



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Wstęp do programowania [S1MNT1>WdP]

Przedmiot

Kierunek studiów

Matematyka nowoczesnych technologii

Rok/Semestr

1/1

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

15

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Barbara Szyszka

barbara.szyszka@put.poznan.pl

mgr inż. Marta Kańczurzevska-Staszak

marta.kanczurzevska-staszak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Podstawowa wiedza z zakresu szkoły średniej. Umiejętność obsługi komputera

Cel przedmiotu

Celem przedmiotu jest zapoznanie studentów z podstawami programowania komputerów oraz nauczanie podstaw programowania w języku Python i MATLAB. W szczególności obejmuje to przekazanie studentom podstawowych informacji o arytmetyce komputerowej, programowaniu strukturalnym, algorytmizacji problemów i ich oprogramowaniu (również w postaci funkcji), nauczanie studentów biegłego posługiwania się zintegrowanym środowiskiem programistycznym

Przedmiotowe efekty uczenia się

Wiedza:

- student ma wiedzę dotyczącą zastosowań narzędzi matematycznych [K_W01(P6S_WG), K_W05(P6S_WG), K_W07(P6S_WG)];

- student zna podstawy techniki obliczeniowych i programowania [K_W01(P6S_WG), K_W05(P6S_WG), K_W07(P6S_WG)].

Umiejętności:

- student potrafi skonstruować algorytm rozwiązania prostego zadania oraz zaimplementować i przetestować go w wybranym środowisku programistycznym [K_U04(P6S_UW), K_U05(P6S_UW), K_U11(P6S_UW)];
- student potrafi zgodnie z ogólnymi wymogami eksploatować urządzenia oraz umie stosować zasady bezpieczeństwa i higieny pracy w laboratorium komputerowym [K_U04(P6S_UW), K_U05(P6S_UW), K_U11(P6S_UW)].

Kompetencje społeczne:

- student ma świadomość poziomu swojej wiedzy [K_K01(P6S_KK), K_K02(P6S_KK)];
- student ma świadomość pogłębiania i poszerzania wiedzy z programowania [K_K01(P6S_KK), K_K02(P6S_KK)].

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykłady: zaliczenie wykładu w formie kolokwium pisemnego o charakterze problemowym i praktycznym;
Laboratoria: dwa sprawdziany w trakcie trwania semestru. Premiowana aktywność w trakcie zajęć.

Treści programowe

Aktualizacja: 01.06.2023r.

Wykłady & Laboratoria:

- arytmetyka komputerowa;
 - reprezentacja maszynowa liczb;
 - kodowanie liczb całkowitych i zmiennoprzecinkowych;
 - konwersja systemu dziesiętnego na dwójkowy i odwrotnie;
 - kodowanie ujemnych liczb całkowitych;
 - kodowanie znaków w komputerze;
- algorytmy;
 - definicja algorytmów;
 - poprawność algorytmów;
 - pseudokod jako jeden ze sposobów zapisu algorytmów;
 - schematy blokowe jako jeden ze sposobów zapisu algorytmów;
 - bloki stosowane w zapisie algorytmów;
 - przykłady znanych algorytmów;
- złożoność obliczeniowa;
 - definicja złożoności obliczeniowej;
 - przypadki złożoności obliczeniowej;
 - notacja wielkiego O;
 - wyznaczanie złożoności obliczeniowej;
- operatory, pętle i instrukcje warunkowe;
 - operatory arytmetyczne i logiczne;
 - operatory przypisania;
 - deklaracja zmiennych;
 - instrukcje warunkowe: if, switch;
 - pętle: for, while, do while;
 - słowa kluczowe break, continue i return;
- tablice;
 - struktura tablicy - macierze i wektory;
 - deklaracja tablicy;
 - odwoływanie się do elementów tablicy;
 - iterowanie po elementach tablicy;
 - działania na tablicach;
- funkcje;
 - motywacja funkcji w programowaniu;
 - przykłady funkcji wbudowanych;

- tworzenie funkcji;
- wywoływanie funkcji;
- funkcje anonimowe;
- porównanie podstawowych instrukcji w matlabie i pythonie.

