

1.	Nazwa kierunku	informatyka
2.	Wydział	Wydział Nauk Ścisłych i Technicznych
3.	Cykl rozpoczęcia	2020/2021 (semestr letni), 2021/2022 (semestr zimowy)
4.	Poziom kształcenia	studia drugiego stopnia
5.	Profil kształcenia	ogólnoakademicki
6.	Forma prowadzenia studiów	niestacjonarna

Moduł kształcenia: Koncepcja języków programowania

Kod modułu: W4-IN-N2-20-F-KJP

1. Liczba punktów ECTS: 4

2. Zakładane efekty uczenia się modułu			
kod	opis	efekty uczenia się kierunku	stopień realizacji (skala 1-5)
M_001	Zna paradygmaty programowania: programowanie proceduralne, programowanie obiektowe, programowanie strukturalne, współbieżne, programowanie imperatywne, funkcyjne i deklarycyjne oraz ich powiązanie z architekturą komputerów (w tym równoległych i wieloprotokolowych)	K_W02 K_W04	1 1
M_002	Rozumie podstawowe konstrukcje programistyczne oraz zna typy danych języków imperatywnych oraz konstrukcje programistyczne charakterystyczne dla podejścia deklarycyjnego i funkcyjnego	K_W04 K_W05	1 1
M_003	Ma wiedzę dotyczącą implementacji mechanizmów charakterystycznych dla konkretnego paradygmatu programowania w wybranych językach programowania	K_W02 K_W04 K_W08 K_W09	1 1 1 1
M_004	Potrafi skonstruować rozwiązanie podanego problemu zgodnie z określonym paradygmatem programowania i zapisać go w wybranym języku programowania	K_U01 K_U02 K_U03	1 1 1
M_005	Potrafi stosować podejście obiektowe, strukturalne, funkcyjne i deklarytywne w wybranych językach programowania	K_U01 K_U02 K_U03	1 1 1
M_006	Potrafi sprawdzić niezawodność programu komputerowego za pomocą testowania w wybranym środowisku programistycznym i udokumentować program	K_U01 K_U02 K_U04	1 1 1

		K_U09	1
M_007	Student potrafi pracować w zespole projektowo-programistycznym	K_K01	1
		K_K03	1
		K_K05	1

3. Opis modułu	
Opis	Celem zajęć jest uzupełnienie wiedzy studentów dotyczącej zasad projektowania i implementowania programów komputerowych w wybranych językach reprezentujących podejście imperatywne, funkcyjne i deklaratywne. Studenci rozwijają swoją wiedzę i umiejętności stosowania różnych paradygmatów programowania, poznając koncepcje języków programowania charakterystycznych dla tych paradygmatów.
Wymagania wstępne	

4. Sposoby weryfikacji efektów uczenia się modułu			
kod	nazwa (typ)	opis	efekty uczenia się modułu
W_001	Ocena prezentacji i implementacji komputerowej	Studenci wykonują samodzielnie oprogramowanie, którego specyfikacja jest podawana przez prowadzącego oraz prezentacje na wybrany z zestawu temat	M_001, M_002, M_003, M_004, M_006, M_007
W_002	Praca kontrolna	Kolokwium pisemne (w tym test wykonany na komputerze w czasie zajęć)	M_001, M_002, M_003, M_005
W_003	Zaliczenie końcowe	Studenci odpowiadają na pytania testowe oraz opisują zagadnienia w odpowiedzi na pytania otwarte	M_001, M_002, M_003, M_004, M_005, M_006

5. Rodzaje prowadzonych zajęć						
kod	rodzaj prowadzonych zajęć			praca własna studenta		sposoby weryfikacji efektów uczenia się
	nazwa	opis (z uwzględnieniem metod dydaktycznych)	liczba godzin	opis	liczba godzin	
Z_001	wykład	Podanie treści kształcenia w formie werbalnej z wykorzystaniem wizualizacji treści. Skupienie się na materiale trudnym pojęciowo i wskazanie źródeł. Ilustracja treści za pomocą przykładów.	15	Zapoznanie się z tematyką wykładu z wykorzystaniem istniejących pakietów metod: podręczników, skryptów, stron internetowych itp.	30	W_003
Z_002	laboratorium	Szczegółowe przygotowanie studentów do rozwiązywania zadań ze wskazaniem na metodologię postępowania, wskazaniem kolejności wykonywanych czynności. Projektowanie rozwiązań i ich implementacja komputerowa. Prezentowanie przez studentów swoich rozwiązań.	30	Rozwiązywanie zadań z poszczególnych tematów wraz z analizą rozwiązań już istniejących – w materiałach i na stronach internetowych. Przygotowanie zagadnień do przedyskutowania lub przygotowanie się do nadrobienia zaległości	45	W_001, W_002