

Programowanie współbieżne i rozproszone - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie współbieżne i rozproszone
Kod przedmiotu	11.3-WI-INFP-PWiR
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	4
Liczba punktów ECTS do zdobycia	5
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, zagłodzenie. Cele programowania współbieżnego. Zalety i wady programów współbieżnych.

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	• sprawdzian • test	• Laboratorium

Opis efektu	Symbole 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	• sprawdzian • test	• Wykład
Jest świadomy potrzeby wykorzystania systemów i programów rozproszonych.	• K_K04	• sprawdzian • test	• Wykład
Potrafi opisać mechanizm projektowania warstwy komunikacyjnej i zagadnień związanych z wymianą informacji w systemach rozproszonych.	• K_W09	• sprawdzian • test	• Laboratorium
Potrafi rozróżnić podstawowe modele architektoniczne stosowane do projektowania systemów rozproszonych.	• K_W09 • K_U15	• sprawdzian • test	• 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, Warszawa, 1996.
2. Foster I.: Designing and Building Parallel Programs, <http://www.mcs.anl.gov/~itf/dbpp/>
3. Coulouris G. et al.: Distributed Systems. Concepts and Design (4th ed.), Addison Wesley, 2005 (wydanie polskie "Systemy rozproszone. Podstawy i projektowanie", WNT 1998)
4. Tanenbaum S.: Distributed Systems. Principles and Paradigms (2nd ed.), Prentice Hall 2002
5. Garg V. K.: Concurrent and Distributed Computing in Java, March 2004, Wiley-IEEE Press
6. Horstmann C. S., Cornell G.: Core Java™ 2: Volume II—Advanced Features, Prentice Hall 2008 (wydanie polskie "Java. Techniki zaawansowane. Wydanie VIII", Helion 2009)
7. Goetz B., Peierls T., Bloch J., Bowbeer J., Holmes D., Lea D.: Java Concurrency in Practice, Addison-Wesley Professional 2006

Literatura uzupełniająca

Uwagi

Zmodyfikowane przez dr inż. Tomasz Gratkowski (ostatnia modyfikacja: 26-04-2017 22:31)

Wygenerowano automatycznie z systemu SyllabUZ