

1.	Nazwa kierunku	informatyka
2.	Wydział	Wydział Nauk Ścisłych i Technicznych
3.	Cykl rozpoczęcia	2019/2020 (semestr zimowy), 2020/2021 (semestr zimowy)
4.	Poziom kształcenia	studia pierwszego stopnia (inżynierskie)
5.	Profil kształcenia	ogólnoakademicki
6.	Forma prowadzenia studiów	stacjonarna

Moduł kształcenia: Programowanie współbieżne

Kod modułu: 08-IO1S-13-PWSP

1. Liczba punktów ECTS: 5

2. Zakładane efekty uczenia się modułu			
kod	opis	efekty uczenia się kierunku	stopień realizacji (skala 1-5)
PWSP_K_9	Potrafi pracować w zespole dwuosobowym i dokonuje właściwego podziału pracy		
PWSP_U_6	Potrafi dokonać dekompozycji problemu obliczeniowego na potrzeby obliczeń równoległych	K_U12	1
PWSP_U_7	Potrafi zaimplementować program współbieżny i zastosować podstawowe mechanizmy współbieżności	K_U12 K_U13 K_U14 K_U15 K_U16	1 1 1 1 1
PWSP_U_8	Potrafi zaimplementować program równoległy dla komputerów ze współdzieloną pamięcią, w tym również z użyciem GPU	K_U12 K_U13	1 1
PWSP_W_1	Ma podstawową wiedzę na temat architektury współczesnych komputerów i jej znaczenia dla programowania równoległego	K_K01 K_K05 K_W06	1 1 1
PWSP_W_2	Ma podstawową wiedzę na temat modeli obliczeń równoległych	K_W09	3
PWSP_W_3	Ma wiedzę na temat sposobów oceny efektywności algorytmów równoległych	K_W04 K_W09	1 1
PWSP_W_4	Ma wiedzę na temat podstawowych algorytmów równoległych i ich zastosowań	K_W09	1
PWSP_W_5	Ma wiedzę na temat architektury GPU i ich zastosowań w obliczeniach ogólnego przeznaczenia	K_W09	1

3. Opis modułu

Opis	Celem zajęć jest przygotowanie studentów do projektowania i implementacji poprawnych algorytmów współbieżnych. W trakcie zajęć studenci uzyskują wiedzę na temat problemów związanych z projektowaniem programów współbieżnych i zaawansowanych mechanizmów ich rozwiązywania. Dodatkowo studenci zdobędą niezbędną wiedzę dotyczącą podstaw projektowania efektywnych algorytmów równoległych.
Wymagania wstępne	

4. Sposoby weryfikacji efektów uczenia się modułu

kod	nazwa (typ)	opis	efekty uczenia się modułu
PWSP_w_1	Prace kontrolne	Co najmniej jedno kolokwium sprawdzające wiedzę z tematyki poruszanej na wykładzie oraz realizowanej w ramach laboratoriów.	PWSP_U_6, PWSP_U_7, PWSP_U_8, PWSP_W_1, PWSP_W_2, PWSP_W_3, PWSP_W_4, PWSP_W_5
PWSP_w_2	Projekt programistyczny	Realizacja projektu / projektów programistycznych pozwalających na praktyczną weryfikację nabytej wiedzy i umiejętności dotyczących programowania równoległego	PWSP_K_9, PWSP_U_6, PWSP_U_7, PWSP_W_5
PWSP_w_3	Egzamin	Test złożony z pytań wielokrotnego wyboru oraz zadań otwartych dotyczących tematów poruszanych na wykładach oraz laboratoriach.	PWSP_U_6, PWSP_U_7, PWSP_W_1, PWSP_W_2, PWSP_W_3, PWSP_W_4, PWSP_W_5

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	
PWSP_fs1	wykład	Podanie treści kształcenia w formie werbalnej z wykorzystaniem wizualizacji treści. Omówienie ważnych kwestii teoretycznych i praktycznych dotyczących programowania współbieżnego.	30	Zapoznanie się z tematyką prezentowaną podczas wykładów oraz przygotowanie się do laboratoriów związanych z wykładami	30	PWSP_w_3
PWSP_fs2	laboratorium	Szczegółowe przygotowanie studentów do tworzenia aplikacji współbieżnych w nowoczesnych językach programowania. Prezentacja i omówienie narzędzi wspierających realizację oprogramowania współbieżnego.	30	Rozwiązywanie zadań praktycznych z poszczególnych tematów wraz z analizą rozwiązań już istniejących. Zrealizowanie projektu programistycznego z zastosowaniem prezentowanych na wykładach metod.	60	PWSP_w_1, PWSP_w_2