



Logic Programming

Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka i Systemy Inteligentne	Cykl dydaktyczny 2022/2023	
Specjalność -	Kod przedmiotu EISIS.li500.624614f2f0150.22	
Jednostka organizacyjna Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej	Języki wykładowe angielski	
Poziom kształcenia Studia inżynierskie I stopnia	Obligatoryjność Do wyboru	
Forma studiów Stacjonarne	Blok zajęciowy Przedmioty ogólne	
Profil studiów Ogólnoakademicki	Przedmiot powiązany z badaniami naukowymi Tak	
Koordynator przedmiotu	Weronika T. Adrian, Antoni Ligęza	
Prowadzący zajęcia	Weronika T. Adrian, Antoni Ligęza, Mateusz Ślażyński	
Okresy Semestr 5, Semestr 7	Forma zaliczenia Zaliczenie	Liczba punktów ECTS 3
	Forma prowadzenia i godziny zajęć Wykład: 14 Ćwiczenia laboratoryjne: 14	

Cele kształcenia dla przedmiotu

C1	Aby studenci znali podstawy programowania w Prologu i umieli tę umiejętność wykorzystać w praktyce. I aby docenili piękno tego języka!
C2	Chcę pokazać Studentom, że poza programowaniem proceduralnym i obiektowym jest jeszcze inny, piękny Świat; to Świat programowania deklaratywnego. Wejście do tego Świata to trochę jak przejście Alicji na "drugą stronę lustra". Na początku trudne; jednak wytrwali odkryją jak pięknym i prostym językiem programowania jest Prolog. Celem kształcenia jest zatem aby Studenci rozumieli podstawy programowania deklaratywnego i celowość jego stosowania - na choćby elementarnym poziomie... Ale z pełnym zrozumieniem. A także by rozumieli rolę, zalety i ograniczenia programowania deklaratywnego w informatyce. I docenili jego piękno - kreowane poprzez prostotę konstrukcji języka - a jednocześnie jego siłę wyrazu. Jako jedyny chyba język programowania, Prolog - a dokładniej program w Prologu - daje się zdefiniować w trzech słowach: "ciąg klauzul Horna". Celem kształcenia jest objaśnienie znaczenia kombinacji tych trzech prostych słów... I próba pokazania, że pomimo tak prostej struktury programowanie w Prologu jest poezją współczesnej informatyki. I - na koniec - dwa cytaty ze zbioru moich "Golden Ideas": "W żadnym innym języku programowania myślenie nie wyprzedza tak kładzenia kodu jak w Prologu." "Prolog to jednak język dla Elit (do studentów, którzy na wykład przychodzą nielicznie)." Tak - Prolog - programowanie deklaratywne - jest dla elit; ale czyż aktualni Studenci - a przyszli absolwenci Wydziału IEiT AGH nie aspirują do tego grona?

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	ma podstawową wiedzę na temat różnych paradygmatów programowania w tym programowania deklaratywnego, w tym zwłaszcza programowania logicznego, oraz opartego na koncepcji relacji i proceduralnego; wie jak dobierać paradygmat do rozwiązywania konkretnych problemów decyzyjnych i obliczeniowych w różnych dziedzinach informatyki	ISI1A_W05, ISI1A_W06	Prezentacja
W2	ma podstawową wiedzę na temat logiki i jej zastosowań w programowaniu; zna składnię i semantykę logiki rachunku zdań i rachunku predykatów	ISI1A_W05, ISI1A_W06	Prezentacja
W3	ma podstawową wiedzę na temat paradygmatu programowania deklaratywnego w logice; zna podstawy opisu problemów i programowania w języku Prolog	ISI1A_W05, ISI1A_W06	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Prezentacja, Zaliczenie laboratorium
Umiejętności - Student potrafi:			
U1	potrafi ocenić przydatność różnych paradygmatów programowania i związanych z nimi środowisk programistycznych	ISI1A_U03, ISI1A_U06, ISI1A_U07	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Prezentacja, Zaliczenie laboratorium
U2	potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować programy zapisane w języku programowania deklaratywnego; potrafi wykorzystać podstawowe pojęcia i konstrukcje języka programowania w logice Prolog do zapisu programów	ISI1A_U06, ISI1A_U07	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Zaliczenie laboratorium

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Przedmiotem kursu jest programowanie w języku Prolog. Prolog jest deklaratywnym językiem programowania logicznego. Kurs obejmuje zarówno podstawy teoretyczne jak i przykłady oraz ćwiczenia praktyczne.

