

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2022/2023

Wydział Informatyki i Telekomunikacji

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	Symbolic programming
KOD PRZEDMIOTU	WiT I oIN D1 22/23
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 składni języka Python. Poznanie podstawowych typów danych i instrukcje sterujących używanych w języku Python. Nabycie umiejętności projektowania i implementacji programów w języku Python. Umiejętność wykorzystania zaawansowanych typów danych języka Python. Umiejętność wykorzystania języka Python w zakresie programowania w modelu obiektowym.

Cel 2 Poznanie składni języka LISP jako przykładu języka symbolicznego pozwalającego na implementowania przetwarzania symbolicznego. Nabycie umiejętności projektowania i implementacji programów napisanych w dialekcie Scheme. 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 zna składnię języka Python oraz typowych dla niego typów danych. Umie wykorzystać zaawansowane możliwości języka Python w zakresie programowania w modelu proceduralnym i obiektowym.

EK2 Wiedza Student rozumie pojęcie programowania imperatywnego i jego różnicę w stosunku do deklaratywnego podejścia matematycznego. Zna podstawy składni języka Lisp. Student rozumie na czym polega zastosowanie mechanizmów typowych dla języka w celu budowy abstrakcji na poziomie procesu obliczeniowego, złożonych procedur, danych.

EK3 Umiejętności Student potra pisać programy użytkowe w języku Python oraz odpowiednio dobrać model programowania i struktury danych. Umie skorzystać z dokumentacji języka Python.

EK4 Umiejętności Student umie skonguować środowisko programistyczne w dialekcie Scheme oraz potra napisać i uruchomić własny program w języku Lisp rozwiązujący postawiony przed nim problem.

6 TREŚCI PROGRAMOWE

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BŁOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Podstawy składni języka Python. Podstawowe struktury danych. Podstawowe operacje na łańcuchach znakowych, listach i słownikach.	3
L2	Definicje funkcji i argumentów, podstawowe funkcje wbudowane, wyrażenia warunkowe, wyrażenia lambda, pętle. Iterowanie po indeksach sekwencji. Domyślne argumenty funkcji. Tworzenie własnych modułów i pakietów. Zasady zasięgu zmiennych. Implementacja rozbudowanych struktur danych. Listy zagnieżdżone, krotki, zbiory, słowniki.	3
L3	Operacje na plikach. Formatowanie napisów. Tworzenie i wykorzystanie iteratorów i generatorów. Obsługa błędów i wyjątków. Wyjątki definiowane przez użytkownika. Predefiniowane akcje typu clean up.	3
L4	Programowanie obiektowe, hierarchia klas. Obiekty klas i metod. Zmienne obiektów (instancji) i zmienne klas. Atrybuty klas i obiektów. Dziedziczenie klas. Zmienne prywatne. Wyjątki jako klasy.	3

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L5	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. Podstawieniowy model stosowania procedur. Stosowana a normalna kolejność obliczeń. Wyrażenia warunkowe i predykaty, analiza przypadków. Predykaty, klauzule i następni. Wyrażenia cond, if, else, relacje pierwotne oraz operacje logiczne takie jak: and, or, not. 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
L6	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 liczb pierwszych. 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. Tworzenie procedur których wyniki same są procedurami. Abstrakcja danych łączenie obiektów danych w dane złożone. Implementacja przykładowych operacji arytmetycznych na liczbach wymiernych. Pary jako - przykład struktur złożonych.	3

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Informacje o przedmiocie. Wprowadzenie do języka Python. Wersje języka. Opis interpretera. Tryby pracy interpretera. Raportowanie błędów. Automatyczne uruchamianie skryptów języka Python. Kodowanie znaków, Konwencje dotyczące komentarzy. Podstawowe typy danych języka Python. Podstawowe operacje na łańcuchach znakowych, listach i słownikach.	3
W2	Definicje funkcji i argumentów, wyrażenia warunkowe, wyrażenia lambda, pętle. Iterowanie po indeksach sekwencji. Domyślne argumenty funkcji. Napisy dokumentujące. Konwencje kodowania i stylu. Rozbudowane struktury danych stos, kolejki. Listy zagnieżdżone, krotki, zbiory, słowniki. Mutowalne i niemutowalne typy danych. Moduły i pakiety oraz ich import. Kompilacja w języku Python. Odwołania relatywne i absolutne.	3
W3	Operacje wejścia i wyjścia języka Python. Konwersja danych do stringów. Formatowanie napisów. Czytanie i pisanie do plików. Tworzenia obiektów typu: persistent. Zgłaszanie i obsługa błędów i wyjątków. Wyjątki definiowane przez użytkownika. Predefiniowane akcje typu clean up.	3

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W4	Przestrzenie nazw. Instrukcje nonlocal i global. Zasięg zmiennych. Programowanie obiektowe w Pythonie. Definicja klas. Obiekty klas i metod. Zmienne obiektów (instancji) i zmienne klas. Atrybuty klas i obiektów. Dziedziczenie klas. Zmienne prywatne. Wyjątki jako klasy. Iteratory i generatory. Biblioteka standardowa. Operacje na wzorcach. Obliczenia matematyczne. Kontrola jakości pisanego oprogramowania, testy.	3
W5	Podstawowe informacje o języku Lisp, jego historii i dialektach. Przedstawienie idei programowania imperatywnego w stosunku do podejścia deklaratywnego. Elementy programowania. Nazwy i środowisko programistyczne. Proste wyrażenia i operacje na prostych typach danych. Podstawy składni. Procedury złożone. Podstawieniowy model stosowania procedur. Stosowana a normalna kolejność obliczania. Wyrażenia warunkowe i predykaty. Procedury jako abstrakcyjne, czarne skrzynki. 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
W6	Budowanie abstrakcji za pomocą danych. Bariery abstrakcji. Pojęcie danych. Dane hierarchiczne i własności domknięcia. Reprezentowanie ciągów, list, struktur listowych. Operacje na listach, odwzorowywanie list. Wyrażenia symboliczne i ich wartościowanie. Hierarchiczne struktury danych, odwzorowywanie drzew. Ciągi jako interfejsy konwencjonalne. Operacje na ciągach. Odwzorowania zagnieżdżone. Zbiory jako listy nieuporządkowane i uporządkowane. Zbiory jako drzewa binarne. Wyszukiwanie informacji w zbiorach. Systemy z operacjami ogólnymi. Ogólne operacje arytmetyczne. Łączenie danych różnych typów. Algebra symboliczna.	3

