

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2023-2025

Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>programowanie współbieżne i rozproszone</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia II stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok II, semestr 2</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy</i>
Język wykładowy	<i>język polski / język angielski</i>
Koordynator	<i>dr inż. Piotr Lasek</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Piotr Lasek, mgr inż. Marcin Mrukowicz</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Projekt	Liczba pkt. ECTS
2	20			30					4

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

egzamin

2. WYMAGANIA WSTĘPNE

Znajomość podstawowych problemów i umiejętność ich rozwiązywania z zakresu algorytmów i struktury danych, programowania obiektowego, systemów operacyjnych, technologii sieciowych, architektury komputerów.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Zapoznanie z podstawowymi pojęciami i problemami związanymi z programowaniem współbieżnym i rozproszonym.
C ₂	Zapoznanie ze architekturami systemów współbieżnych i rozproszonych.
C ₃	Zapoznanie z technologiami i narzędziami do tworzenia programów współbieżnych i rozproszonych.
C ₄	Kształtowanie umiejętności tworzenia poprawnych programów współbieżnych i rozproszonych przy wykorzystaniu różnych technologii i narzędzi.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu Student	Odniesienie do efektów kierunkowych
EK_01	Student zna pojęcia i problemy z zakresu programowania współbieżnego i rozproszonego oraz architektury sprzętowej systemów równoległych i rozproszonych.	K_W01
EK_02	Student zna i poprawnie identyfikuje technologie i narzędzia tworzenia programów współbieżnych i rozproszonych; charakteryzuje ich ważniejsze elementy.	K_W01
EK_03	Student potrafi poprawnie zaimplementować programy współbieżny/rozproszone. Potrafi przeanalizować działanie takiego programu.	K_U03
EK_04	Student rozumie wyzwania i potrzebę stałego śledzenia najnowszych osiągnięć związanych technologiami za pomocą których można tworzyć nowoczesne systemy współbieżne i rozproszone.	K_U03

3.3 Treści programowe**A. Problematyka wykładu**

Podstawowe pojęcia i terminologia w programowaniu współbieżnym i rozproszonym (Program - proces - wątek, Równoległość i współbieżność, Przeplot, Własność bezpieczeństwa, Własność żywotności, Zakleszczenie, Zagłodzenie)
Klasyczne problemy współbieżności (Wzajemne wykluczanie, Producenci i konsumenci, Czytelnicy i pisarze, Pięciu filozofów)
Modele formalne (Sieci Petriego)
Komunikacja asynchroniczna (Bufor, Notacja Lindy, Język Ada, Notacja CSP)
Model scentralizowany (Semafor, Monitory, Muteksy)
Modele architektur sprzętowych (Taksonomia Flynna)
Zarządzanie procesami w architekturach POSIX (Funkcje systemu operacyjnego: fork, pipe, exec)

Awarie i algorytmu uzgadniania (Wiarygodność, Problem bizantyjskich generałów, Algorytm Lamporta)
Elementy programowanie wielowątkowego przy użyciu języka Java i C++ (Procesy i wątki, Synchronizacja, Współbieżność wysokiego poziomu)

B. Problematyka ćwiczeń laboratoryjnych

Tworzenie procesów za pomocą poleceń exec i fork w systemach UNIX, w tym synchronizowanie ich działania.
Programowe tworzenie potoków (ang. pipeline): anonimowych i nazwanych.
Pamięć współdzielona.
Komunikaty międzyprocesowe.
Tworzenie programów wielowątkowych w języku Java.
Tworzenie programów sieciowych w języku Java.
Tworzenie własnych prostych protokołów sieciowych warstwy aplikacji.
Tworzenie programów wykorzystujących zdalne wywoływanie metod np. ZeroC ICE, gRPC.
Tworzenie programów, wykorzystujących znormalizowane interfejsy pomiędzy serwerem HTTP a zewnętrznym programem np. CGI, FastCGI.

3.4 Metody dydaktyczne

Wykład: wykład problemowy wspomagany prezentacją multimedialną oraz studiami przypadków.

Ćwiczenia laboratoryjne: praktyczne zadania programistyczne przy komputerze.

