



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie deklaratywne [N1Inf1>PDEK]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
2/3

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
niestacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
12

Laboratorium
12

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

2,00

Koordynatorzy

dr inż. Grażyna Brzykcy
grazyna.brzykcy@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student powinien dysponować podstawową wiedzą z zakresu logiki obliczeniowej oraz teorii mnogości. Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych opisanych za pomocą formalnego aparatu matematycznego oraz umiejętność pozyskiwania informacji na temat sposobów ich rozwiązywania ze wskazanych źródeł.

Cel przedmiotu

Przekazanie studentom podstawowej wiedzy dotyczącej deklaratywnego paradygmatu programowania na przykładzie języka Prolog. Rozwijanie u studentów umiejętności rozwiązywania problemów algorytmicznych z wykorzystaniem języka programowania wysokiego poziomu opartego na mechanizmach rekurencji oraz przeszukiwaniu przestrzeni rozwiązań. Kształtowanie u studentów świadomości różnic między paradygmatem deklaratywnym a pozostałymi paradygmatami programowania.

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie języków i paradygmatów programowania oraz interpretacji proceduralnej i deklaratywnej kodu programu.[K1st_W4]

2. Posiada wiedzę podstawową nt. funkcjonowania automatycznych mechanizmów sterowania wnioskowaniem w językach deklaratywnych opartych na logice obliczeniowej.[K1st_W6]
3. Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu zadań informatycznych z zakresu przetwarzania rekurencyjnych struktur danych i ich implementacji w deklaratywnych językach programowania.[K1st_W7]
4. Dysponuje wiedzą elementarną w zakresie form i metod metaprogramowania, przetwarzania symbolicznych form reprezentacji danych i wiedzy.[K1st_W7]

Umiejętności:

1. Potrafi ocenić przydatność języków, metod i narzędzi służących do rozwiązywania zadań typowych dla informatyki, oraz wskazywać właściwe obszary zastosowań metod i narzędzi programistycznych paradygmatu deklaratywnego.[K1st_U10]
2. Posiada umiejętność formułowania algorytmów i ich programowania w zakresie zadań o charakterze symbolicznym i tekstowym.[K1st_U11]
3. Umie opracować i zaimplementować rozwiązanie problemu programistycznego w kategoriach paradygmatu deklaratywnego z zastosowaniem prostych i złożonych (rekurencyjnych) struktur danych. [K1st_U11]

Kompetencje społeczne:

1. Rozumie potrzebę ciągłego pogłębiania swojej wiedzy i doskonalenia umiejętności w zakresie narzędzi programistycznych i rozwijających się paradygmatów programowania.[K1st_K1]
2. Potrafi myśleć i działać w sposób kreatywny, pozostając otwartym na pozainformatyczne aspekty działalności inżyniera-informatyka związane z konstruowaniem oprogramowania.[K1st_K3]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Za pomocą oceny wiedzy nabytej w ramach wykładu - w formie testu (wielokrotny wybór, pytania otwarte) z punktowanymi pytaniami o różnej skali trudności. Połowa maksymalnej liczby punktów wymagana jest do otrzymania oceny pozytywnej.

Za pomocą oceny wiedzy i umiejętności nabytych w ramach ćwiczeń laboratoryjnych - w formie sprawdzianu z czterema zadaniami, każde za 5 punktów. Połowa łącznej maksymalnej liczby punktów wymagana jest do otrzymania oceny pozytywnej.

Treści programowe

Cechy charakterystyczne programowania deklaratywnego, w szczególności programowania w Prologu. Sposób reprezentowania wiedzy i prowadzenia obliczeń. Zmienne deklaratywne i klauzule Horna. Prezentacja najważniejszych struktur danych i zasad definiowania rekurencyjnego. Teoria logiczna, jako baza wiedzy, i automaty do dowodzenia twierdzeń, jako narzędzia obliczeniowe. Predykaty do sterowania procesem obliczeniowym. Metaprogramowanie i semantyka, deklaratywna i proceduralna, programów prologowych.

Tematyka zajęć

Wykład 1: Wprowadzenie do języka Prolog (przykład prostego programu w języku Prolog, rozszerzanie programu przez wprowadzanie reguł prologowych, rekurencyjna definicja reguły prologowej, zasady generowania odpowiedzi na postawione pytania)

Wykład 2: Budowa składniowa i interpretacja programów prologowych (reprezentacja danych w Prologu, mechanizm uzgadniania termów, deklaratywna i proceduralna interpretacja programu prologowego)

Wykład 3: Listy, operatory i operacje arytmetyczne (reprezentacja list w Prologu, wybrane operacje na listach w Prologu, notacja operatorów, operacje arytmetyczne w Prologu)

Wykład 4: Sterowanie mechanizmem nawrotów (mechanizm odcięć (ang. cuts), przykłady wykorzystywania odcięć w programie prologowym, negacja przez niepowodzenie, problemy związane z zastosowaniem odcięć i negacji w Prologu)

Wykład 5: Predefiniowane (systemowe) predykaty prologowe -metapredykaty (sprawdzanie typu termów, kompozycja i dekompozycja termów: =., functor, arg, name, różne rodzaje operacji równości w Prologu, manipulacja bazą danych w Prologu, tworzenie zbiorów rozwiązań)

Wykład6: Styl i technika programowania w Prologu (ogólne zasady poprawnego programowania w Prologu).

Laboratorium polega na projektowaniu algorytmów i ich implementacji w Prologu.

1. Wprowadzenie do Prologu (postać programu, wykonywanie obliczeń, podstawowe predykaty wbudowane, samodzielne definiowanie programu do znajdowania relacji między osobami).
2. Definiowanie rekurencyjne i iteracyjne (technika akumulatorów).
3. Działania na listach (procedury `member/2` i `append/3`, dołączanie usuwanie elementów z listy).
4. Metapredykaty sterujące (odcięcie, programowe wymuszanie nawrotów, negacja, predykaty `we/wy`, znajdowanie zbiorów rozwiązań).
5. Działania na termach (rozpoznawanie termów, definiowanie akceptora języka regularnego).
6. Sprawdzian.

Metody dydaktyczne

Wykład: prezentacja multimedialna, przykłady rozwiązań.

Ćwiczenia laboratoryjne: praktyczna realizacja zadań programistycznych o rosnącym stopniu trudności.

Literatura

Podstawowa

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.

Uzupełniająca

1. Kowalski R.: Logika w rozwiązywaniu zadań, WNT, Warszawa 1989.
2. Nilsen U., Małuszyński J.: Logic, Programming and PROLOG, John Wiley & Sons, 2000
3. Van Roy P., Haridi S.: Programowanie. Koncepcje, techniki i modele, Helion, Gliwice 2005.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	24	1,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	26	1,00