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

- arytmetyka komputerowa;
 - reprezentacja maszynowa liczb;
 - kodowanie liczb całkowitych i zmiennoprzecinkowych;
 - konwersja systemu dziesiętnego na dwójkowy i odwrotnie;
 - kodowanie ujemnych liczb całkowitych;
 - kodowanie znaków w komputerze;
- algorytmy;
 - definicja algorytmów;
 - poprawność algorytmów;
 - pseudokod jako jeden ze sposobów zapisu algorytmów;
 - schematy blokowe jako jeden ze sposobów zapisu algorytmów;
 - bloki stosowane w zapisie algorytmów;
 - przykłady znanych algorytmów;
- złożoność obliczeniowa;
 - definicja złożoności obliczeniowej;
 - przypadki złożoności obliczeniowej;
 - notacja wielkiego O;
 - wyznaczanie złożoności obliczeniowej;
- operatory, pętle i instrukcje warunkowe;
 - operatory arytmetyczne i logiczne;
 - operatory przypisania;
 - deklaracja zmiennych;
 - instrukcje warunkowe: if, switch;
 - pętle: for, while, do while;
 - słowa kluczowe break, continue i return;
- tablice;
 - struktura tablicy - macierze i wektory;
 - deklaracja tablicy;
 - odwoływanie się do elementów tablicy;
 - iterowanie po elementach tablicy;
 - działania na tablicach;
- funkcje;
 - motywacja funkcji w programowaniu;
 - przykłady funkcji wbudowanych;
 - tworzenie funkcji;
 - wywoływanie funkcji;
 - funkcje anonimowe;
- struktury danych w pythonie:
 - słowniki;
 - krotki;
 - zbiory;
- porównanie podstawowych instrukcji w matlabie i pythonie.

Program laboratorium obejmuje następujące zagadnienia:

- arytmetyka komputerowa;
 - reprezentacja maszynowa liczb;
 - kodowanie liczb całkowitych i zmiennoprzecinkowych;
 - konwersja systemu dziesiętnego na dwójkowy i odwrotnie;
 - kodowanie ujemnych liczb całkowitych;
 - kodowanie znaków w komputerze;
- algorytmy;
 - definicja algorytmów;
 - poprawność algorytmów;

- pseudokod jako jeden ze sposobów zapisu algorytmów;
- schematy blokowe jako jeden ze sposobów zapisu algorytmów;
- bloki stosowane w zapisie algorytmów;
- przykłady znanych algorytmów;
- złożoność obliczeniowa;
 - definicja złożoności obliczeniowej;
 - przypadki złożoności obliczeniowej;
 - notacja wielkiego O;
 - wyznaczanie złożoności obliczeniowej;
- operatory, pętle i instrukcje warunkowe;
 - operatory arytmetyczne i logiczne;
 - operatory przypisania;
 - deklaracja zmiennych;
 - instrukcje warunkowe: if, switch;
 - pętle: for, while, do while;
 - słowa kluczowe break, continue i return;
- tablice;
 - struktura tablicy - macierze i wektory;
 - deklaracja tablicy;
 - odwoływanie się do elementów tablicy;
 - iterowanie po elementach tablicy;
 - działania na tablicach;
- funkcje;
 - motywacja funkcji w programowaniu;
 - przykłady funkcji wbudowanych;
 - tworzenie funkcji;
 - wywoływanie funkcji;
 - funkcje anonimowe;
- struktury danych w pythonie:
 - słowniki;
 - krotki;
 - zbiory;
- porównanie podstawowych instrukcji w matlabie i pythonie.

Metody dydaktyczne

Wykłady: prezentacja multimedialna uzupełniana przykładami;

Laboratoria: ćwiczenia praktyczne i pisanie programów w języku Python i MATLAB.

Literatura

Podstawowa:

- Cormen T.H., Leiserson Ch.E., Rivest R.L. Wprowadzenie do algorytmów, WNT, 1994;
- Brzózka J., Dorobczyński L. MATLAB : środowisko obliczeń naukowo-technicznych, MIKOM, 2008;
- Summerfield M. Python 3: kompletne wprowadzenie do programowania, Helion, 2010.

Uzupełniająca:

- Mrozek B., Mrozek Z. MATLAB i Simulink Poradnik użytkownika. Wydanie II, Helion, Wrocław, 2004;
- Lutz M. Python. Wprowadzenie. Wydanie IV, Helion 2010.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	47	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	53	2,00