

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2015/2016

Wydział Fizyki, Matematyki i Informatyki

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Języki symboliczne
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	
KOD PRZEDMIOTU	WFMiI I oIN D1 15/16
KATEGORIA PRZEDMIOTU	Przedmioty specjalnościowe
LICZBA PUNKTÓW ECTS	4.00
SEMESTRY	5

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	SEMINARIUM	PROJEKT
5	18	0	18	0	0	0

3 CELE PRZEDMIOTU

Cel 1 Poznanie paradygmatu programowania imperatywnego i jego różnicy w stosunku do deklaratywnego podejścia matematycznego

Cel 2 Poznanie składni języka LISP jako przykładu języka symbolicznego pozwalającego na implementowania przetwarzania symbolicznego. Językiem ten jest zarazem językiem ogólnego przeznaczenia umożliwiającym programowanie w stylu proceduralnym, funkcyjnym i obiektowym.

Cel 3 Nabycie umiejętności projektowania i implementacji programów napisanych w dialekcie Scheme

Cel 4 Projektowanie i implementacja programów demonstrujących koncepcje i użycie mechanizmów rekurencji, procesów iteracyjnych, procedur prostych i złożonych oraz abstrakcji danych.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Zaliczone przedmioty: Wstęp do programowania oraz Algorytmy i struktury danych.

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Student rozumie pojęcie programowania imperatywnego i jego różnicę w stosunku do deklaratywnego podejścia matematycznego. Zna podstawy składni języka LISP.

EK2 Wiedza Student rozumie na czym polega zastosowanie mechanizmów typowych dla języka w celu budowy abstrakcji na poziomie procesu obliczeniowego, złożonych procedur, danych.

EK3 Umiejętności Student umie skonfigurować środowisko programistyczne w dialekcie Scheme oraz potrafi napisać i uruchomić własny program w języku LISP rozwiązujący postawiony przed nim problem.

EK4 Umiejętności Student potrafi zaprojektować i zaimplementować program wykorzystujący mechanizmy rekurencji i procesów iteracyjnych oraz potrafi budować programy na odpowiednim poziomie abstrakcji.

6 TREŚCI PROGRAMOWE

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Informacje o przedmiocie. Podstawowe informacje o języku Lisp, jego historii i dialektach. Przedstawienie idei programowania imperatywnego w stosunku do podejścia deklaratywnego. Elementy programowania. Nazwy i środowisko.	3
W2	Proste wyrażenia i operacje na prostych typach danych. Podstawy składni. Wyrażenia symboliczne i ich wartościowanie. Procedury złożone, podstawieniowy model stosowania procedur. Stosowana a normalna kolejność obliczania. Wyrażenia warunkowe i predykaty. Procedury jako abstrakcyjne, czarne skrzynki.	3
W3	Rekursja liniowa i iteracyjna. Rekursja drzewiasta. Procedury rekurencyjne generujące procesy iteracyjne. Definicje wewnętrzne i struktura blokowa. Złożoność obliczeniowa algorytmów - rzędy wielkości. Formułowanie abstrakcji za pomocą procedur wyższych rzędów. Tworzenie procedur za pomocą lambda-abstrakcji. Tworzenie zmiennych lokalnych za pomocą let. Procedury jako metody ogólne. Procedury jako wyniki.	3
W4	Budowanie abstrakcji za pomocą danych. Bariery abstrakcji. Pojęcie danych. Dane hierarchiczne i własności domknięcia. Reprezentowanie ciągów, listy, struktury listowe. Operacje na listach, odwzorowywanie list. Hierarchiczne struktury danych, odwzorowywanie drzew. Ciągi jako interfejsy konwencjonalne. Operacje na ciągach. Odwzorowania zagnieżdżone.	3

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W5	Zbiory jako listy niuporządkowane i uporządkowane. Zbiory jako drzewa binarne. Wyszukiwanie informacji w zbiorach. Systemy z operacjami ogólnymi. Ogólne operacje arytmetyczne. Łączenie danych różnych typów. Algebra symboliczna. Modularność, obiekty i stany. Przypisanie i stan lokalny. Lokalne zmienne stanu.	3
W6	Środowiskowy model obliczeń. Reguły obliczania. Ramki jako magazyny stanów lokalnych. Definicje wewnętrzne. Modelowanie z użyciem danych modyfikowalnych. Modyfikowalne struktury listowe. Reprezentowanie kolejek. Reprezentowanie tablic. Współbieżność, mechanizmy sterowania współbieżnością. Strumienie, zastosowanie paradygmatu strumieniowego. Obliczenia odroczone. Modularność programów funkcyjnych i modularność obiektów.	3

