

## Karta zaj

Informacje ogólne			
Nazwa zaj : <b>Programowanie współbieżne i rozproszone</b>			
Nazwa uczelni: <b>Wyższa Szkoła Zarządzania i Bankowości w Krakowie</b>			
Wydział: <b>Wydział Nauk Stosowanych</b>			
Kierunek studiów: <b>Informatyka</b>			
Poziom studiów: pierwszego stopnia			
Forma studiów: niestacjonarne, stacjonarne		Profil kształcenia: praktyczny	Zakres kształcenia:
Rok/Semestr: 3/5		Status zaj : obowiązkowy	Języki wykładowe: polski
Studia niestacjonarne	Forma zaj	wykłady	wiczenia laboratoryjne
	Wymiar zaj (w godz.)	16	16
Studia stacjonarne	Forma zaj	wykłady	wiczenia laboratoryjne
	Wymiar zaj (w godz.)	30	30
Koordynator zaj		dr Włodzimierz Funika	
Prowadzący		dr Włodzimierz Funika	
Cele kształcenia		C1. Zapoznanie się z problematyką obliczeń wielkiej skali (High Performance Computing) C2. Poznanie metodologii konstruowania programów równoległych C3. Zapoznanie się ze standardem Message Passing Interface (MPI) C4. Poznanie zasad programowania w architekturze klient-serwer C5. Zapoznanie się i zrozumienie pojęć u podstaw teorii współbieżności C6. Zdobycie podstawowej wiedzy z zakresu programowania współbieżnego w Javie C7. Zdobycie podstawowej wiedzy z zakresu konstruowania aplikacji rozproszonych	
Wymagania wstępne		Języki i techniki programowania, umiejętność programowania w języku C, Java	

Efekty uczenia się			Odniesienie do efektów uczenia się dla kierunku	Odniesienie do charakterystyk PRK poziomu 6
Wiedza	EU1	Rozumie problemy obliczeniowej skali oraz podejścia do ich rozwiązywania oparte na współczesnych architekturach komputerowych.	K_W07	P6U_W P6S_WG
	EU2	Ma teoretyczną wiedzę o podstawowych zasadach konstruowania programów współbieżnych i rozproszonych.	K_W04	P6U_W P6S_WG
Umiejętności	EU3	Potrafi tworzyć efektywne aplikacje pracujące w rozproszonych systemach komputerowych używając właściwych narzędzi programistycznych.	K_U08 K_U09	P6U_U P6S_UW
Kompetencje społeczne	EU4	Student potrafi pracować w grupie uytakowników dla wypracowania podejścia do rozwiązania problemu rozproszonego lub współbieżnego.	K_K03 K_K04	P6U_U P6S_KO

