

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Języki i paradygmaty programowania		Kod 1010334541010334960
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) (brak)	Rok / Semestr 2 / 4
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 20 Ćwiczenia: - Laboratoria: 20 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (brak)		(ogólnouczelniany, z innego kierunku) (brak)
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Grażyna Brzykcy email: grazyna.brzykcy@put.poznan.pl tel. 616653724 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student ma podstawową wiedzę w zakresie matematyki, obejmującą algebrę, analizę i logikę oraz wiedzę w zakresie podstawowych konstrukcji programistycznych, implementacji algorytmów, języków formalnych i platform programowania.
2	Umiejętności:	Student potrafi konstruować algorytmy z wykorzystaniem podstawowych technik algorytmicznych, dokonać analizy ich złożoności i potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów.
3	Kompetencje społeczne	Student ma świadomość ważności dokładnego definiowania programu i zachowania standardów notacyjnych.
Cel przedmiotu: Zapoznanie studentów z deklaratywnymi stylami programowania i zasadami doboru stylu i języka programowania do charakteru zadania. Opanowanie przez studentów umiejętności posługiwania się konstrukcjami języków deklaratywnych w środowiskach programowania funkcyjnego i programowania w logice. Zwrócenie uwagi na związek pomiędzy rekurencyjnymi strukturami danych, a programami rekurencyjnymi.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie technik projektowania i wykorzystywania rekurencyjnych struktur danych i ich implementacji. - [K_W04]		
2. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę o modelach obliczeniowych i podstawowych deklaratywnych konstrukcjach programistycznych. - [K_W05]		
3. Student orientuje się w obecnym stanie oraz trendach rozwojowych informatyki. - [K_W19]		
Umiejętności:		
1. Student potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i w deklaratywny sposób opisać wyniki tego zadania. - [K_U03]		
2. Student potrafi konstruować algorytmy z wykorzystaniem technik programowania w logice i programowania funkcyjnego. - [K_U09]		
3. Student potrafi posłużyć się środowiskami i platformami programowania deklaratywnego do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania deklaratywnego. - [K_U10]		
Kompetencje społeczne:		
1. Student ma świadomość ważności i rozumie skutki działalności inżyniera-informatyka i związaną z tym odpowiedzialność za podejmowane decyzje. - [K_K02]		
2. Student zna wagę zasad starannego definiowania programów, przestrzegania poprawności językowej i terminowego oddania prac. - [K_K07]		

Sposoby sprawdzenia efektów kształcenia		
<p>Wykład Zaliczenie z oceną, która jest ustalana na podstawie wyniku testu pisemnego (sprawdzenie znajomości podstawowych pojęć i konstrukcji stosowanych w programowaniu deklaratywnym).</p> <p>Laboratorium Ocena z laboratorium odpowiada liczbie uzyskanych punktów a wejściówki, sprawdzian (samodzielne definiowanie programów) i aktywność programistyczną na zajęciach, przy czym ocenę 3,0 można uzyskać po zgromadzeniu co najmniej 50,1% maksymalnej liczby punktów.</p>		
Treści programowe		
<p>Wykład Logika jako język programowania (proceduralne ujęcie zasady rezolucji). Struktury danych, procedury i charakterystyczne konstrukcje Prologu. Programowanie funkcyjne: typy danych, sposoby definiowania funkcji, przykładowe języki i środowiska wykonawcze. Kierunki rozwoju programowania deklaratywnego. Podstawy wybranych nieklasycznych technik programowania, takich jak przetwarzanie ewolucyjne, programowanie ograniczeń, sztuczne sieci neuronowe.</p> <p>Metody kształcenia Na wykładzie wykorzystuje się prezentację multimedialną uzupełnioną przykładami prezentowanymi na tablicy, sformułowane są pytania do studentów i analizowane przedstawiane odpowiedzi.</p> <p>Aktualizacja 2017: - Erlang jako język programowania funkcyjnego, - sztuczne sieci neuronowe jako paradygmat programowania.</p> <p>Laboratorium Projektowanie algorytmów i ich implementacja w języku programowania w logice (Prolog) i w języku funkcyjnym (Erlang).</p> <p>Metody kształcenia: - studenci wykorzystują narzędzia umożliwiające samodzielną pracę w domu, - prezentowane są i analizowane różne rozwiązania zadań przygotowywane przez studentów, - prowadzone są eksperymenty obliczeniowe.</p> <p>Aktualizacja 2017: - nowe środowisko programowania funkcyjnego z językiem Erlang.</p>		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Bartoszek J., Cybulka J.: Programowanie deklaratywne. Wyd. PP, 1999. 2. Brzykcy G., Meissner. A.: Programowanie w Prologu i programowanie funkcyjne. Materiały do ćwiczeń, Wyd. PP, 1999. 3. Clocksin W. F., Mellish C. S.: Prolog. Programowanie, Wyd. Helion, 2003. 4. Haber F.: Learn you some ERLANG for great good! A beginner's guide (on-line learnyousomeerlang.com), 2017. 5. Van Roy P., Haridi S.: Programowanie. Konceptcje, techniki i modele, Helion, Gliwice 2005. 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Arabas J.: Wykłady z algorytmów ewolucyjnych, WNT Warszawa, 2001. 2. Armstrong J.: Programming Erlang. The Pragmatic Programmers, 2013. 3. Kowalski R.: Logika w rozwiązywaniu zadań, WNT, Warszawa 1989. 4. Nilsen U., Małuszyński J.: Logic, Programming and PROLOG, John Wiley & Sons, 2000. 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. Wykłady	20	
2. Laboratoria	20	
3. Przygotowanie do zajęć	60	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4

Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	40	2
Zajęcia o charakterze praktycznym	80	3