

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie współbieżne i równoległe, PG_00060227						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	październik 2023 r.		Rok akademicki realizacji przedmiotu		2025/2026		
Poziom kształcenia	I stopnia - inżynierskie		Grupa zajęć		Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	3		Język wykładowy		polski		
Semestr studiów	5		Liczba punktów ECTS		3.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Fizyki Technicznej i Matematyki Stosowanej -> Instytut Fizyki i Informatyki Stosowanej -> Zakład Metod Obliczeniowych Fizyki Chemicznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. Jan Franz				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	30.0	0.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
	Adresy kursu na platformie eNauczanie: Moodle ID: 1308 Programowanie współbieżne i równoległe https://enauczanie.pg.edu.pl/2025/course/view.php?id=1308						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45		5.0		25.0	75
Cel przedmiotu	Celem kursu jest wprowadzenie do zasad i technik programowania współbieżnego i równoległego w języku Java, ze szczególnym uwzględnieniem ich zastosowania w prostych problemach fizycznych. Studenci uczą się projektować, implementować i analizować programy wielowątkowe, rozumieją klasyczne problemy synchronizacji i wzorce projektowe oraz potrafią oceniać wydajność i skalowalność rozwiązań równoległych.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K6_K01] Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych. Potrafi inspirować i organizować proces uczenia się innych osób.		Student zapoznaje się ze scenariuszami, w których wykorzystanie technik i metod programowania współbieżnego lub równoległego jest niezbędne do rozwiązania problemu.		[SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce		
	[K6_W05] Posiada wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.		Student poznaje metody rozwiązania klasycznych problemów współbieżności.		[SW1] Ocena wiedzy faktograficznej		
	[K6_U03] Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.		Student ćwiczy programowanie współbieżne z wykorzystaniem języka programowania Java.		[SU1] Ocena realizacji zadania		