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

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	10
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	20
Opracowanie wyników	20
Przygotowanie raportu, projektu, prezentacji, dyskusji	20
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	106
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Sprawozdanie z ćwiczenia laboratoryjnego

F2 Kolokwium

OCENA PODSUMOWUJĄCA

P1 średnia

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Warunkiem jest zaliczenie wszystkich laboratoriów, kolokwiów i projektu na ocenę co najmniej 3.0

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Student nie zna podstawowych elementów składni języka Python.
NA OCENĘ 3.0	Student zna składnię języka Python i umiejętnie wykorzystuje podstawowe typy danych.
NA OCENĘ 3.5	Student wie jak konstruować klasy w Pythonie. Student wie jak pisać własne klasy do obsługi błędów.

NA OCENĘ 4.0	Student wie jak poprawnie i modułowo zrealizować złożony projekt w Pythonie wraz z obsługą błędów, oraz dobrą dokumentacją.
NA OCENĘ 4.5	Student wie jak poprawnie i modułowo zrealizować złożony projekt w Pythonie wraz z obsługą błędów, oraz dobrą dokumentacją. Umie wykorzystać podstawowe biblioteki.
NA OCENĘ 5.0	Student wie jak poprawnie i modułowo zrealizować złożony projekt w Pythonie wraz z obsługą błędów oraz dobrą dokumentacją. Umie wykorzystać podstawowe biblioteki i umiejętnie integrować kod pisany w Pythonie z językiem C.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Student nie zna podstawowych elementów składni języka Lisp.
NA OCENĘ 3.0	Student zna podstawy składni języka Lisp.
NA OCENĘ 3.5	Student zna składnię języka Lisp umiejętnie wykorzystuje podstawowe typy danych.
NA OCENĘ 4.0	Student wie jak konstruować procedury złożone w Lispie.
NA OCENĘ 4.5	Student wie jak konstruować procedury operujące na procedurach, procedury wyższego rzędu, używać procedur jako argumentów oraz korzystać z lambda-abstrakcji.
NA OCENĘ 5.0	Student umie zastosować mechanizmy typowe dla języka Lisp w celu budowy abstrakcji na poziomie procesu obliczeniowego, procedur złożonych i danych.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	Student nie potra napisać prostego programu w języku Python z wykorzystaniem struktur listowych.
NA OCENĘ 3.0	Student umiejętnie programuje w Pythonie z wykorzystaniem jego podstawowej składni.
NA OCENĘ 3.5	Student umiejętnie wykorzystuje podstawowe typy danych w Pythonie.
NA OCENĘ 4.0	Student umiejętnie programuje w Pythonie w modelu obiektowym.
NA OCENĘ 4.5	Student umie zarówno napisać poprawny program w Pythonie jak i zaprojektować testy tego oprogramowania.
NA OCENĘ 5.0	Student programuje poprawnie w Pythonie, testuje oraz dokumentuje działanie programów.
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 2.0	Student nie potra napisać prostego programu w języku Lisp z wykorzystaniem struktur listowych.
NA OCENĘ 3.0	Student umie obliczać wartości prostych wyrażeń i kombinacje. Umiejętnie stosuje notację preksową.

NA OCENĘ 3.5	Student umie obliczać wartości prostych wyrażeń i kombinacje. Umiejętnie stosuje notację preksową oraz definiuje procedury złożone w Lispie.
NA OCENĘ 4.0	Student umie obliczać wartości prostych wyrażeń i kombinacje. Umiejętnie stosuje notację preksową oraz definiuje procedury złożone w Lispie. Umiejętnie wykorzystuje mechanizmy rekurencji oraz liniowych procesów iteracyjnych.
NA OCENĘ 4.5	Student umie implementować procedury operujące na procedurach, procedury wyższego rzędu, używać procedur jako argumentów oraz korzystać z lambda-abstrakcji.
NA OCENĘ 5.0	Posiada umiejętność budowania abstrakcji zarówno na poziomie obliczeń jak i danych oraz operować na danych symbolicznych.

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_W04	Cel 1	W1 W2 W3 W4	N1	F1 F2 P1
EK2	I1_W04	Cel 2	W5 W6	N1	F1 F2 P1
EK3	I1_U01b I1_U06b I1_U10	Cel 1	L1 L2 L3 L4	N2	F1 F2 P1
EK4	I1_U01b I1_U06b I1_U10	Cel 2	L5 L6	N2	F1 F2 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] | **Harold Abelson and Gerald Jay Sussman Sussman** — *Struktura i interpretacja programów komputerowych*, Warszawa, 2002, WNT
- [2] | **Mark Lutz** — *Learning Python*, Beijing, 2013, O'Reilly Media, Inc.

LITERATURA UZUPEŁNIAJĄCA

- [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

[4] Mark Pilgrim — *Dive into Python*, Miejscość, 2004, Wydawnictwo

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Tomasz Gąciarz (kontakt: tomasz.gaciarz@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)

.....