## Treści programowe

Wykład	
W1	Wstęp: Rola obliczeń rozproszonych i współbieżnych w rozwoju cywilizacji. Charakterystyka programowania równoległego i rozproszonego. Krótkie przedstawienie i porównania paradygmatów komunikacji w programowaniu równoległym i rozproszonym. Paradygmat przekazywania wiadomości w programowaniu równoległym. Określenie obszaru problemów w jakim przydatny jest paradygmat przekazywania wiadomości (message passing). Powiązanie omawianego sposobu programowania z cechami platformy sprzętowej. Wstępne omówienie dostępnych technik.
W2	Message Passing Interface: Omówienie standardu MPI. Podstawowe mechanizmy udostępniane przez specyfikację. Różne sposoby przekazywania wiadomości wraz z omówieniem ich przydatności w różnych sytuacjach.
W3	Message Passing Interface: Omówienie komunikacji kolektywnej oraz dodatkowych mechanizmów dostępnych w MPI tj. typy złożone, wirtualne topologie. Rozszerzenia dostępne w wersji 2 specyfikacji MPI: dynamiczne procesy, "jednostronna komunikacja", równoległe operacje wejścia-wyjścia.
W4	Podstawowe pojęcia przetwarzania współbieżnego: Definicja pojęć takich jak programowanie współbieżne, system równoległy, system rozproszony. Wprowadzenie pojęcia procesu, akcji, procesów współbieżnych, komunikacji, synchronizacji. Sposoby opisu współbieżności. Język Erlang, Scala. Klasyczne problemy współbieżności
W5	Problem wzajemnego wykluczania i jego rozwiązania: Definicja problemu wzajemnego wykluczania. Przykłady zapotrzebowania na sterowanie współbieżnością, współpraca. Sekcja krytyczna. Pojęcie wywołania, bezpieczeństwo, sprawiedliwość. Blokady, zagłodzenie. Rozwiązania: zakaz przerwy, specjalne rozkazy. Rozwiązania programowe. Semafor binarny, uogólniony. Zmienne warunkowe. Monitor.
W6	Mechanizmy współbieżności w Javie: Procesy a wątki, zarządzanie procesami. Wątki Javy. Etapy tworzenia wątków. Porównanie wątków Javy a procesów UNIX-a. Wzajemne wykluczanie a zamki. Rozwiązanie w Javie. Implementacja zamków: zamek płoty a zamek blokujący. Monitor w Javie. Implementacja semaforów.
W7	Mechanizmy współbieżności w Javie: Zarządzanie współbieżnością: współpraca. Warunkowa synchronizacja. Semantyka wait/notify. Porównanie rodków zarządzania współbieżnością w Javie a semaforów. Mechanizmy współbieżności w Unixie: potoki, pamięć wspólna, potoki, semafony, kolejki komunikatów.
W8	Algorytmy równoległe: Podstawowe wymagania stawiane programom równoległym: współbieżność, skalowalność, lokalność danych, modularność. Model zadania-kanaly tworzenia programów równoległych.
W9	Algorytmy równoległe: Podstawowe typy algorytmów równoległych, przykłady. Metoda PCAM -- partition, communication, agglomeration, mapping.
W10	Algorytmy równoległe: Analiza jakości działania programów równoległych: przepięszenie, efektywność, funkcja izoefektywności.
W11	Algorytmy równoległe: Przykłady konstruowania algorytmów równoległych.
W12	Programowanie zorientowane na usługi: Paradygmat programowania orientowanego serwisowo. Programowanie z wykorzystaniem Java RMI.
W13	Programowanie zorientowane na usługi: Programowanie w wykorzystaniu serwisów webowych. Wstęp do XML, UDDI i SOAP.
W14	Programowanie zorientowane na usługi: Język opisu serwisów webowych - WSDL. Rejestr dostępnych serwisów. Przykładowe problemy rozwiązane w opisywanym paradygmacie.
W15	Systemy rozproszone – zasady konstruowania: Przezroczystość, skalowalność, komunikacja, synchronizacja zegarów. Problematyka Big Data.
Laboratorium	
L1	Wprowadzenie w koncepcję programowania równoległego i środowisko programowania z przesyłaniem komunikatów. Przygotowanie, kompilacja i uruchomienie prostego programu równoległego w standardzie MPI.
L2	Komunikacja w MPI. Poznawanie trybów wymiany komunikatów w aplikacjach MPI. Implementacja programu równoległego przesyłającego proste typy danych.
L3	Operacje globalne. Implementacja aplikacji w standardzie MPI rozwiązującej konkretny problem z dziedziny algebry macierzy przy pomocy funkcji MPI dokonujących operacji globalnych.
L4	Projekt aplikacji rozproszonej o architekturze typu klient-serwer (w oparciu o model RPC). Zapoznanie się z technologi RMI jako realizacją modelu RPC dla języka obiektowego (na przykładzie analizy prostej aplikacji rozproszonej)
L5	Rozszerzenie wiedzy o RMI poprzez stopniowe modyfikowanie przykładowej aplikacji (mechanizmy dynamicznego pobierania klas poprzez siebie, interfejs Serializable, moduł RMI Activation)

L6	Samodzielna implementacja pełnego systemu opartego na RMI.
L7	Zapoznanie się z usługami sieciowymi. Implementacja klienta usługi sieciowej w Javie i w Perlu.
L8	Podstawy programowania wielowątkowego w Javie. Problem wyścigu; implementacja programu demonstrującego ten problem.
L9	Sposoby synchronizacji wątków w Javie; rozwiązanie problemu wyścigu. Pojęcie semafora. Implementacja klasy semafora binarnego i ogólnego.
L10	Oczekiwanie warunkowe jako sposób koordynacji działań między wątkami. Realizacja oczekiwania warunkowego w Javie. Problem producentów-konsumentów i jego implementacja w Javie.
L11	Programowanie współbieżne w języku Scala. Rozwiązanie problemu producentów-konsumentów w Scali.
L12	Zapoznanie się z językiem opisu serwisów webowych - WSDL. Implementacja klienta w Javie oraz w Perlu do wybranych istniejących w Internecie serwisów.
L13	Zapoznanie się z konfiguracją kontenera Web Serwisów. Implementacja prostego serwisu webowego oraz prostego klienta sprawdzającego jego funkcjonalność.
L14	Wykorzystanie UDDI jako rejestru Web Serwisów. Ćwiczenia mające na celu opanowanie publikowania danych w UDDI i wyszukiwanie w nim informacji.
L15	Oddawanie zaległych zadań, wystawianie zaliczeń.