Treści przedmiotu	1. Wprowadzenie i motywacja Wykład: Wprowadzenie do współbieżności, motywacja z symulacji fizycznych (wiele cząstek, wiele zdarzeń), przegląd wątków i procesów. Laboratorium: Utworzenie i uruchomienie prostych wątków, symulacja równoległych rzutów swobodnych. 2. Cykl życia wątku Wykład: Stany wątku, tworzenie i kończenie, metody sterowania (start, join, sleep, interrupt), koszty wątków. Laboratorium: Implementacja równoległych oscylatorów harmoniczných, porównanie czasu działania z wykonaniem sekwencyjnym. 3. Podstawy synchronizacji Wykład: Warunki wyścigu, sekcje krytyczne, podstawowa synchronizacja za pomocą synchronized. Laboratorium: Symulacja rozpadu promieniotwórczego wielu jąder, bezpieczna aktualizacja wspólnego licznika. 4. Skalowanie wydajności Wykład: Prawo Amdahla i prawo Gustafsona, przyspieszenie równoległe, efektywność w obliczeniach fizycznych. Laboratorium: Estymacja liczby π metodą Monte Carlo przy różnych liczbach wątków, porównanie przyspieszenia z teorią. 5. Blokady i warunki Wykład: Obiekty blokad, ReentrantLock, zmienne warunkowe dla kontrolowanego dostępu. Laboratorium: Symulacja błędzenia losowego na linii, koordynacja aktualizacji za pomocą blokad i warunków. 6. Wzorzec producent-konsument Wykład: Klasyczny problem producent-konsument, kolejki blokujące, sprawiedliwość, unikanie przepełnienia i pustki. Laboratorium: Implementacja systemu detektor-analizator, gdzie producenci generują zdarzenia cząstek, a konsumenci je analizują. 7. Studium przypadku I: równoległa analiza danych Wykład: Równoległe zbieranie i przetwarzanie danych, strategie synchronizacji w analizie danych fizycznych. Laboratorium: Implementacja równoległego tworzenia histogramów energii cząstek przy użyciu kolekcji bezpiecznych dla wątków. 8. Wzorzec czytelnik-pisarz Wykład: Problem czytelnik-pisarz, zarządzanie równoczesnym dostępem odczytu i zapisu, sprawiedliwość i zagłódzenie. Laboratorium: Implementacja wspólnego zbioru danych pomiarowych z równoczesnym odczytem i okazjonalnym zapisem. 9. Cykliczna alokacja zasobów Wykład: Abstrakcyjna formuła problemu cyklicznej alokacji zasobów, zakleszczenie, zakleszczenie pozorne, zagłódzenie oraz strategie rozwiązywania. Laboratorium: Implementacja systemu wielu procesów współbieżnych wymagających dwóch zasobów, demonstracja zakleszczenia i zastosowanie strategii unikania. 10. Pula wątków i egzekutory Wykład: Ramy Executor, pule wątków (stałe i dynamiczne), zadania, Callable, Future. Laboratorium: Symulacja wielu błędzeń losowych przy użyciu puli wątków, zbieranie wyników przez futures. 11. Zaawansowane narzędzia synchronizacji Wykład: Semaforey, bariery, fazowe koordynowanie zadań współbieżnych. Laboratorium: Synchronizacja oscylatorów harmoniczných w krokach czasowych przy użyciu barier, symulacja falowej ewolucji kolektywnej. 12. Model ForkJoin Wykład: Algorytmy dziel i zwyciężaj, zadania ForkJoin, rekursywna dekompozycja problemów. Laboratorium: Implementacja sumowania dużego zbioru danych metodą ForkJoin, np. zliczanie cząstek w symulowanych eksperymentach. 13. Strumienie i tablice równoległe Wykład: API strumieni, operacje równoległe, redukcje i agregacje. Laboratorium: Obliczanie energii kinetycznych wielu cząstek w strumieniach równoległych, redukcja wyników do średnich i maksimów. 14. Studium przypadku II: równoległe symulacje fizyczne Wykład: Przykłady większych symulacji równoległych: dyfuzja, model Isinga, rozpraszanie. Laboratorium: Implementacja równoległej symulacji dyfuzji 2D cząstek, analiza skalowania wydajności. 15. Podsumowanie i perspektywy Wykład: Przegląd pojęć współbieżności, wzorców projektowych i lekcji o wydajności; perspektywy rozwoju w HPC i klastrach. Laboratorium: Zadanie integracyjne: symulacja rozpadu promieniotwórczego z tworzeniem histogramu i analizą wydajności równoległej względem sekwencyjnej.		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Laboratorium: ćwiczenia programistyczne	50.0%	75.0%
	Wykład: egzamin	50.0%	25.0%

Zalecana lista lektur	Podstawowa lista lektur	1. A. Karbowski, E. Niewiadomska-Szynkiewicz, Programowanie równoległe i rozproszone : praca zbiorowa, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2009.
	Uzupełniająca lista lektur	1. P. Majdzik, "Programowanie współbieżne. Systemy czasu rzeczywistego", PWN, Warszawa 2013 2. Z. J. Czech, "Wprowadzenie do obliczeń równoległych", PWN, Warszawa, 2013 3. Z. Weiss, T. Gruzlewski, Programowanie współbieżne i rozproszone w przykładach i
	Adresy eZasobów	Podstawowe https://docs.oracle.com/javase/tutorial/essential/concurrency/ - Oficjalny samouczek Oracle dotyczący współbieżności w Javie. Zawiera wprowadzenie do podstaw obsługi wątków (tworzenie, synchronizacja, blokady), a także przegląd narzędzi wyższego poziomu z pakietu java.util.concurrent (egzekutory, obiekty blokad, kolekcje współbieżne, model Fork/Join, zmienne atomowe). Omawia również typowe problemy, takie jak zakłócenia między wątkami, błędy spójności pamięci, zakleszczenia i blokady warunkowe.
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	1. Oblicz przyspieszenie programu współbieżnego. 2. Konwertuj program sekwencyjny na program współbieżny	
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.