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	14
Ćwiczenia laboratoryjne	14
Samodzielne studiowanie tematyki zajęć	62
Łączny nakład pracy studenta	Liczba godzin 90
Liczba godzin kontaktowych	Liczba godzin 28

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	Wprowadzenie do programowania deklaratywnego i programowania logicznego; elementarny przegląd paradygmatów programowania. Sztuczna Inteligencja, Inżynieria Wiedzy i rola języka Prolog w tych obszarach.	W1, U1	Wykład
2.	Podstawy formalne programowania w logice: elementy składni, unifikacja, rezolucja. Strategia Backtracking Depth-First Search.	W1, W2, W3	Wykład, Ćwiczenia laboratoryjne
3.	Programowanie w języku Prolog.	W3, U1, U2	Wykład, Ćwiczenia laboratoryjne
4.	Wybrane zastosowania i kierunki rozwojowe programowania w logice.	W1, U1	Wykład

Informacje rozszerzone

Metody i techniki kształcenia:

Mini wykład, Kształcenie zdalne, Praca grupowa

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Wykład	Prezentacja, Zaliczenie laboratorium	Zgodnie z regulaminem studiów
Ćwiczenia laboratoryjne	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego	Zgodnie z regulaminem studiów

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Aby otrzymać zaliczenie, konieczne jest uzyskanie 50% z poniższych komponentów:

— zadania laboratoryjne; — kolokwium.

W przypadku braku zaliczenia w pierwszym terminie, termin poprawkowy pozwoli na kolejne podejście do powyższych zadań.

Sposób obliczania oceny końcowej

Ocena końcowa wyliczana będzie na podstawie zajęć laboratoryjnych i będzie liczona jako średnia ważona komponentów:

- zadania laboratoryjne: 50% oceny. - kolokwium: 50% oceny.

Aby otrzymać zaliczenie, konieczne jest uzyskanie 50% punktów za każdy komponent z osobna. Dodatkowo, ocena może zostać podwyższona ze względu na aktywność na zajęciach lub wykładzie.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

W przypadku nieobecności, należy skontaktować się z prowadzącym, aby otrzymać dodatkowe zadania.

Wymagania wstępne i dodatkowe

Znajomość podstawowych zagadnień z zakresu logiki formalnej.

Student będzie korzystał z systemu kontroli wersji git.

Generalnie:

Znajomość matematyki w zakresie wymaganym dla studentów studiów informatycznych.

Entuzjazm i zapał do STUDIOWANIA - w tym samodzielnego poszerzania i pogłębiania wiedzy.

Uważność na zajęciach, zdolność do samodzielnego myślenia i wytężonej pracy. Otwartość, kreatywność i zdolność do myślenia niestandardowego.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Prowadzenie notatek z wykładu jest usilnie rekomendowane.

Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne: Obecność jest obowiązkowa. Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionych problemów.

Zaliczenie modułu jest możliwe po zaliczeniu: wszystkich zajęć laboratoryjnych oraz kolokwium.

Literatura

Obowiązkowa

1. Dave Stuart Robertson, Quick Prolog (<http://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/quickprolog.html>)
2. GEIST Lab z Prologu, http://ai.ia.agh.edu.pl/wiki/pl:prolog:prolog_lab
3. Jan Wielemaker et. al., SWI-Prolog
4. Patrick Blackburn, Johan Bos, Kristina Striegnitz, Learn Prolog Now! (<http://www.learnprolognow.org/>)
5. I. Bratko, Prolog Programming for Artificial Intelligence, 4th ed, Addison-Wesley, 2011.
6. Michael Covington et. al., Prolog Programming in Depth.
7. Dennis Merritt, Adventure In Prolog, Amzi!, 2010.
8. Dennis Merritt, Building Expert Systems in Prolog, Amzi!, 2010.
9. Ulf Nilsson and Jan Małuszyński, Logic, Programming and Prolog (2ed)., Wiley, 2006
10. Michael Covington et. al., Prolog Programming in Depth., 1995.
11. Attila Csenki: Applications of Prolog. 2009 Attila Csenki & Ventus Publishing ApS

