**Nazwa przedmiotu:**

Podstawy programowania i przetwarzania danych

**Koordynator przedmiotu:**

dr hab. inż. Marek Gągolewski, prof. PW

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Inżynieria i Analiza Danych

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-DS000-ISP-0115

**Semestr nominalny:**

1 / rok ak. 2019/2020

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. godziny kontaktowe – 80 h; w tym
 a) obecność na wykładach – 30 h
 b) obecność na ćwiczeniach – 15 h
 c) obecność na laboratoriach – 30 h
 d) konsultacje – 5 h
2. praca własna studenta – 50 h; w tym
 a) przygotowanie do ćwiczeń i kolokwium – 10 h
 b) przygotowanie do laboratoriów – 10 h
 c) zapoznanie się z literaturą – 5 h
 d) zrobienie prac domowych – 25 h
Razem 130 h, co odpowiada 5 pkt. ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1. obecność na wykładach – 30 h
2. obecność na ćwiczeniach – 15 h
3. obecność na laboratoriach – 30 h
4. konsultacje – 5 h
Razem 80 h, co odpowiada 3 pkt. ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. obecność na laboratoriach – 30 h
2. przygotowanie do laboratoriów – 15 h
3. zrobienie prac domowych – 25 h
Razem 70 h, co odpowiada 3 pkt. ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 15h |
| Laboratorium:  | 30h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

.

**Cel przedmiotu:**

Kurs jest wprowadzeniem do programowania imperatywnego z użytkowymi elementami technik programowania obiektowego na przykładzie języka Python 3. Student poznaje pojęcie algorytmu, funkcji, programu, rekurencji, tablicy (listy), a także najbardziej podstawowe algorytmy, które mogą być wykorzystywane w przetwarzaniu danych, m.in. proste algorytmy sortowania i działania na wektorach oraz macierzach. Zapoznaje się także z wybranymi funkcjami z biblioteki pakietów dla środowiska Python, np. służącymi do generowania wykresów lub operacji na wektorach i macierzach (tensorach).

Na zajęciach ćwiczeniowych student rozwija umiejętności analizy zagadnień problemowych i tworzenia algorytmów służących do ich rozwiązania z wykorzystaniem poznanych na wykładzie wiadomości teoretycznych.
Na zajęciach laboratoryjnych student uczy się praktycznych umiejętności tworzenia pełnych programów, które są oparte na poznanych algorytmach. Szczególną uwaga zwraca się więc na: implementację programu z użyciem gotowych, udokumentowanych bibliotek, umiejętność przetestowania programu, jego użycia na konkretnych danych wejściowych, interpretację otrzymanego wyniku. Na 5-7 zajęciach laboratoryjnych student rozwiązuje samodzielnie zadania sprawdzające.

**Treści kształcenia:**

1. Pojęcie problemu i algorytmu
2. Typy skalarne: bool, int, float, str. Reprezentacja liczb całkowitych i zmiennopozycyjnych. Operatory logiczne, arytmetyczne, relacyjne i przypisania. Priorytety operatorów. Błędy arytmetyki zmiennopozycyjnej
3. Instrukcja warunkowa i pętle. Miary złożoności algorytmów.
4. Struktura programu. Funkcje. Dokumentowanie i testowanie kodu. Przekazywanie parametrów przez wartość a przez referencję.
5. Implementacja algorytmów typu: metoda bisekcji znajdowania zera funkcji, największy wspólny dzielnik, sprawdzanie, czy liczba jest pierwsza itd.
6. Listy (ciągi, tablice)
7. Implementacja algorytmów typu: zastosowywanie danej operacji na każdym elemencie listy, obliczanie skumulowanej sumy, iterowanej różnicy, operacje na iloczynie kartezjańskim elementów dwóch list, operacje na odpowiadających sobie elementach dwóch i więcej równolicznych list; sortowanie proste, sortowanie kubełkowe, wyszukiwanie mody (dominanty) w liście, zliczanie liczby unikatowych elementów w wektorze itp.
8. Rekurencja. Algorytmy z nawrotami. Algorytmy typu dziel i rządź
9. Implementacja algorytmów typu: wieże z Hanoi, problem 8 hetmanów, wyszukiwanie binarne (połówkowe), znajdowanie pary najbliższych punktów w R2 itp.
10. Elementy programowania obiektowego: proste klasy, pola i metody
11. Typy zmienialne i niezmienialne. Typy iterowalne i sekwencyjne. Słowniki i zbiory
12. Implementacja algorytmów typu: zliczanie elementów, wyznaczanie średniej w podgrupach
13. Wektory i macierze (NumPy). Zwektoryzowane operacje na wektorach i macierzach (tensorach)
14. Implementacja prostych algorytmów przy użyciu gotowych funkcji i metod na podstawie dokumentacji, np. 1-najbliższych sąsiadów, proste eksperymenty symulacyjne itp.

**Metody oceny:**

Wykład:
Wykład informacyjny, problemowy, studium przypadku
Ćwiczenia:
Wspólne rozwiązywanie zadań, burza mózgów
Laboratorium:
Warsztaty przy użyciu komputera, samodzielne i wspólne rozwiązywanie zadań, burza mózgów

**Egzamin:**

nie

**Literatura:**

1. Harel D., Rzecz o istocie informatyki, Wydawnictwa Naukowo-Techniczne, Warszawa, 2008
2. Cormen T.H., Leiserson Ch., Rivest R.L., Wprowadzenie do algorytmów, Wydawnictwo Naukowe PWN, Warszawa 2013.
3. Wirth N., Algorytmy+Struktury danych=Programy, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
4. Lutz M., Learning Python, O'Reilly Media, 2013
5. Bressert E., SciPy and NumPy, O'Reilly Media, 2012
6. Gagolewski M., Bartoszuk M., Cena A., Przetwarzanie i analiza danych w języku Python, Wydawnictwo Naukowe PWN, Warszawa, 2016

**Witryna www przedmiotu:**

.

**Uwagi:**

.

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

Zna podstawy programowania w języku Python (w tym konstrukcje programistyczne: operacje przypisania, pętle, wyrażenia warunkowe, funkcje) i najważniejsze typy danych (w tym skalary, listy, słowniki, zbiory, wektory i macierze) oraz pojęcie problemu i algorytmu.

Weryfikacja:

.

**Powiązane charakterystyki kierunkowe:** DS\_W08, DS\_W14

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka W02:**

Zna proste metody i algorytmy wykorzystywane w przetwarzaniu danych

Weryfikacja:

.

**Powiązane charakterystyki kierunkowe:** DS\_W09

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi implementować proste algorytmy w postaci funkcji oraz oceniać ich złożoność obliczeniową i pamięciową oraz stosować je do konstrukcji prostych programów

Weryfikacja:

.

**Powiązane charakterystyki kierunkowe:** DS\_U11, DS\_U13, DS\_U14

**Powiązane charakterystyki obszarowe:** I.P6S\_UW

**Charakterystyka U02:**

Umie stosować wybrane narzędzia zaimplementowane w pakietach dla środowiska Python 3, w tym funkcje do tworzenia wykresów oraz wybrane operacje na wektorach i macierzach

Weryfikacja:

.

**Powiązane charakterystyki kierunkowe:** DS\_U13, DS\_U14

**Powiązane charakterystyki obszarowe:** I.P6S\_UW

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

Zna potrzebę uczenia się przez całe życie i podnoszenia kompetencji zawodowych.

Weryfikacja:

.

**Powiązane charakterystyki kierunkowe:** DS\_K01

**Powiązane charakterystyki obszarowe:** I.P6S\_KK