



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Algorytmika i programowanie

Sylabus zajęć

Informacje podstawowe

Kierunek studiów Nauczanie matematyki i informatyki		Cykl dydaktyczny 2024/25 Kod zajęć 06NMIS.12K.07400.24 Języki wykładowe polski Obligatoryjność Obowiązkowy Blok zajęciowy Przedmioty kierunkowe
Specjalność -		
Jednostka organizacyjna Wydział Matematyki i Informatyki		
Poziom studiów studia pierwszego stopnia		
Forma studiów studia stacjonarne		
Profil studiów profil ogólnoakademicki		
Koordynator zajęć	Michał Rzeczkowski	
Prowadzący zajęcia	Michał Rzeczkowski	
Okres Semestr 2	Forma zajęć / liczba godzin / forma zaliczenia • Wykład: 30, Egzamin • Ćwiczenia: 15, Zaliczenie z oceną • Laboratorium: 45, Zaliczenie z oceną	Liczba punktów ECTS 8

Cele kształcenia dla zajęć

Kod	Cel
C1	Zapoznanie studentów z podstawowymi pojęciami z algorytmiki.
C2	Zaznajomienie studentów ze środowiskami programistycznymi do programowania w języku Python.
C3	Zapoznanie studentów z podstawami programowania w języku Python.
C4	Zapoznanie studentów z podstawowymi kryteriami oceny jakości algorytmów.

Wymagania wstępne

Umiejętność programowania w języku wizualnym.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	zna i rozumie pojęcie algorytmu.	NMI_K1_W07, NMI_K1_W08	Test
W2	zna i rozróżnia takie pojęcia jak IDE, edytor, notebook (jupyter, Google Colaboratory), kompilator, interpreter.	NMI_K1_W06, NMI_K1_W07	Test
W3	zna podstawy Pythona - funkcje print i input, operacja przypisania, obliczenia, kontrola przepływu programu (pętle, instrukcje warunkowe), praca z danymi tekstowymi (wycinanie i podstawowe metody klasy string), podstawowe typy danych (liczbowe, logiczne, łańcuchy).	NMI_K1_W07	Test
W4	zna podstawowe struktury danych w języku Python (listy, zbiory, krotki, słowniki) oraz metody z nimi związane.	NMI_K1_W07	Test
W5	zna pojęcie funkcji i rozumie ich znaczenie w programowaniu. Zna pojęcia związane z argumentami funkcji (parametry nazwane, opcjonalne, *args i **kwargs, zasięg zmiennych). Rozumie pojęcie rekurencji. Zna zasady metody "dziel i zwyciężaj" oraz przykłady algorytmów zaprojektowanych tą metodą.	NMI_K1_W07	Test
W6	zna podstawowe metody klasy String. Zna zasady operacji na plikach i składnię w Pythonie, która te operacje umożliwia.	NMI_K1_W07, NMI_K1_W08	Test
W7	rozumie pojęcie złożoności czasowej algorytmu. Zna i rozumie notację asymptotyczną. Zna podstawowe algorytmy sortowania i przeszukiwania.	NMI_K1_W05, NMI_K1_W07	Test
W8	zna podstawy programowania obiektowego w języku Python oraz składnię pozwalającą na tworzenie klas i metod.	NMI_K1_W05, NMI_K1_W07	Test
W9	zna różne sposoby i zasady tworzenia i importowania bibliotek oraz posługiwania się dokumentacją.	NMI_K1_W07	Test
Umiejętności - Student/ka:			
U1	potrafi zapisać proste algorytmy za pomocą schematów blokowych, w pseudokodzie oraz w języku Python.	NMI_K1_U08, NMI_K1_U09	Kolokwium pisemne, Test, Projekt
U2	potrafi przygotować środowisko programistyczne do pracy z językiem Python. Potrafi pracować z Pythonem z poziomu linii poleceń (terminala)	NMI_K1_U07, NMI_K1_U09	Test, Projekt

