

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	JĘZYKI I PARADYGMATY PROGRAMOWANIA	
IT/P/I/NST/B ₁ -5			PROGRAMMING LANGUAGES AND PARADIGMS	
Język wykładowy		polski		
Rok akademicki		2019/2020		
Kierunek		Informatyka techniczna		
w zakresie				
Poziom studiów		studia pierwszego stopnia		
Profil studiów		praktyczny		
Forma studiów		studia niestacjonarne		
Semestr / semestry		trzeci zimowy		
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	15 [h]	3 ECTS
		Ćwiczenia laboratoryjne	30 [h]	
		
Powiązanie przedmiotu	z profilem studiów	kształtuje umiejętności praktyczne		2 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		3 ECTS
	z dyscypliną	informatyka techniczna i telekomunikacja		3 ECTS
Forma nauczania		tradycyjna- zajęcia zorganizowane w Uczelni		
Wymagania wstępne		Wymagana znajomość przedmiotów: podstawy programowania, programowanie obiektowe, programowanie niskopoziomowe		
Jednostka prowadząca		Katedra Informatyki		
Koordynator		dr Artur Hermanowicz		
Osoby prowadzące		dr Artur Hermanowicz		
Adres strony internetowej pjo		www.wim.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		artur.hermanowicz@uthrad.pl		

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>Wykłady</p> <p>Programowanie proceduralne: składnia i semantyka języków programowania, metody opisu składni, typy, podprogramy (procedury i funkcje). Metody przekazywania parametrów do podprogramów: przez wartość, zmienną, referencję [2h] – W1, W2.</p> <p>Programowanie imperatywne: zmienne, struktura programu, organizacja wywołań podprogramów, przydział pamięci na stosie i na stercku [2h] – W1, W2.</p> <p>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 [7h] – W1, W2, W3.</p> <p>Programowanie funkcyjne: funkcje jako model programowania, rachunek lambda, działania na listach, dopasowywanie wzorca, nadawanie typów, rekursja. Procedury jako argumenty i wyniki [2h] – W1, W2.</p> <p>Programowanie w logice: rachunek predykatów w języku Prolog, rezolucja, listy [2h] – W1, W2.</p> <p>Ćwiczenia laboratoryjne</p> <p>Zapoznanie się ze środowiskiem programistycznym Eclipse. Tworzenie prostych aplikacji konsolowych w Javie [4h] – U1, U2, K1.</p> <p>Programowanie obiektowe w Javie. Działania na łańcuchach i tablicach. Obsługa plików tekstowych i binarnych [4h] – U1, U2, K1.</p> <p>Budowa aplikacji okienkowej w Javie, programowanie kontrolek, projektowanie interfejsu użytkownika, obsługa zdarzeń [6h] – U1, U2, K1.</p> <p>Wyświetlanie grafiki w oknie, animacja [6h] – U1, U2, K1.</p> <p>Programowanie funkcyjne w Haskellu [6h] – U1, U2, K1.</p> <p>Programowanie w logice z wykorzystaniem języka Prolog [4h] – U1, U2, K1.</p>
Metody dydaktyczne (kształcenia):	<p>Metody podające - wykład informacyjny – W1, W2, W3</p> <p>Metody praktyczne – ćwiczenia laboratoryjne - U1, U2, K1</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. 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.</p> <p>Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco:</p> <p>Ćwiczenia laboratoryjne – warunkiem zaliczenia jest osiągnięcie wszystkich wymaganych efektów kształcenia dla tej formy zajęć i uzyskanie pozytywnych ocen za pomocą przyjętych dla przedmiotu metod oceniania.</p> <p>Ocena końcowa z ćwiczeń laboratoryjnych stanowi sumę ocen: 90 % sprawdziany praktyczne przy komputerze, 10% aktywność na zajęciach.</p> <p>Wykład – 100% ocena z egzaminu.</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	Posiada wiedzę z podstaw informatyki w zakresie języków i paradygmatów programowania. Zna najważniejsze paradygmaty występujące we współczesnym programowaniu: imperatywne, proceduralne, obiektowe, funkcyjne i logiczne.	K_WG05	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
W2	Ma uporządkowaną wiedzę w zakresie algorytmów i struktur danych w odniesieniu do różnych paradygmatów i języków programowania.	K_WG05	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
W3	Ma uporządkowaną, podbudowaną teoretycznie wiedzę z zakresu baz danych, w aspekcie ich programowania z wykorzystaniem różnych języków i paradygmatów.	K_WG10	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U1	Potrafi sformułować algorytm oraz go zaimplementować w wybranym języku programowania wykorzystując odpowiedni paradygmat programowania.	K_UW05	ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U2	Potrafi ocenić, na podstawowym poziomie, przydatność stosowanych w programowaniu języków i paradygmatów i wybrać odpowiedni do zaprogramowania jednego z typowych algorytmów.	K_UW16	ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
K1	Ma świadomość ciągłego rozwoju technik i metod programowania, istnienia i powstawania wielu różnych języków i paradygmatów programowania oraz wynikającą z tego konieczność stałego aktualizowania i poszerzania swojej wiedzy.	K_KK01	ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	aktywność na zajęciach
Stopień osiągnięcia kierunkowych efektów uczenia się: K_WG05+++, K_WG10+, K_UW05++, K_UW16+, K_KK01+					

Literatura podstawowa, literatura uzupełniająca, pomoce naukowe	
Literatura podstawowa:	
1. Abelson H., Sussman G.J., Sussman J.: <i>Struktura i interpretacja programów komputerowych</i> , WNT, Warszawa 2002.	
2. Eckel B.: <i>Thinking in Java</i> . Edycja polska, wyd. 4, Helion, Gliwice 2006.	
3. Troelsen A.: <i>Język C# i platforma.NET</i> , PWN, Warszawa 2006.	
4. Wierzbicki M.: <i>Java. Programowanie obiektowe</i> , Helion, Gliwice 2006.	
Literatura uzupełniająca:	
1. Ben-Ari M.: <i>Logika matematyczna w informatyce</i> , WNT, Warszawa 2006.	
2. Clocksin W.F., Mellish C.S.: <i>Prolog. Programowanie</i> , Helion, Gliwice 2002	
3. Horstman C.S., Cornell G.: <i>Java, Podstawy (tom 1), Techniki zaawansowane (tom 2)</i> , wyd. 8, Helion, Gliwice 2008.	
4. Jakubczyk K.: <i>Turbo Pascal i Borland C++</i> . Przykłady, Helion, Gliwice 2002, 2006	
5. Van Roy P., Haridi S.: <i>Programowanie. Koncepcje, techniki i modele</i> , Helion, Gliwice 2005.	
6. Wirth N.: <i>Wstęp do programowania systematycznego</i> , WNT, Warszawa 1989.	

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	10 [h]	X
Udział w ćwiczeniach laboratoryjnych	X	X	30 [h]
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	X	15 [h]	X
Udział w konsultacjach	4 [h]	X	X
Przygotowanie do zaliczenia / egzaminu	X	X	X
Udział w egzaminie / zaliczeniu	1 [h]	X	X
Sumaryczne obciążenie pracą studenta	5 [h]/ 0,2 ECTS	25 [h]/1,0 ECTS	45[h]/ 1,8 ECTS
Punkty ECTS za przedmiot	3 ECTS		

Informacje dodatkowe, uwagi

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.

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.

Terminy odbywania zajęć: zgodnie z planem zajęć.

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