KARTA OPISU MODUŁU KSZTAŁCENIA

Nazwa modułu/przedmiotu	Kod
Programowanie deklaratywne	**1010514331010500085**

Kierunek studiów	Profil kształcenia (ogólnoakademicki, praktyczny)	Rok / Semestr
Informatyka	**ogólnoakademicki**	**2 / 3**
Ścieżka obieralności/specjalność	Przedmiot oferowany w języku:	Kurs (obligatoryjny/obieralny)
-	**polski**	**obligatoryjny**

Stopień studiów:	**Forma studiów** (stacjonarna/niestacjonarna)
I stopień	**niestacjonarna**

Godziny	Liczba punktów
Wykłady: **12** Ćwiczenia: **-** Laboratoria: **12** Projekty/seminaria: **-**	**3**

Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny)	(ogólnouczelniany, z innego kierunku)
kierunkowy	**z danego kierunku**

Obszar(y) kształcenia i dziedzina(y) nauki i sztuki	**Podział ECTS (liczba i %)**
nauki techniczne	**3 100%**
nauki techniczne	**3 100%**

Odpowiedzialny za przedmiot / wykładowca:

Dr inż. A. Michalski
email: Artur.Michalski@cs.put.poznan.pl
tel. tel. (0-61) 665-2923, fax: (0-61) 877 1525
Informatyki
ul. Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot / wykładowca:

dr inż. Agnieszka Ławrynowicz
email: agnieszka.lawrynowicz@cs.put.poznan.pl
tel. (+48) 61 6653026
Informatyki
ul. Piotrowo 2, 60-965 Poznań

Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:

1	**Wiedza:**	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z logiki matematycznej (obliczeniowej) oraz teorii mnogości.
2	**Umiejętności:**	Powinien posiadać umiejętność rozwiązywania podstawowych problemów opisanych za pomocą formalnego aparatu matematycznego oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	**Kompetencje społeczne**	Powinien również rozumieć konieczność poszerzania swoich kompetencji i pogłębiania swoich kwalifikacji w zakresie metod i narzędzi programowania. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu:

1. Przekazanie studentom podstawowej wiedzy dotyczącej deklaratywnego paradygmatu programowania (na przykładzie języka Prolog)
2. Rozwijanie u studentów umiejętności rozwiązywania prostych problemów z wykorzystaniem języka programowania wysokiego poziomu opartego na logice obliczeniowej
3. Kształtowanie u studentów świadomości różnic między paradygmatem deklaratywnym a pozostałymi paradygmatami programowania

Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia

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. 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]

Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez kolokwium na zakończenie semestru, polegające na rozwiązaniu określonego problemu programistycznego; zadanie będące przedmiotem kolokwium ma charakter wieloetapowego algorytmu, a uzyskanie oceny pozytywnej możliwe jest po poprawnym zaimplementowaniu ponad połowy wszystkich jego kroków, składających się na kompletne rozwiązanie końcowe oraz wykazaniu zdolności do konstruowania własnych (oryginalnych) składowych rozwiązania; część kroków poprawnego rozwiązania obejmuje bowiem także metody prezentowane w trakcie semestru, jednak student musi wykazać się przede wszystkim samodzielnością w rozwiązywaniu zadania, aby otrzymać zaliczenie końcowe; kryterium to jest najistotniejsze przy ocenie pracy
- ocenę wiedzy teoretycznej wykazanej na zaliczeniu pisemnym z wykładu (student nie może korzystać materiałów dydaktycznych) w formie testu wielokrotnego wyboru, składającego się z 20 pytań o łącznej wartości 20 punktów, z których połowa wymagana jest do otrzymania oceny pozytywnej (skala ocen: 2.0: 0-10pkt., 3.0: 11-12pkt., 3.5: 13-14pkt., 4.0: 15-16pkt., 4.5: 17-18pkt., 5.0: 19-20pkt.)

Treści programowe

Wykład 1: Wprowadzenie do języka Prolog
- Przykład prostego programu w języku Prolog: Relacje pokrewieństwa
- Rozszerzanie programu przez wprowadzanie reguł prologowych
- Rekurencyjna definicja reguły prologowej
- Zasady generowania odpowiedzi na postawione pytania
- Deklaratywna i proceduralna interpretacja programu prologowego

Wykład 2: Budowa składniowa i interpretacja programów prologowych
- Reprezentacja danych w Prologu
- Mechanizm uzgadniania termów
- Formalna interpretacja deklaratywna programu prologowego
- Formalna interpretacja proceduralna programu prologowego
- Przykład interpretacji programu: Problem małpy i banana
- Porządek kauzul prologowych i celów
- Związki języka Prolog z logiką formalną
- Podsumowanie

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
- Manipulowanie przepływem sterowania w Prologu
- Predykaty: bagof, setof i findall

Wykład 6: Operacje wejścia/wyjścia w Prologu
- Operacje na plikach sekwencyjnych
- Przetwarzanie plików termów
- Manipulowanie danymi znakowymi
- Kompozycja i dekompozycja atomów
- Wczytywanie programów prologowych: consult i reconsult

Wykład 7: Styl i technika programowania w Prologu
- Ogólne zasady poprawnego programowania w Prologu
- Jak interpretować programy prologowe?
- Styl programowania
- Efektywność programów prologowych

Program ćwiczeń laboratoryjnych odpowiada tematycznie programowi wykładu z wyjątkiem operacji wejścia/wyjścia.
Część wyżej wymienionych treści programowych jest realizowana w ramach pracy własnej studenta.
Metody dydaktyczne:
1. wykład: prezentacja multimedialna w tym także przykłady zadań
2. ćwiczenia laboratoryjne: realizacja praktycznych zadań programistycznych o rosnącym stopniu trudności

Literatura podstawowa:

1. Prolog. Programowanie, W.F. Clocksin, C.S. Mellish, Helion, Gliwice, 2003
2. Logika w rozwiązywaniu zadań, R.A. Kowalski, WNT, Warszawa, 1989

Literatura uzupełniająca:

1. Prolog - programming for AI, I. Bratko, Addison-Wesley, 1990
2. Micro-Prolog, K.L. Clark, F.G. McGabe, WNT, Warszawa, 1985
3. Prolog, F. Kluźniak, S. Szpakowicz, WNT, Warszawa, 1983

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych	12
2. przygotowanie do ćwiczeń laboratoryjnych	10
3. przygotowanie do kolokwium	18
4. udział w wykładach	12
5. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym (16 godz. + 2 godz.)	18
6. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	2

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	72	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	26	1
Zajęcia o charakterze praktycznym	22	1