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U3	potrafi ułożyć algorytm rozwiązujący proste zadania matematyczne (liczenie sumy silni, porównywanie elementów itp.) i zaimplementować go w języku Python. Potrafi wczytywać i wyświetlać dane, konwertować odpowiednie typy danych.	NMI_K1_U08, NMI_K1_U09	Kolokwium pisemne, Test, Projekt, Sprawdziany z umiejętności programowania
U4	potrafi utworzyć i iterować podstawowe struktury danych w Pythonie oraz efektywnie wykorzystać je w tworzeniu programów.	NMI_K1_U08, NMI_K1_U09	Test, Projekt, Sprawdziany z umiejętności programowania
U5	potrafi zaimplementować funkcję i użyć jej w programie. Potrafi ułożyć algorytm rekurencyjny do rozwiązania problemów o charakterze rekurencyjnym. Sprawnie wykorzystuje możliwości typowych funkcji zaimplementowanych w Pythonie.	NMI_K1_U05, NMI_K1_U08, NMI_K1_U09	Kolokwium pisemne, Test, Projekt, Sprawdziany z umiejętności programowania
U6	umie wykorzystać metody klasy String do pracy z tekstem. Potrafi pracować z plikami (operacje wczytywania, zapisywania i modyfikacji pliku, umiejętne posługiwanie się ścieżkami do pliku).	NMI_K1_U05, NMI_K1_U08, NMI_K1_U09	Test, Projekt, Sprawdziany z umiejętności programowania
U7	potrafi prawidłowo ocenić złożoność czasową prostego algorytmu posługując się notacją asymptotyczną.	NMI_K1_U05, NMI_K1_U08, NMI_K1_U09	Kolokwium pisemne, Test
U8	potrafi tworzyć klasy oraz ich instancje oraz wykorzystywać je do pisania efektywnego i czytelnego kodu.	NMI_K1_U05, NMI_K1_U08, NMI_K1_U09	Test, Projekt, Sprawdziany z umiejętności programowania
U9	potrafi rozwiązywać złożone zadania programistyczne wykorzystując wybrane biblioteki w języku Python.	NMI_K1_U05, NMI_K1_U07, NMI_K1_U08, NMI_K1_U09	Projekt
U10	potrafi identyfikować różne rodzaje błędów w programach oraz dokonywać niezbędnych poprawek.	NMI_K1_U08, NMI_K1_U09	Projekt, Sprawdziany z umiejętności programowania
U11	potrafi opisać działanie programu na podstawie zapisanego kodu.	NMI_K1_U05, NMI_K1_U08, NMI_K1_U09	Projekt, Sprawdziany z umiejętności programowania
U12	potrafi w efektywny sposób posługiwać się konstrukcjami służącymi do kontroli przepływu programu (operacje przypisania, operacje arytmetyczne, instrukcje warunkowe, pętle, operacje) oraz strukturami danych w rozwiązywaniu zadań algorytmicznych i programistycznych.	NMI_K1_U08, NMI_K1_U09	Projekt, Sprawdziany z umiejętności programowania
Kompetencji społecznych - Student/ka:			
K1	rozumie znaczenie algorytmiki i umiejętności programowania we współczesnym świecie oraz w procesie kształcenia.	NMI_K1_K01, NMI_K1_K02	Test

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Znaczenie algorytmiki we współczesnym świecie i procesie uczenia. Algorytm jako istota każdego programu. Sposoby zapisywania algorytmów. Tworzenie prostych algorytmów w postaci schematów blokowych, pseudokodu, prostych programów w języku Python.	W1, U1, K1	Wykład, Ćwiczenia, Laboratorium
2.	Przygotowanie środowiska programistycznego do pracy z językiem Python. Przegląd różnych środowisk.	W2, U2	Wykład, Laboratorium
3.	Podstawy Pythona - funkcje print i input, operacja przypisania, obliczenia, kontrola przepływu programu (pętle, instrukcje warunkowe), praca z danymi tekstowymi (wycinanie i podstawowe metody klasy string), typy danych. Różne rodzaje błędów programistycznych i sposoby radzenia sobie z błędami.	W3, U10, U11, U12, U3	Wykład, Ćwiczenia, Laboratorium
4.	Podstawowe struktury danych w języku Python (listy, zbiory, krotki, słowniki) i ich umiejętne wykorzystanie.	W4, U11, U12, U4	Wykład, Laboratorium
5.	Funkcje i ich zastosowania. Argumenty funkcji (nazwane, opcjonalne, *args i **kwargs). Zasięg zmiennych. Rekurencja, metoda "dziel i zwyciężaj".	W5, U10, U11, U12, U5	Wykład, Ćwiczenia, Laboratorium
6.	Klasa string i operacje na plikach.	W6, U12, U6	Wykład, Laboratorium
7.	Czasowa złożoność algorytmu. Notacja asymptotyczna. Algorytmy sortowania i przeszukiwania.	W7, U11, U12, U7	Wykład, Ćwiczenia, Laboratorium
8.	Elementy programowania obiektowego.	W8, U8	Wykład, Laboratorium
9.	Rozwiązywanie złożonych zagadnień programistycznych z wykorzystaniem wybranych bibliotek języka Python.	W9, U10, U11, U12, U8, U9	Wykład, Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Wykład	Wykład z prezentacją multimedialną wybranych zagadnień
Ćwiczenia	Metoda ćwiczeniowa, Metoda laboratoryjna
Laboratorium	Metoda laboratoryjna