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Konfiguracja środowiska programistycznego dla dialektu języka LISP Scheme. Obliczanie wartości prostych wyrażeń. Kombinacje wyrażeń. Operatory i argumenty, notacja prefiksowa. Zagnieżdżanie kombinacji. Formatowanie kodu źródłowego. Zmienne i ich wartości. Nazwy i środowisko. Definiowanie procedur, formy specjalne. Parametry formalne w definicji procedury.	3
L2	Podstawieniowy model stosowania procedur. Stosowana a normalna kolejność obliczeń. Wyrażenia warunkowe i predykaty, analiza przypadków. Predykaty, klauzule i następniki. Wyrażenia cond, if, else, relacje pierwotne oraz operacje logiczne takie jak: and, or, not	3
L3	Implementacja pierwiastkowania metodą Newtona. Obliczanie pierwiastków sześciennych. Implementacje programów z wykorzystaniem zmiennych lokalnych. Parametry formalne jako zmienne związane. Zakres zmiennych związanych deklarowanych jako parametry formalne procedury. Wiązanie składniowe.	3
L4	Obliczanie $n!$ i przykłady rekursji. Implementacja $n!$ jako liniowy proces rekurencyjny oraz liniowy proces iteracyjny. Rekursja ogonowa. Obliczanie liczb Fibonacciego jako przykład rekursji rozgałęziającej się (drzewiastej). Implementacja operacji potęgowania, poszukiwanie największych wspólnych dzielników oraz lic pierwszych.	3
L5	Procedury operujące na procedurach, procedury wyższego rzędu. Użycie procedur jako argumentów na przykładzie implementacji sumowania szeregów. Lambda-abstrakcje. Tworzenie zmiennych lokalnych za pomocą let. Implementacja zastosowania procedur jako metod ogólnych poprzez znajdowanie pierwiastków równań przez bisekcję. Budowa procedury wyznaczającej liczbę będącej tzw. punktem stałym. Tworzenie procedur których wyniki same są procedurami.	3

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L6	Abstrakcja danych łączenie obiektów danych w dane złożone. Implementacja przykładowych operacji arytmetycznych na liczbach wymiernych. Pary jako - przykład struktur złożonych. Tworzenie struktur hierarchicznych - struktur złożonych z części, które same są złożone z części. Budowanie ciągów i drzew za pomocą par. Implementacja list oraz podstawowych na nich operacji. Implementacja drzew. Operowanie na danych symbolicznych. Implementacja przykładowego, prostego programu różniczkowania symbolicznego. Implementacja zbiorów, tworzenie drzew Huffmana.	3

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Ćwiczenia projektowe

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	36
Konsultacje przedmiotowe	4
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	60
Opracowanie wyników	10
Przygotowanie raportu, projektu, prezentacji, dyskusji	10
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	120
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Kolokwium

