (sposoby) weryfikacji		
	Test	Kolokwium	Sprawozdanie
W1		x	
W2		x	

U1			x
U2			x
U3	x		x
U4	x		x
K1			x
K2			x

5. Literatura

Literatura podstawowa

1. Sedgewick R, Gliwice 2012r. „Algorytmy”, Wydanie IV, Helion
2. Schildt H, Gliwice 2020r. „Java Kompendium programisty”, Helion
3. Aho A, Hopcroft J, Ullman J, Gliwice 2003r. „Projektowanie i analiza algorytmów”, Helion
4. Loudon K, Gliwice 2003 r. „Algorytmy w C”, Helion
5. Wróblewski Piotr, Gliwice 2015r. Algorytmy, struktury danych i techniki programowania, Wydanie V, Helion

Literatura uzupełniająca

1. Drozdek A, Warszawa 1996r. „struktury danych w języku C”, WNT
2. Sedgewick R, Warszawa 2003r. „Algorytmy w C++”, RM
3. Sedgewick R, Wayne K, Gliwice 2018 Programowanie w języku Java. Podejście interdyscyplinarne. Wydanie II, Helion
4. Koffman E, Gliwice 2006r. „Struktury danych i techniki obiektowe na przykładzie Javy 5.0”, Helion
5. Barr A, Gliwice 2006r. „Znajdź błąd. Sztuka analizowania kodu”, Helion Gliwice

6. Nakład pracy studenta - bilans godzin i punktów ECTS

Aktywność studenta		Obciążenie studenta Liczba godzin
Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innych osób prowadzących zajęcia	Wykład	18
	Ćwiczenia laboratoryjne	18
Praca własna studenta	Przygotowanie sprawozdania	35
	Przygotowanie do zajęć	20
	Studiowanie literatury	25
	Inne (przygotowanie do egzaminu)	30
	Konsultacje	3
Łączny nakład pracy studenta		149
Liczba punktów ECTS		5

* Godzina (dydaktyczna) oznacza 45 minut