


**KAPITAŁ LUDZKI**  
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez  
 Unię Europejską w ramach  
 Europejskiego Funduszu  
 Społecznego

**UNIA EUROPEJSKA**  
 EUROPEJSKI  
 FUNDUSZ SPOŁECZNY


<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Wprowadzenie do programowania		11.3.1761	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>drugiego stopnia</b>
Wydział Filologiczny	Filologia angielska	forma	stacjonarne
		moduł	przetwarzanie języka naturalnego
		specjalnościowy	
		specjalizacja	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
mgr Mateusz Miotk; dr Janusz Młodzianowski; dr Adam Kostulak			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		5	
Ćw. audytoryjne		60 godzin - uczestnictwo w zajęciach;	
<b>Sposób realizacji zajęć</b>		30 godzin - wykonanie zadań domowych;	
zajęcia on-line, zajęcia w sali dydaktycznej		20 godzin - wykonanie projektu programistycznego;	
<b>Liczba godzin</b>		15 godzin - przygotowanie do kolokwium.	
Ćw. audytoryjne: 60 godz.		Razem: 125 godzin = 5 ECTS.	
<b>Termin realizacji przedmiotu</b>			
2022/2023 letni			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
- fakultatywny (do wyboru)		- polski w wymiarze 90.00%	
- obowiązkowy		- angielski w wymiarze 10.00%	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
- Metoda projektów (projekt badawczy, wdrożeniowy, praktyczny)		<b>Sposób zaliczenia</b>	
- Rozwiązywanie zadań		Zaliczenie na ocenę	
- Wykonywanie doświadczeń		<b>Formy zaliczenia</b>	
		- wykonanie pracy zaliczeniowej - projekt lub prezentacja	
		- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru	
		- kolokwium	
		<b>Podstawowe kryteria oceny</b>	
		Ocenie podlega stopień opanowania i zrozumienie materiału, na który składają się następujące składowe:	
		<ul style="list-style-type: none"> <li>• Rozwiązywanie zadań laboratoryjnych 20%</li> <li>• Kolokwium 40%</li> <li>• Projekt programistyczny 40%</li> </ul>	
<b>Sposób weryfikacji założonych efektów uczenia się</b>			

zakładany efekt uczenia się	Wykonanie projektu	Rozwiązywanie zadań	Wykonywanie doświadczeń	Kolokwium
	Wiedza			
K_W01	X	X	X	X
K_W03	X	X	X	X
K_W08	X	X	X	X
K_W12	X	X	X	X
	Umiejętności			
K_U01	X	X	X	X
K_U02	X	X	X	X
K_U03	X	X	X	X
K_U07	X	X	X	X
	Kompetencje społeczne			
K_K01	X	X		
K_K02	X	X		
K_K03	X			

**Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi****A. Wymagania formalne**

brak

**B. Wymagania wstępne**

brak

**Cele kształcenia**

Celem przedmiotu jest zaznajomienie studentów z podstawowymi technikami i metodami programowania, na przykładzie języka programowania Python.

**Treści programowe**

- Pojęcie programu. Instalacja i wprowadzenie do środowiska języka Python.
- Podstawowe typy zmiennych.
- Pojęcie funkcji w programowaniu.
- Sterowanie programem za pomocą instrukcji warunkowych w programowaniu.
- Struktury cykliczne w języku Python: pętlę for oraz while.
- Zaawansowane struktury danych: listy, krotki, zbiory oraz słowniki.
- Operacje wejścia-wyjścia oraz na plikach w języku Python.
- Pojęcie algorytmu i schematu blokowego. Podstawowe algorytmy wyszukiwania oraz sortowania.
- Technika dziel i zwyciężaj w programowaniu.
- Podstawowe algorytmy tekstowe.
- Pojęcie rekurencji w programowaniu.
- Zarządzanie bibliotekami w języku Python.
- Wyrażenia regularne w języku Python.
- Obsługa plików tekstowych w języku Python.
- Format JSON oraz ich użycie w języku Python
- Zastosowanie bibliotek: numpy, nltk, matplotlib, seaborn, pygal.