OCENA PODSUMOWUJĄCA**P1** Średnia ważona ocen formujących**KRYTERIA OCENY**

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 3.0	Student rozumie pojęcie programowania imperatywnego i jego różnicę w stosunku do deklaratywnego podejścia matematycznego.
NA OCENĘ 3.5	Student zna pojęcie programowania imperatywnego. Zna podstawy składni języka LISP.
NA OCENĘ 4.0	Student rozumie pojęcie programowania imperatywnego. Zna na dobrym poziomie składnię języka LISP.
NA OCENĘ 4.5	Student rozumie pojęcie programowania imperatywnego. Zna na dobrym poziomie składnię języka LISP. Wie jakie konstrukcje językowe są adekwatne do rozważanych problemów.
NA OCENĘ 5.0	Student rozumie pojęcie programowania imperatywnego. Zna bardzo dobrze składnię języka LISP. Wie jakie konstrukcje językowe są adekwatne do rozważanych problemów i potrafi projektować systemy na nich bazujące.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 3.0	Student wie jak definiować procedury złożone. Umie przedstawić stosowaną i normalną kolejność obliczania danej procedury.
NA OCENĘ 3.5	Student wie jak definiować procedury złożone. Umie przedstawić stosowaną i normalną kolejność obliczania danej procedury. Potrafi formułować abstrakcje za pomocą procedur wyższego rzędu.
NA OCENĘ 4.0	Student wie jak definiować procedury złożone. Umie przedstawić stosowaną i normalną kolejność obliczania danej procedury. Potrafi formułować abstrakcje za pomocą procedur wyższego rzędu z wykorzystaniem lambda-abstrakcji.
NA OCENĘ 4.5	5 Oprócz wymagań takich jak na ocenę 4.0 student potrafi dobrać odpowiednie struktury danych do rozważanego problemu.
NA OCENĘ 5.0	Student wie jakie mechanizmy abstrakcji są adekwatne do rozważanego problemu oraz potrafi wskazać mechanizmy i odpowiednie struktury danych do ich realizacji typowe dla języka LISP.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 3.0	Student potrafi skonfigurować środowisko programistyczne dialektu Scheme. Potrafi zrozumieć i zapisać program na podstawie lektury kodu źródeł.
NA OCENĘ 3.5	Potrafi napisać program w języku LISP realizujący przyjęte założenia.
NA OCENĘ 4.0	Potrafi przeprowadzić analizę pozwalającą ustalić wymagania jakie stawiamy programowi.
NA OCENĘ 4.5	Potrafi uogólnić rozwiązanie w celu dostosowania go do innych typów danych.

NA OCENĘ 5.0	Potrafi przeanalizować zaproponowane rozwiązanie w kontekście jego zalet w stosunku do innych języków programowania.
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 3.0	Student potrafi rozwiązać problem z wykorzystaniem procedur rekurencyjnych
NA OCENĘ 3.5	Potrafi przedstawić rozwiązanie zadanego problemu przy pomocy rekursji liniowej i iteracyjnej.
NA OCENĘ 4.0	Potrafi zaimplementować procedury wyższego rzędu z użyciem lambda-abstrakcji. Potrafi wykorzystywać procedury jako wyniki.
NA OCENĘ 4.5	Potrafi wykorzystywać struktury listowe, drzewiaste oraz zbiory oraz dokonywać na nich odpowiednich operacji.
NA OCENĘ 5.0	Potrafi łączyć różne typy danych oraz modelować z użyciem danych modyfikowalnych.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	I1_W01, I1_W04, I1_W06, I1_W07, I1_W09, I1_W10, I1_W12, I1_W13, I1_U01, I1_U02, I1_U04, I1_U05, I1_U07, I1_U14, I1_U16	Cel 1 Cel 2 Cel 3 Cel 4	W1 W2 W3 L1 L2	N1 N2	F1
EK2	I1_W06, I1_W08, I1_W10, I1_W13	Cel 2 Cel 3 Cel 4	W1 W2 W3 W4 W5 L1 L2 L3 L4 L5	N1 N2	F1

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK3	I1_W06, I1_W10, I1_U07, I1_U09	Cel 2 Cel 3 Cel 4	W1 W2 W3 W4 W5 W6 L1 L2 L3 L4 L5	N1 N2 N3	F1 P1
EK4	I1_W06, I1_W09, I1_W10, I1_U09, I1_U14, I1_U16	Cel 4	W1 W2 W3 W4 W5 W6 L1 L2 L3 L4 L5 L6	N1 N2 N3	F1 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

[1] | **Harold Abelson and Gerald Jay Sussman Sussman** — *Struktura i interpretacja programów komputerowych*, Warszawa, 2002, WNT

LITERATURA UZUPEŁNIAJĄCA

[1] | **Matthias Felleisen Robert Bruce Findler Matthew Flatt Shriram Krishnamurthi** — *How to Design Programs*, London, 2001, The MIT Press

[2] | **Brian Harvey Matthew Wright** — *Simply Scheme*, London, 1999, The MIT Press

[3] | **[Daniel P. Friedman, Matthias Felleisen** — *The Little Schemer*, London, 1995, The MIT Press

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Tomasz Gąciarz (kontakt: tga@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr inż. Tomasz Gąciarz (kontakt: tga@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....