Forma zajęć	Warunki zaliczenia zajęć
Wykład	Zdobycie z testu powyżej 50 punktów. Progi punktowe: $x \leq 50$ punktów - niedostateczny $50 < x \leq 60$ punktów - dostateczny $60 < x \leq 70$ punktów - dostateczny plus $70 < x \leq 80$ punktów - dobry $80 < x \leq 90$ punktów - dobry plus $x > 90$ punktów - bardzo dobry
Ćwiczenia	Zdobycie powyżej 50 punktów z kolokwium pisemnego (100 punktów). Progi punktowe: $x \leq 50$ punktów - niedostateczny $50 < x \leq 60$ punktów - dostateczny $60 < x \leq 70$ punktów - dostateczny plus $70 < x \leq 80$ punktów - dobry $80 < x \leq 90$ punktów - dobry plus $x > 90$ punktów - bardzo dobry
Laboratorium	Sprawdziany z umiejętności programowania - 5 sprawdzianów każdy za 20 punktów. Projekt będzie składał się z trzech części: dwie pierwsze części za 30 punktów, ostatnia część za 40 punktów. Progi punktowe: $x \leq 100$ punktów - niedostateczny $100 < x \leq 120$ punktów - dostateczny $120 < x \leq 140$ punktów - dostateczny plus $140 < x \leq 160$ punktów - dobry $160 < x \leq 180$ punktów - dobry plus $x > 180$ punktów - bardzo dobry Aby otrzymać zaliczenie z laboratorium student będzie musiał uzyskać minimum 101 punktów.

Literatura

Obowiązkowa

1. Harel D., "Rzecz o istocie informatyki", Wydawnictwa Naukowo-Techniczne, Warszawa 2008.
2. Wirth N., "Algorytmy + struktury danych = programy", Wydawnictwa Naukowo-Techniczne, Warszawa 2004.
3. <https://www.w3schools.com/python/>

Dodatkowa

1. <https://docs.python.org/pl/3.8/tutorial/index.html>

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Wykład	30
Ćwiczenia	15
Laboratorium	45
Przygotowanie projektu	40
Czytanie wskazanej literatury	30

Przygotowanie do zaliczenia	80
Łączny nakład pracy studenta	Liczba godzin 240
Liczba punktów ECTS	ECTS 8

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
NMI_K1_K01	Absolwent/ka jest gotów/gotowa do poznania ograniczeń własnej wiedzy i rozumienia potrzeby dalszego kształcenia,
NMI_K1_K02	Absolwent/ka jest gotów/gotowa do popularyzowania wiedzy z zakresu matematyki i informatyki,
NMI_K1_U05	Absolwent/ka potrafi mówić o zagadnieniach matematycznych i informatycznych poprawnym, zrozumiałym językiem,
NMI_K1_U07	Absolwent/ka potrafi dobrać i wykorzystać odpowiednie narzędzia i technologie wspomagające pracę nauczyciela,
NMI_K1_U08	Absolwent/ka potrafi wykorzystać algorytmy i struktury danych do wydajnego rozwiązania problemu,
NMI_K1_U09	Absolwent/ka potrafi zaprojektować i zapisać program w wybranych językach programowania,
NMI_K1_W05	Absolwent/ka zna i rozumie złożone zależności występujące w obrębie wiedzy matematycznej i informatycznej,
NMI_K1_W06	Absolwent/ka zna i rozumie narzędzia i technologie wspierające pracę nauczyciela i wychowawcy, w tym narzędzia chmurowe,
NMI_K1_W07	Absolwent/ka zna i rozumie zagadnienia związane z algorytmiką i programowaniem w wybranych językach programowania (wizualnych i tekstowych),
NMI_K1_W08	Absolwent/ka zna i rozumie teoretyczne podstawy informatyki, m.in. sposoby reprezentacji informacji w komputerze, zasady przetwarzania informacji, budowę i zasady działania komputera i sieci komputerowych, w tym sieci Internet,