

## KARTA KURSU

Nazwa	Programowanie Funkcyjne
Nazwa w j. ang.	Functional programming

Koordinator	dr Roman Czapla	Zespół dydaktyczny
		dr Roman Czapla dr Iryna Artyschchuk
Punktacja ECTS*	2	

### Opis kursu (cele kształcenia)

Przedmiot ma na celu zapoznanie studentów z koncepcjami i technikami programowania w języku Python ze szczególnym uwzględnieniem paradygmatu funkcyjnego. Przedmiot prowadzony jest w języku polskim.

### Warunki wstępne

Wiedza	Student zna podstawowe pojęcia związane z programowaniem
Umiejętności	Student posiada umiejętność opisanie prostych algorytmów.
Kursy	Wstępne kursy nie są wymagane.

### Efekty uczenia się

	Efekt uczenia się	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	<b>W01:</b> zna składnię języka Python oraz wybrane funkcje z biblioteki standardowej;	K_W03
	<b>W02:</b> ma wiedzę na temat instrukcji sterujących przepływem programu oraz tworzenia własnych funkcji w języku Python;	K_W03
	<b>W03:</b> zna podstawowe struktury danych (ze szczególnym uwzględnieniem typów niemutowalnych) języka Python i wie jak z nich korzystać w swoich programach pisanych w zgodzie z paradygmatem proceduralnym;	K_W03
	<b>W04:</b> zna podstawowe zasady programowania funkcyjnego.	K_W03

	Efekt uczenia się	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student:	
	<b>U01:</b> potrafi korzystać z podstawowych struktur danych i algorytmów tworząc proste programy w języku Python.	K_U04
	<b>U02:</b> potrafi korzystać z instrukcji sterujących funkcji wbudowanych i modułów oraz potrafi projektować własne funkcje na potrzeby programów pisanych w stylu funkcyjnym;	K_U04
	<b>U03:</b> potrafi zastosować funkcje wyższego rzędu w języku Python oraz umie wykorzystywać funkcje lambda, map(), filter(), reduce() i inne narzędzia wspierające programowanie funkcyjne;	K_U04
	<b>U04:</b> potrafi implementować algorytmy z wykorzystaniem rekurencji.	K_U04

	Efekt uczenia się	Odniesienie do efektów kierunkowych
	Po zakończeniu kursu student:	
Kompetencje społeczne	<b>K01:</b> potrafi korzystać z różnych źródeł informacji (zasobów sieci Internet) do poszerzania własnej wiedzy i zdobywania nowych umiejętności.	K_K02
	<b>K02:</b> potrafi przekazać wiedzę informatyczną w sposób zrozumiały dla innych.	K_K01

### Studia stacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	10					30					

### Studia niestacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	6					20					

### Opis metod prowadzenia zajęć

Podczas zajęć laboratoryjnych studenci rozwiązują zadania, analizują przykładowe programy oraz prezentują i omawiają rozwiązania zadań.

### Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01					x			x					
W02					x			x					
W03					x			x					
W04					x			x					
U01					x			x					
U02					x			x					
U03					x			x					
U04					x			x					
K01								x					
K02								x					

Kryteria oceny	<p>Osiągnięcie efektów kształcenia podanych powyżej uprawnia studentów do uzyskania oceny nie wyższej niż dostateczna.</p> <p>Ocenę dobrą lub bardzo dobrą może uzyskać student, który wykaże się dobrą lub bardzo dobrą znajomością języka Python oraz umiejętnością pisania programów z uwzględnieniem paradygmatu funkcyjnego.</p>
Uwagi	

#### Treści merytoryczne (wykaz tematów)

1. Wprowadzenie do pracy z językiem Python
  - a. środowisko programistyczne PyCharm
  - b. interpreter języka Python
  - c. struktura programu w języku Python
    - składnia i konwencje leksykalne
    - komentarze
    - zmienne i obiekty
  - d. obsługa standardowego wyjścia i wejścia
  - e. podstawowe funkcje wbudowane
2. Podstawy programowania proceduralnego w języku Python
  - a. podstawowe typy danych
    - całkowite rodzaje danych
    - zmiennoprzecinkowe rodzaje danych
    - ciągi tekstowe
  - b. kontenery wybudowane
    - krotki i listy
    - zbiory i słowniki
    - iteracja i kopiowanie kolekcji
  - c. struktury kontrolne i funkcje
    - konstrukcje rozgałęziające
    - pętle
    - własne funkcje
  - d. wybrane moduły i pakiety
3. Wprowadzenie do programowania funkcyjnego w języku Python:
  - a. historia i podstawowe koncepcje
  - b. programowanie imperatywne vs. programowanie funkcyjne
  - c. funkcje lambda, map(), filter(), reduce()
  - d. funkcje jako obiekty pierwszej klasy w języku Python
  - e. rekurencja i przykłady algorytmów rekurencyjnych
  - f. typy niemutowalne
  - g. listy składane i wyrażenia generatora
  - h. biblioteki wspierające programowanie funkcyjne w języku Python – biblioteka functools

#### Wykaz literatury podstawowej

Wskazane przez prowadzącego rozdziały:

1. E. Matthes, *Python. Instrukcje dla programisty. Wydanie III*, Helion, 2023;
2. S. F. Lott, *Programowanie funkcyjne w Pythonie. Jak pisać zwięzły, wydajny i ekspresywny kod. Wydanie III*, Helion, 2023;
3. A. Martelli, A. Martelli Ravenscroft, S. Holden, P. McGuire, *Python w pigułce. Podręczny przewodnik po wersjach 3.10 i 3.11*, Promise, 2023;
4. A. Sweigart, *Rekurencyjna książka o rekurencji. Zostań mistrzem rozmów kwalifikacyjnych poświęconych językom Python i JavaScript*, Helion, 2023.

## Wykaz literatury uzupełniającej

1. Ch. Mayer, *Kod Pythona w jednym wierszu. Jak profesjonaliści piszą programy doskonałe*, Helion 2021;
2. P. J. Deitel, H. Deitel, *Python dla programistów. Big Data i AI. Studia przypadków*, Helion, 2020;
3. D. Kopec, *Klasyczne problemy informatyki w Pythonie*, PWN, 2020;
4. M. Lutz, *Python. Leksykon kieszonkowy. Wydanie V*, Helion 2014;
5. M. Lutz, *Python. Wprowadzenie. Wydanie IV*, Helion 2010.

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia stacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	5
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	15
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do egzaminu	0
Ogółem bilans czasu pracy		60
Liczba punktów ECTS w zależności od przyjętego przelicznika		2

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia niestacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	6
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	4
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	30
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do egzaminu	0
Ogółem bilans czasu pracy		60
Liczba punktów ECTS w zależności od przyjętego przelicznika		2