**Wykaz literatury****Literatura podstawowa**

- P. Broda, D. Smołucha, Informatyka. Podręcznik część II, Wyd. Operon, 2006.
- M. Dawson, Python dla każdego. Podstawy programowania, Wyd. Helion, 2010.
- T. Gaddins, Python dla zupełnie początkujących, Wyd. Helion, 2019.
- E. Grubiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. M. Sysło, Informatyka część I, Wyd. WSiP, 2002.
- A. Sweigart, Automatyzacja nudnych zadań z Pythonem. Nauka programowania, Wyd. Helion, 2015.
- A. Szepietowski, Matematyka dyskretna, Wyd. UG 2004.
- P. Wróblewski, Algorytmy, struktury danych i techniki programowania, Wyd. Helion, 2010.
- <https://www.codecademy.com>
- <https://edabit.com>
- <https://exercism.io/>

- <https://www.hackerrank.com>
- <https://matplotlib.org/stable/index.html>
- <https://seaborn.pydata.org/>
- <http://www.pygal.org/en/stable/>

**Literatura uzupełniająca**

- T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, Wprowadzenie do algorytmów, Wyd. Naukowe PWN, 2012.
- P. Krugiołka, Linux. Jak dostroić bestię do swoich potrzeb, Wyd. Helion, 2012.
- E. Matthes, Python. Instrukcje dla programisty, Wyd. Helion, 2016.

**Kierunkowe efekty uczenia się**

K\_W01; K\_W03; K\_W08; K\_W12  
K\_U01; K\_U02; K\_U03; K\_U07  
K\_K01; K\_K02; K\_K03

**Wiedza**

Student:

K\_W01: zna w pogłębionym stopniu wybrane zagadnienia stanowiące zaawansowaną wiedzę ogólną właściwą dla informatyki oraz jej nauk pomocniczych i pokrewnych;

K\_W03: zna w pogłębionym stopniu w języku polskim i angielskim zaawansowaną terminologię z zakresu programowania, algorytmiki, w szczególności dla języka Python;

K\_W08: zna w pogłębionym stopniu powiązania programowania z innymi naukami humanistycznymi, społecznymi i innymi;

K\_W12: zna i rozumie zaawansowane metody wykorzystywania modułów w Pythonie w zakresie przetwarzania tekstów.

**Umiejętności**

Student:

K\_U01: potrafi właściwie korzystać ze zdobytej wiedzy przedmiotowej w celu formułowania i rozwiązywania problemów oraz innowacyjnego wykonywania zadań z zakresu programowania, dobierając i stosując właściwe metody i narzędzia;

K\_U02: potrafi właściwie dobierać źródła i czerpać z nich informacje z nich pochodzących w zakresie programowania w języku Python;

K\_U03: potrafi posługiwać się zaawansowaną terminologią specjalistyczną z zakresu programowania oraz informatyki;

K\_U07: potrafi samodzielnie planować i realizować własne uczenie się przez całe życie i ukierunkowywać innych w tym zakresie.

**Kompetencje społeczne (postawy)**

Student:

K\_K01: jest gotów do krytycznej oceny posiadanej wiedzy w zakresie informatyki, nauk pomocniczych i pokrewnych oraz uznaje znaczenie tej wiedzy w rozwiązywaniu problemów poznawczych i praktycznych;

K\_K01: rozumie konieczność ciągłego doksztalcania się i rozwoju;

K\_K02: jest gotów do samodzielnego podejmowania decyzji, krytycznej oceny działań własnych, działań zespołów, którymi kieruje, i organizacji, w których uczestniczy, przewodzenia grupie i ponoszenia odpowiedzialności za nią;

K\_K03: jest gotów do wypełniania zobowiązań społecznych wynikających z roli nauczyciela w tym do inspirowania i organizowania działalności na rzecz interesu publicznego i środowiska społecznego.

**Kontakt**

mateusz.miotk@ug.edu.pl