



Elementy programowania  
Karta opisu przedmiotu

Informacje podstawowe

<p><b>Kierunek studiów</b> Geodezja i kartografia</p> <p><b>Specjalność</b> geodezja i geoinformatyka</p> <p><b>Jednostka organizacyjna</b> Wydział Inżynierii Kształtowania Środowiska i Geodezji</p> <p><b>Poziom studiów</b> studia pierwszego stopnia (inżynier)</p> <p><b>Forma studiów</b> stacjonarne</p> <p><b>Profil studiów</b> ogólnoakademicki</p>	<p><b>Cykl kształcenia</b> 2022/23</p> <p><b>Kod przedmiotu</b> ID000000IGIGFS.I4C.0612.22</p> <p><b>Języki wykładowe</b> polski</p> <p><b>Obligatoryjność</b> Obowiązkowy</p> <p><b>Blok zajęciowy</b> Przedmioty specjalnościowe</p> <p><b>Dyscypliny</b> Inżynieria lądowa i transport</p> <p><b>Przedmiot powiązany z badaniami naukowymi</b> Nie</p> <p><b>Przedmiot kształtujący umiejętności praktyczne</b> Nie</p>	
<p><b>Nauczyciel akademicki odpowiedzialny za przedmiot</b></p>	<p>Radosław Zajdel</p>	
<p><b>Pozostali prowadzący</b></p>	<p>Radosław Zajdel</p>	
<p><b>Okres</b> Semestr 3</p>	<p><b>Forma zaliczenia</b> Zaliczenie na ocenę</p> <p><b>Forma prowadzenia i godziny zajęć</b> Wykład: 15 Ćwiczenia laboratoryjne: 30</p>	<p><b>Liczba punktów ECTS</b> 4.0</p>

## Cele kształcenia dla przedmiotu

C1	<p>Kurs poświęcony jest rozwinięciu umiejętności programowania w języku Python. Uczestnicy kursu mają możliwość poznania różnych technik programowania, począwszy od zapoznania z paradygmatami programowania i podstaw ich realizacji w języku Python do omówienia najbardziej popularnych i użytecznych pakietów z punktu widzenia przetwarzania i analizy danych (Numpy, Pandas). Materiał kursu ukierunkowany jest na omówienie i ćwiczenie różnych technik programowania. Celem kursu jest praktyczne wykorzystanie zdobytych umiejętności do tworzenia narzędzi przydatnych w pracy geodety/geoinformatyka oraz zastosowaniach naukowych. Kurs przewiduje:</p> <ul style="list-style-type: none"> <li>• poznanie podstawowych elementów konstrukcyjnych języka Python, umożliwiających efektywne rozwiązywanie skomplikowanych zadań (np. iteratory, generatory list, funkcje lambda, typy sekwencyjne i iterowalne, słowniki, zbiory, instrukcje sterujące, funkcje),</li> <li>• efektywna praca w IDE Jupyter Notebook</li> <li>• znajomość podstawowych paradygmatów programowania (strukturalne, obiektowe, funkcyjne) w kontekście języka Python,</li> <li>• umiejętność korzystania z dodatkowych bibliotek programistycznych w celu dodawania nowych funkcjonalności do tworzonych aplikacji,</li> <li>• poznanie narzędzi pozwalających efektywnie analizować dane. Obliczenia na wektorach, macierzach z wykorzystaniem pakietu NumPy oraz praca na ramach danych w pakiecie Pandas. Zostanie także poruszony temat wizualizacji danych z wykorzystaniem biblioteki matplotlib.</li> </ul>
----	---

## Efekty uczenia się dla przedmiotu

Kod	Efekty uczenia się w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
<b>Wiedzy - Student zna i rozumie:</b>			
W1	Student posiada szczegółową wiedzę z na temat technik programowania, w szczególności w języku Python	GK_P6S_WG03	Zaliczenie pisemne, Aktywność na zajęciach
W2	Student zna teoretyczne aspekty paradygmatów programowania i ich praktycznej realizacji w języku Python	GK_P6S_WG03	Zaliczenie pisemne, Aktywność na zajęciach
<b>Umiejętności - Student potrafi:</b>			
U1	Student potrafi tworzyć skrypty oraz proste programy w języku Python, aby stosować je w praktyce geodezyjnej	GK_P6S_UW03	Kolokwium, Wykonanie ćwiczeń
U2	Student potrafi wykorzystywać zewnętrzne narzędzia i biblioteki programistyczne, a szczególnie biblioteki dedykowane dla geodetów i geoinformatyków, wspomagające wytwarzanie oprogramowania	GK_P6S_UW03	Kolokwium, Wykonanie ćwiczeń
U3	Student potrafi współdziałać i pracować w grupie, przyjmując w niej różne role	GK_P6S_UO19	Aktywność na zajęciach, Wykonanie ćwiczeń
<b>Kompetencje społecznych - Student jest gotów do:</b>			
K1	Student jest gotów do ciągłego doksztalcania się, potrafi inspirować i organizować uczenie innych osób podczas prowadzenia projektów	GK_P6S_KK01	Aktywność na zajęciach, Wykonanie ćwiczeń