Ocena studenta			
Metody/Narzędzia dydaktyczne	N1	prezentacja multimedialna	wykład laboratorium
	N2	wykonanie ćwiczeń laboratoryjnych	laboratorium
	N3	praca w grupach	laboratorium
Sposoby oceny/metody weryfikacji uczenia się	Ocena formująca		
	F1	Ocena ćwiczeń laboratoryjnych	laboratorium
	F2	Ocena z kolokwium/kolokwiów	wykład
	Ocena podsumująca		
	P1	Ocena z egzaminu/zaliczenia	wykład
	P2	średnia ocen uzyskanych podczas zajęć	laboratorium

Kryteria oceny				
	EU1	EU2	EU3	EU4
Na ocenę 3	51%	51%	51%	51%
Na ocenę 3,5	62%	62%	62%	62%
Na ocenę 4	74%	74%	74%	74%
Na ocenę 4,5	86%	86%	86%	86%
Na ocenę 5	95%	95%	95%	95%

Literatura	
Literatura podstawowa	<ol style="list-style-type: none"> <li>1. Foster Ian: Designing and Building Parallel Programs. Addison-Wesley, 1995</li> <li>2. Weiss Zbigniew, Grulewski Tadeusz: Programowanie współbieżne i rozproszone. Wydawnictwo Naukowo-Techniczne, Warszawa 1993</li> <li>3. Tanenbaum Andrew S., van Steen Maarten: Systemy rozproszone. Wydawnictwo Naukowo-Techniczne, Warszawa 2006</li> </ol>
Literatura uzupełniająca	<ol style="list-style-type: none"> <li>1. Gabassi Michel, Dupouy Bertrand: Przetwarzanie rozproszone w systemie UNIX. Lupus, Warszawa 1996</li> <li>2. Funika Włodzimierz, Koperek Paweł, Kitowski Jacek: Management of Heterogeneous Cloud Resources with Use of the PPO. Springer-Verlag, Euro-Par Workshops 2020: 148-159 2021</li> </ol>

Nakład pracy studenta		
	Studia niestacjonarne	Studia stacjonarne
Godziny kontaktowe z nauczycielem akademickim lub inną osobą prowadzącą zajęcia (wykłady, wiczenia, laboratoria, konwersatoria)	32	60
Przygotowanie do zajęć, w tym studiowanie zalecanej literatury podstawowej i uzupełniającej	48	20
Przygotowanie projektu	0	0
Przygotowanie się do egzaminu / zaliczenia	20	20
Inne (np. esej, prezentacja, referat, koreferat, sprawozdanie z wykonanych zadań)	0	0
<b>Łączny nakład pracy studenta w godz.</b>	100	100
<b>Liczba punktów ECTS</b>	4	4

Macierz realizacji zajęć					
Efekty uczenia się	Odniesienie danego efektu do kierunkowych efektów uczenia się	Cele kształcenia	Treści programowe	Metody/Narzędzia dydaktyczne	Sposoby oceny
EU1	K_W07	C1, C3, C7	W1, W15	N1, N2, N3	F2, P1
EU2	K_W04	C2, C3, C4, C5, C6, C7	W4, W5, W6, W7, W8, W10, W14, L3, L5, L7, L9, L12, L13, L15	N1, N2, N3	F1, F2, P1
EU3	K_U08, K_U09	C1, C2, C3, C4, C5, C6, C7	W2, W3, W6, W9, W11, W12, W13, W15, L1, L2, L4, L5, L6, L8, L10, L11, L12, L13, L14	N1, N2, N3	F1, P1, P2
EU4	K_K03, K_K04	C2, C4, C5, C6, C7	W5, W7, W9, W10, W11, W14, L6, L7, L11, L14	N1, N2, N3	F1, P2