

Programowanie współbieżne i rozproszone - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie współbieżne i rozproszone
Kod przedmiotu	11.3-WI-INF-PWiR
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2020/2021

Informacje o przedmiocie	
Semestr	4
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	dr inż. Tomasz Gratkowski

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	18	1,2	Egzamin
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę

Cel przedmiotu

- zapoznanie studentów z podstawowymi technikami programowania równoległego
- zapoznanie studentów z podstawowymi technikami programowania rozproszonego
- ukształtowanie wśród studentów zrozumienia konieczności stosowania technik programowania równoległego i rozproszonego
- ukształtowanie podstawowych umiejętności w zakresie pisania oprogramowania wykorzystującego zrównoleglenie oraz architekturę rozproszoną

Wymagania wstępne

Podstawy programowania,
Programowanie obiektowe,
Architektura komputerów I i II,
Systemy operacyjne I i II,
Język Java i technologie Web.

Zakres tematyczny

Programowanie równoległe - podstawowe pojęcia: programowanie współbieżne, proces, współbieżność procesów, zasada podziału czasu, jednoczesność, komunikacja i synchronizacja między procesami, wzajemne wykluczanie, blokada, zagłódzenie. Cele programowania współbieżnego. Zalety i wady programów współbieżnych. Przetwarzanie strumieniowe.

Semafor: semafor ogólny, semafor binarny.

Tworzenie aplikacji współbieżnych w języku Java. Monitory. Inne mechanizmy synchronizacji wątków: kolejki blokujące, bariery, rejestry, przekaźniki.

Klasyczne problemy współbieżności: problem producenta i konsumenta, problem czytelników i pisarzy, problem pięciu filozofów.

Charakterystyka i cele projektowe systemów rozproszonych. Komunikacja międzyprocesowa. Wytyczne projektowe i cechy komunikacji międzyprocesowej. Architektury stosowane przy budowaniu systemów rozproszonych.

Wywoływanie zdalnych procedur. Zdalne wywoływanie metod. Budowanie aplikacji rozproszonych w oparciu o Remote Method Invocation z języka Java. Integracja aplikacji z innymi programistycznymi środowiskami rozproszonymi.

Czas i zegary logiczne. Koordynacja działań w systemach rozproszonych. Algorytmy elekcji. Transakcje i sterowanie współbieżnością w systemach rozproszonych.

Zakleszczenia i algorytmy ich wykrywania w systemie rozproszonym.

Metody kształcenia

wykład: wykład konwencjonalny/tradycyjny

laboratorium: ćwiczenia laboratoryjne

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrąfi objaśnić mechanizmy koordynacji działań w systemach rozproszonych.	• K_W09 • K_U15	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium

Opis efektu	Symbolce efektów	Metody weryfikacji	Forma zajęć
Potrafi zaprojektować i utworzyć oprogramowanie zorientowane obiektowo wykorzystujące mechanizmy programowania współbieżnego i rozproszonego.	• K_W09 • K_U15	• bieżąca kontrola na zajęciach	• Laboratorium
Student potrafi wytłumaczyć potrzebę stosowania programowania współbieżnego.	• K_W09	• egzamin - ustny, opisowy, testowy i inne	• Wykład
Jest świadomy potrzeby wykorzystania systemów i programów rozproszonych.	• K_K04	• egzamin - ustny, opisowy, testowy i inne	• Wykład
Potrafi opisać mechanizm projektowania warstwy komunikacyjnej i zagadnień związanych z wymianą informacji w systemach rozproszonych.	• K_W09 • K_U15	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium
Potrafi rozróżniać podstawowe modele architektoniczne stosowane do projektowania systemów rozproszonych.	• K_W09	• egzamin - ustny, opisowy, testowy i inne	• Wykład

Warunki zaliczenia

Wykład - warunkiem zaliczenia jest uzyskanie pozytywnej oceny z egzaminu pisemnego lub testu.

Laboratorium - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium.

Składowe oceny końcowej = wykład: 50% + laboratorium: 50%

Literatura podstawowa

1. Ben-Ari M.: Podstawy programowania współbieżnego i rozproszonego, WNT, 1996.
2. Foster I.: Designing and Building Parallel Programs, (2020) <http://www.mcs.anl.gov/~itf/dbpp/>
3. Coulouris G. et al.: Distributed Systems. Concepts and Design, Addison Wesley, 2011
wydanie polskie: Systemy rozproszone. Podstawy i projektowanie, WNT 1998
4. Tanenbaum S., Maarten van Steen: Distributed Systems. Principles and Paradigms, Prentice Hall, 2016
wydanie polskie: Systemy rozproszone - Zasady i paradygmaty, WNT, 2005
darmowa wersja książki (2020) - Distributed systems <https://www.distributed-systems.net/index.php/books/ds3/ds3-sneak-preview/>
5. Garg V. K.: Concurrent and Distributed Computing in Java, Wiley-IEEE Press 2004
6. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea: Java Concurrency in Practice, Addison-Wesley Professional 2006
7. Scott Oaks, Henry Wong: Java Threads, Third Edition, O'Reilly 2004
8. Cay S. Horstmann, Gary Cornell: Core Java, Vol. 1: Fundamentals, Prentice Hall PTR, 2018
wydanie polskie: Java. Podstawy. Cay S. Horstmann, 2019 HELION
9. Cay S. Horstmann: Core Java, Vol. 2: Advanced Features, Prentice Hall PTR, 2019
wydanie polskie: Java. Techniki zaawansowane. Cay S. Horstmann, 2020 HELION
10. Kathy Sierra, Bert Bates: Head First Java, 2nd Edition, O'Reilly Media, 2009
wydanie polskie: Java. Rusz głową! Wydanie II, Helion 2010
11. Burns B.: Projektowanie systemów rozproszonych. Wzorce i paradygmaty dla skalowalnych, niezawodnych usług, Helion, 2018

Literatura uzupełniająca

1. Roger Wattenhofer: Principles of Distributed Computing, Spring 2016, (2020) https://disco.ethz.ch/courses/podc_allstars/lecture/podc.pdf
2. Distributed Systems for fun and profit (2020) <http://book.mixu.net/distsys/single-page.html>

Uwagi

Zmodyfikowane przez dr inż. Tomasz Gratkowski (ostatnia modyfikacja: 25-04-2020 14:49)

Wygenerowano automatycznie z systemu SyllabUZ