## Bilans punktów ECTS

Forma aktywności studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	15
Ćwiczenia laboratoryjne	30

Przygotowanie do egzaminu/zaliczenia	20	
Przygotowanie do zajęć	20	
Przygotowanie raportu	20	
Konsultacje	3	
<b>Łączny nakład pracy studenta</b>	<b>Liczba godzin</b> 108	<b>ECTS</b> 4.0
<b>Zajęcia z bezpośrednim udziałem nauczyciela</b>	<b>Liczba godzin</b> 48	<b>ECTS</b> 1.9
<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>Liczba godzin</b> 50	<b>ECTS</b> 2.0

\* godzina (lekcyjna) oznacza 45 minut

## Treści programowe

Lp.	Treści programowe	Formy prowadzenia zajęć
1.	Wykład 1: Wprowadzenie teoretyczne, w tym: geneza, idea Open Source, porównanie z innymi językami, klasyfikacje języków programowania, definicja języków wysokiego poziomu, charakterystyka języka Python, różnice Python 2 vs. Python 3, zapoznanie z Jupyter Notebook, PEP-8, czytelność kodu Wykład 2: Podstawowe typy danych, sekwencyjne typy danych, mutable vs. immutable Wykład 3: Działania na sekwencjach, formatowanie tekstu, kontrola przepływu, Wykład 4: Typy mapujące, dokumentacja, funkcje, wyrażenia lambda Wykład 5: Moduły, obsługa błędów, działania na plikach Wykład 6: Programowanie zorientowane obiektowo Wykład 7: Działanie na macierzach, Numpy Wykład 8: Ramki danych, Pandas, Wizualizacja danych, Matplotlib	Wykład
2.	Ćwiczenia 1: Praktyczne wprowadzenie, w tym: zapoznanie z środowiskiem programowania, interpreterem i pomocnymi narzędziami, podstawowe komendy w kodzie Ćwiczenia 2-3: Właściwości typów danych i operacje na nich, rzutowanie typów, operacje arytmetyczne, definicje zmiennych, operatory porównań Ćwiczenia 3-4: Kontrola przepływu i działania na typach sekwencyjnych Ćwiczenia 5-6: Działania na słownikach, funkcje Ćwiczenia 7-8: Tworzenie modułów, Działania na plikach Ćwiczenia 9-10: Klasy i obiekty Ćwiczenia 11-12: Działania na Macierzach Ćwiczenia 13-14: Działania na Ramakach danych Ćwiczenia 15: Zaliczenie przedmiotu	Ćwiczenia laboratoryjne

## Informacje rozszerzone

### Metody nauczania:

Metoda problemowa, Praca w grupie, Pracownia komputerowa, Dyskusja, Wykład, Ćwiczenia, blended learning

<b>Aktywności</b>	<b>Metody zaliczenia</b>	<b>Udział procentowy w ocenie łącznej przedmiotu</b>
Wykład	Zaliczenie pisemne	30%
Ćwiczenia laboratoryjne	Aktywność na zajęciach, Kolokwium, Wykonanie ćwiczeń	70%

## **Wymagania wstępne**

Informatyka geodezyjna, Technologia informacyjna

## **Literatura**

### **Obowiązkowa**

1. Programming Python. Powerful Object-Oriented Programing. Edition IV; Lutz, M. (2010)
2. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. Edition II; McKinney, W. (2011)
3. Python documentation; <https://docs.python.org/3/>

### **Dodatkowa**

1. Clean Code: A Handbook of Agile Software Craftsmanship, Martin R. C., (2014)
2. Python. Rusz głową! Wydanie II, Barry P. (2017)
3. Automatyzacja nudnych zadań z Pythonem. Nauka. Sweigart A. (2017)

## Kierunkowe efekty uczenia się

Kod	Treść
GK_P6S_KK01	Absolwent jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów praktycznych i poznawczych związanych z zawodem geodety oraz zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemu, a także do krytycznej oceny posiadanej wiedzy i odbieranych treści.
GK_P6S_UO19	Absolwent potrafi planować i organizować pracę indywidualną oraz w zespole, a także współdziałać z innymi osobami w ramach prac zespołowych (także interdyscyplinarnych).
GK_P6S_UW03	Absolwent potrafi zaprojektować oraz zaimplementować w środowisku programistycznym własną aplikację wspomagającą realizację zadań geodezyjnych oraz posługiwać się narzędziami informatycznymi wykorzystywanymi w geodezji.
GK_P6S_WG03	Absolwent zna i rozumie w stopniu zawansowanym zagadnienia z zakresu systemów i sieci komputerowych oraz metod i technik programowania, niezbędne do instalacji, obsługi i wykorzystania narzędzi informatycznych stosowanych w geodezji.