

1.	Nazwa kierunku	matematyka
2.	Wydział	Wydział Nauk Ścisłych i Technicznych
3.	Cykl rozpoczęcia	2019/2020 (semestr zimowy)
4.	Poziom kształcenia	studia pierwszego stopnia
5.	Profil kształcenia	ogólnoakademicki
6.	Forma prowadzenia studiów	stacjonarna

**Moduł kształcenia:** Algorytmy i programowanie

**Kod modułu:** 03-MO1S-19-AiP

**1. Liczba punktów ECTS:** 6

2. Zakładane efekty uczenia się modułu			
kod	opis	efekty uczenia się kierunku	stopień realizacji (skala 1-5)
AiP_1	zna podstawy wybranego języka programowania wyższego rzędu; projektuje, programuje i testuje programy w procesie rozwiązywania problemów; w programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów oraz zmienne i tablice, rekurencje	K_U26 K_U27 K_W08 NI_U04 NI_W04	5 5 5 5 5
AiP_10	zna różne metody i techniki programowania: podejście zachłanne, programowanie dynamiczne	K_W08 NI_U03 NI_W03	3 3 3
AiP_11	posiada umiejętność oceny ograniczeń narzędzi komputerowych	K_W08	5
AiP_2	testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów	K_U26 NI_U04	5 5
AiP_3	formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów; zna pojęcie algorytmu i stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków, w pseudokodzie oraz w wybranym języku programowania	K_U25 K_U26 K_W08 NI_W03	5 5 5 5
AiP_4	zna i zapisuje klasyczne algorytmy za pomocą listy kroków, schematu blokowego lub pseudokodu oraz implementuje je wybranym języku programowania; zna i omawia sytuacje, w których wykorzystuje się klasyczne algorytmy	K_W08 NI_U03	5 5

		NI_U04	5
		NI_W03	5
AiP_5	zna podstawowe własności algorytmów; prezentuje przykłady zastosowań algorytmiki w innych dziedzinach nauki	K_W08	3
		NI_U03	3
		NI_W03	3
AiP_6	rozwija znajomość algorytmów i wykonuje eksperymenty z algorytmami; rozumie potrzebę programowania z użyciem zaawansowanych algorytmów	K_U25	3
		K_U27	3
		NI_K02	3
		NI_U03	3
		NI_W03	3
AiP_7	zna i rozumie pojęcie złożoności obliczeniowej (czasowej i pamięciowej) oraz notacji asymptotycznej	K_W08	4
		NI_U03	3
		NI_W03	4
AiP_8	zapisuje wybrane algorytmy klasyczne w postaci iteracyjnej oraz rekurencyjnej	K_U26	5
		NI_U03	5
		NI_U04	5
		NI_W03	5
AiP_9	porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji	K_U26	4
		K_U27	4
		NI_U03	4
		NI_W03	4

### 3. Opis modułu

<b>Opis</b>	<p>1. Wprowadzenie do języków programowania; podział języków programowania; sposób wykonywania (kompilacja a interpretacja).</p> <p>2. Podstawy programowania w języku Python. Instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne oraz zmienne i tablice (listy).</p> <p>3. Podprogramy w języku Python - funkcje i procedury.</p> <p>4. Elementy algorytmiki: problem i jego specyfikacja; algorytm i różne sposoby jego zapisu.</p> <p>5. Elementy analizy algorytmów. Rozmiar danych, złożoność obliczeniowa (czasowa i pamięciowa). Typy złożoności: pesymistyczna, optymistyczna, średnia. Notacja asymptotyczna, rzędy wielkości funkcji.</p> <p>6. Algorytmy iteracyjne i rekurencyjne; metoda dziel i zwyciężaj.</p> <p>7. Algorytmy klasyczne w tym m.in.:</p> <ul style="list-style-type: none"> <li>- obliczania wartości wielomianu za pomocą schematu Hornera,</li> <li>- algorytmy Euklidesa w wersji iteracyjnej i rekurencyjnej wraz z zastosowaniami,</li> <li>- operujące na liczbach (badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW),</li> <li>- operujące na tekstach (porównywanie tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową),</li> <li>- wyszukiwania elementów w dowolnej tablicy (algorytm sekwencyjny) oraz w tablicy uporządkowanej (metoda wyszukiwania binarnego)</li> </ul>
-------------	---