Dodatkowa

1. MICHAEL A. COVINGTON, ROBERTO BAGNARA, RICHARD A. O'KEEFE, JAN WIELEMAKER, SIMON PRICE, Coding guidelines for Prolog, Cambridge University Press, 2011.
2. Implementacja Prologu i dokumentacja: <http://www.swi-prolog.org/>
3. Materiały do Prologu: <http://ai.ia.agh.edu.pl/wiki/pl:dydaktyka:pp:start> <http://home.agh.edu.pl/~ligeza/wiki/prolog:course>

Badania i publikacje

Badania

1. Knowledge Representation and Reasoning, Artificial Intelligence, Constraint Programming

Publikacje

1. Antoni Ligeża: Logical Foundations for Rule-Based Systems. Springer, 2006.
2. Improving efficiency in constraint logic programming through constraint modeling with rules and hypergraphs / Antoni LIGĘZA // W: FedCSIS [Dokument elektroniczny] : proceedings of the Federated Conference on Computer Science and Information Systems 2012 : September 9-12, 2012 Wrocław, Poland / eds. M. Ganzha, L. Maciaszek, M. Paprzycki. — Dane tekstowe. — Warsaw : Polskie Towarzystwo Informatyczne ; Los Alamitos : IEEE Computer Society Press, 2012. — Dane na dysku Flash. — W bazie Web of Science ISBN 978-83-60810-48-4. — ISBN: 978-83-60810-51-4. — S. 101-107.
3. A study of methodological issues in design and development of rule-based systems: proposal of a new approach / Antoni LIGĘZA, Grzegorz J. NALEPA // Data Mining and Knowledge Discovery ; ISSN 1384-5810. — 2011 vol. 1 iss. 2, s. 117-137. — Bibliogr. s. 135-137
4. Models and tools for improving efficiency in constraint logic programming / Antoni LIGĘZA // Decision Making in Manufacturing and Services ; ISSN 1896-8325. — 2011 vol. 5 no. 1-2, s. 69-78. — Bibliogr. s. 78, Abstr.. — tekst: http://journals.bg.agh.edu.pl/DECISION/2011-01-02/DM_2011_1_2_06.pdf
5. The HeKatE methodology : hybrid engineering of intelligent systems / Grzegorz J. NALEPA, Antoni LIGĘZA // International Journal of Applied Mathematics and Computer Science ; ISSN 1641-876X. — 2010 vol. 20 no. 1: Computational intelligence in modern control systems, s. 35-53. — Bibliogr. s. 51-53

Kierunkowe efekty uczenia się

Kod	Treść
ISI1A_U03	Potrafi korzystać z literatury fachowej, zdobywać potrzebne informacje w sieci Internet, dokonywać interpretacji zdobytych informacji oraz wyciągać wnioski i formułować opinie.
ISI1A_U06	Potrafi algorytmizować wybrane problemy, ocenić ich złożoność obliczeniową, estymować czas wykonania, dobrać właściwe algorytmy do zadanego problemu, stosować metody i techniki Sztucznej Inteligencji.
ISI1A_U07	Potrafi projektować i rozwijać aplikacje z wykorzystaniem poznanych technologii oraz języków programowania. Potrafi doskonalić umiejętności nabyte w trakcie studiów.
ISI1A_W05	Ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie języków formalnych, kompilatorów oraz języków programowania.
ISI1A_W06	Ma uporządkowaną, podbudowaną teoretycznie wiedzę z inżynierii oprogramowania, modelowania oprogramowania, zarządzania projektem informatycznym, wdrażania i komercjalizacja rozwiązań informatycznych.