

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Języki i paradygmaty programowania</b>		Kod <b>1010331541010334960</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>2 / 4</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: <b>-</b> Laboratoria: <b>30</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>(brak)</b>		(ogólnouczelniany, z innego kierunku) <b>(brak)</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>  <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>4 100%</b>  <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>  dr inż. Grażyna Brzykcy email: grazyna.brzykcy@put.poznan.pl tel. 616653714 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	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	<b>Umiejętności:</b>	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	<b>Kompetencje społeczne</b>	Student ma świadomość ważności dokładnego definiowania programu i zachowania standardów notacyjnych.
<b>Cel przedmiotu:</b> 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.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
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]		
<b>Umiejętności:</b>		
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]		
<b>Kompetencje społeczne:</b>		
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]		

<b>Sposoby sprawdzenia efektów kształcenia</b>		
<p>Wykład - zaliczenie na ocenę odbywa się na podstawie test pisemnego (sprawdzenie znajomości podstawowych pojęć i konstrukcji stosowanych w programowaniu deklaratywnym) i/lub przygotowania przez studenta opracowania na temat wskazanego środowiska do definiowania sztucznych sieci neuronowych.</p> <p>Laboratorium - zaliczenie na ocenę na podstawie uzyskanych punktów (ocena 3,0 za co najmniej 50,1% maksymalnej liczby punktów) za wejściówki, 2 sprawdziany (samodzielne definiowanie programów) i aktywność programistyczną na zajęciach.</p>		
<b>Treści programowe</b>		
<p><b>Wykład</b></p> <p>Logika jako język programowania (proceduralne ujęcie zasady rezolucji). Struktury danych, procedury i charakterystyczne konstrukcje Prologu. Rekurencyjne struktury danych i rekurencyjne programy prologowe. Programowanie funkcyjne: typy danych, sposoby definiowania funkcji, przykładowe języki i środowiska wykonawcze. Kierunki rozwoju programowania deklaratywnego. Podstawy wybranych nieklasycznych technik programowania: przetwarzanie ewolucyjne, programowanie ograniczeń, sztuczne sieci neuronowe.</p> <p>Aktualizacja 2017: zmiana języka funkcyjnego na Erlang.</p> <p><b>Laboratorium</b></p> <p>Projektowanie algorytmów i ich implementacja w języku programowania w logice (Prolog) i w języku funkcyjnym (Erlang).</p> <p>Aktualizacja 2017: nowe środowisko programowania funkcyjnego (Erlang).</p> <p>Zastosowane metody kształcenia:</p> <ul style="list-style-type: none"> <li>- wykład: wykorzystywane są prezentacje multimedialne uzupełniane przykładami na tablicy oraz prezentacje studentów; formułowane pytania do studentów i inicjowana dyskusja,</li> <li>- laboratorium: studenci korzystają z narzędzi umożliwiających wykonywanie zadań w domu (oprogramowanie open-source), demonstrowane są i analizowane różne rozwiązania zadań, prowadzone są eksperymenty obliczeniowe.</li> </ul>		
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. Bartoszek J., Cybulka J.: Programowanie deklaratywne. Wyd. PP, 1999.</li> <li>2. Brzykcy G., Meissner. A.: Programowanie w Prologu i programowanie funkcyjne. Materiały do ćwiczeń, Wyd. PP, 1999.</li> <li>3. Clocksin W. F., Mellish C. S.: Prolog. Programowanie, Wyd. Helion, 2003/2005.</li> <li>4. Haber F.: Learn you some ERLANG for great good! A beginner's guide (on-line learnyousomeerlang.com), 2017.</li> <li>5. Van Roy P., Haridi S.: Programowanie. Koncepcje, techniki i modele, Helion, Gliwice 2005/2010.</li> </ol>		
<p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. Arabas J.: Wykłady z algorytmów ewolucyjnych, WNT Warszawa, 2001.</li> <li>2. Armstrong J.: Programming Erlang. The Pragmatic Programmers, 2013.</li> <li>3. Kowalski R.: Logika w rozwiązywaniu zadań, WNT, Warszawa 1989.</li> <li>4. Nilsen U., Małuszyński J.: Logic, Programming and PROLOG, John Wiley &amp; Sons, 2000.</li> </ol>		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
Czynność	Czas (godz.)	
1. Wykłady	30	
2. Laboratoria	30	
3. Przygotowanie do zajęć	40	
<b>Obciążenie pracą studenta</b>		
forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2
Zajęcia o charakterze praktycznym	70	3