	<ul style="list-style-type: none"> <li>- sortujące (sortowanie przez wstawianie, przez wybieranie, bąbelkowe, przez scalanie, szybkie),</li> <li>- znajdowania określonego elementu w zbiorze: maksymalnego, lidera oraz idola,</li> <li>- generowania liczb pierwszych metodą sita Eratostenesa,</li> <li>- jednoczesnego wyszukiwania elementu najmniejszego i największego (algorytm iteracyjny oraz rekurencyjny wykorzystujący metodę dziel i zwyciężaj),</li> <li>- szybkiego potęgowania liczb w wersji iteracyjnej i rekurencyjnej.</li> </ul> <p>8. Różne metody i techniki programowania:</p> <ul style="list-style-type: none"> <li>- podejście zachłanne (wydawania reszty najmniejszą liczbą nominałów, pakowanie plecaka),</li> <li>- programowanie dynamiczne (pakowanie plecaka, szukania najdłuższego wspólnego podciągu).</li> </ul> <p>9. Implementacja poznanych algorytmów w języku Python.</p>
<b>Wymagania wstępne</b>	

<b>4. Sposoby weryfikacji efektów uczenia się modułu</b>			
<b>kod</b>	<b>nazwa (typ)</b>	<b>opis</b>	<b>efekty uczenia się modułu</b>
AiP_w_1	kolokwium na konwersatorium	Kolokwium pisemne na ostatnich lub przedostatnich zajęciach; zadania podobnego typu do zadań rozwiązywanych w trakcie zajęć konwersatoryjnych	AiP_10, AiP_3, AiP_4, AiP_7, AiP_8, AiP_9
AiP_w_2	kolokwia na laboratorium	Dwa kolokwia w semestrze; zadania podobnego typu do zadań rozwiązywanych w trakcie zajęć laboratoryjnych	AiP_1, AiP_2, AiP_4, AiP_8, AiP_9
AiP_w_3	zadania domowe	ocena zadań domowych; możliwość odpytania z wybranych zagadnień/zadań zadanych na pracę w domu	AiP_1, AiP_10, AiP_11, AiP_2, AiP_3, AiP_4, AiP_5, AiP_6, AiP_7, AiP_8, AiP_9
AiP_w_4	egzamin	Warunkiem przystąpienia do egzaminu jest zaliczenie konwersatorium oraz laboratorium; weryfikacja znajomości pojęć i faktów w oparciu o analizę odpowiedzi na pytania egzaminacyjne	AiP_1, AiP_10, AiP_4, AiP_7, AiP_8, AiP_9

<b>5. Rodzaje prowadzonych zajęć</b>						
<b>kod</b>	<b>rodzaj prowadzonych zajęć</b>			<b>praca własna studenta</b>		<b>sposoby weryfikacji efektów uczenia się</b>
	<b>nazwa</b>	<b>opis (z uwzględnieniem metod dydaktycznych)</b>	<b>liczba godzin</b>	<b>opis</b>	<b>liczba godzin</b>	
AiP_fs_1	wykład	wykład z wykorzystaniem pomocy audiowizualnych	30	przyswojenie wiadomości z wykładu przy pomocy udostępnionych materiałów wykładowych; lektura uzupełniająca podręczników;	20	AiP_w_4
AiP_fs_2	laboratorium	praca w laboratorium z wykorzystaniem komputera w oparciu o otwarte środowiska programistyczne	30	praca własna z wykorzystaniem ogólnodostępnego oprogramowania, doskonalenie umiejętności zdobytych podczas zajęć	50	AiP_w_2, AiP_w_3
AiP_fs_3	konwersatorium	konwersatorium, w trakcie którego studenci rozwiązują, pod kierunkiem prowadzącego, zadania kształtujące umiejętności wymienione w zestawie efektów kształcenia modułu	15	przyswojenie wiedzy z wykładów, samodzielna praca ze zbiorami zadań,	25	AiP_w_1, AiP_w_3