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	egzamin, kolokwium	wykład
EK_02	egzamin, obserwacja w trakcie zajęć	wykład, laboratorium
EK_03	kolokwium, obserwacja w trakcie zajęć	laboratorium
EK_04	obserwacja w trakcie zajęć	laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Kolokwium pisemne w formie testu Efekt EK_01, EK_02: W celu uzyskania zaliczenia z wykładu, student rozwiązuje kolokwium testowe na poziomie minimum 50%. Wykład - egzamin pisemny w formie testu:
--

Efekt EK_01, EK_02

Rozwiązując test egzaminacyjny, student wykazuje, że zna pojęcia z zakresu programowania współbieżnego i rozproszonego oraz architektury sprzętowej systemów równoległych i rozproszonych, a także potrafi identyfikować, nazywać technologie i narzędzia do tworzenie programów współbieżnych i rozproszonych.

Ocena przyznawana na podstawie wyniku procentowego zgodnie z poniższą tabelą ocen:

Uzyskany wynik [%]	Ocena
0 - 49	niedostateczna
50 - 59	dostateczna
60 - 69	dostateczna plus
70 - 79	dobra
80 - 89	dobra plus
90 - 100	bardzo dobra

Student może uzyskać dodatkowe punkty (procenty) na podstawie zadań domowych, które mogą być zadawane na zajęciach wykładowych. Punkty te mogą zostać doliczone do wyniku egzaminu maksymalnie podwyższając ocenę o 0,5 stopnia.

Efekt EK_02:

DST: Student potrafi zidentyfikować i ogólnie nazwać technologie i narzędzia tworzenia aplikacji współbieżnych i rozproszonych.

DB: Student potrafi zidentyfikować i ogólnie nazwać technologie i narzędzia tworzenia aplikacji współbieżnych i rozproszonych oraz omówić ich ważniejsze elementy.

BDB: Student potrafi zidentyfikować i ogólnie nazwać technologie i narzędzia tworzenia aplikacji współbieżnych i rozproszonych oraz omówić ich wszystkie elementy.

Laboratorium – praktyczne zadania programistyczne:

Efekt EK_03:

DST: Student potrafi poprawnie zaimplementować proste programy współbieżne i rozproszone oraz analizować ich działanie.

DB: Student potrafi poprawnie zaimplementować średnio-zaawansowane programy współbieżne i rozproszone oraz analizować ich działanie.

BDB: Student potrafi poprawnie zaimplementować zaawansowane programy współbieżne i rozproszone oraz analizować ich działanie.

Pod uwagę brana jest również aktywność na zajęciach, która może obniżyć lub podwyższyć ocenę końcową o pół stopnia.

Efekt EK_o4:

„zal”/„nzal” na podstawie aktywności w trakcie zajęć.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	50
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	3
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	50
SUMA GODZIN	103
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

LITERATURA PODSTAWOWA:

1. Literatura podstawowa: 1. Ben-Ari, M.: Podstawy programowania współbieżnego i rozproszonego. Wydawnictwa Naukowo-Techniczne, Warszawa, 2009.
2. Brzuszek, M., Stpiczyński, P.: Programowanie współbieżne i rozproszone w języku Java. UMCS, Lublin, 2012 (dostęp on-line: http://informatyka.umcs.lublin.pl/files/Skrypty/stpiczynski_brzuszek_programowanie_ws_pobiezne_i_rozproszone_w_jezyku_java.pdf).
3. Czech, Z.J.: Wprowadzenie do obliczeń równoległych. Wydawnictwo Naukowe PWN, Warszawa, 2013.

LITERATURA UZUPEŁNIAJĄCA:

1. Barteczko, K.: Programowanie obiektowe i zdarzeniowe w Javie. Wydawnictwo PJWSTK, Warszawa, 2005.
2. Barteczko, K., Drabik, W., Starosta, B.: Nowe metody programowania, tom I. Wydawnictwo PJWSTK, Warszawa, 2005.
3. Barteczko, K., Drabik, W., Starosta, B.: Nowe metody programowania, tom II. Wydawnictwo PJWSTK, Warszawa, 2006.

4. Herlihy, M., Shavit, N.: Sztuka programowania wieloprocessorowego. Wydawnictwo Naukowe PWN, Warszawa, 2010.
5. Kaliszewska, M., Pieciukiewicz, T., Sobczak, A., Stencel, K.: Technologie internetowe. Wydawnictwo PJWSTK, Warszawa, 2007.