Concurrent and distributed programming - course description

General information

Course name	Concurrent and distributed programming
Course ID	11.3-WE-INFP-CaDP-Er
Faculty	Faculty of Engineering and Technical Sciences
Field of study	Computer Science
Education profile	academic
Level of studies	First-cycle Erasmus programme
Beginning semester	winter term 2021/2022

Course information

Semester	4
ECTS credits to win	6
Course type	obligatory
Teaching language	english
Author of syllabus	

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Exam
Laboratory	30	2	-	-	Credit with grade

Aim of the course

- Familiarize students with basic techniques of concurrent programming
- Familiarize students with basic techniques of distributed programming
- Teach students the fundamental skills of using concurrent and distributed programming techniques
- Learn basic skills in developing software used balanced and distributed architecture

Prerequisites

Principles of programming, Java programming, Computer architectures I and II.

Scope

Concurrent programming – basic concept: process, shared resources, critical section, mutual exclusion, synchronization, deadlock, starvation.

Aims of concurrent programming. Advantages and disadvantages of concurrent programming.

Semaphores: general semaphore, binary semaphore, synchronization of processes with usage of semaphores.

Concurrent programming in Java. Monitors. Additional methods of threads synchronization: blocking queued, barriers, countdown of latch and exchanger.

Classical problems of concurrent programming: dining philosophers problem, producer-consumer problem, readers-writers problems.

Characterization of **Distributed Systems**. Inter-process communication. Guidelines for design of inter-process communication.

Remote procedure call (RPC). Remote method invocation (RMI). How to build of distributed applications in Java RMI. Integration different distributed environments.

Time and coordination in distributed systems. Logical clock. Election algorithm. Transactions and concurrency control in distributed systems. Algorithms for deadlock detection in distributed systems.

Teaching methods

Lecture: conventional lecture
Laboratory: laboratory exercises, group work

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Can describe the mechanism of communication layer design and the issues connected with data exchange in distributed systems		• a quiz • a test	• Laboratory
Can design and create object-oriented software employing concurrent and distributed programming mechanisms		• an ongoing monitoring during classes	• Laboratory
Can distinguish basic architectural models used for the design of distributed systems		• an exam - oral, descriptive, test and other	• Lecture

Outcome description	Outcome symbols	Methods of verification	The class form
Can explain the mechanisms of coordination of activities in distributed systems		• a quiz • a test	• Laboratory
Can explain the need for application of concurrent programming		• an exam - oral, descriptive, test and other	• Lecture
Is aware of the need to use distributed systems and programs		• an exam - oral, descriptive, test and other	• Lecture

Assignment conditions

Lecture - obtaining a positive grade in written exam.
Laboratory - the main condition to get a pass are sufficient marks for all exercises and tests conducted during the semester.
Calculation of the final grade: = lecture 50% + laboratory 50%.

Recommended reading

1. Ben-Ari M.: Principles of Concurrent and Distributed Programming, Addison-Wesley, 2006.
2. Foster I.: Designing and Building Parallel Programs, (2020) https://www.mcs.anl.gov/~itf/dbpp/
3. Coulouris G. et al.: Distributed Systems. Concepts and Design, Addison Wesley, 2005.
4. Tanenbaum S., Maarten van Steen: Distributed Systems. Principles and Paradigms, Prentice Hall, 2016
   (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. Cay S. Horstmann, Gary Cornell: Core Java, Vol. 1: Fundamentals, Prentice Hall PTR, 2018
7. Cay S. Horstmann: Core Java, Vol. 2: Advanced Features, Prentice Hall PTR, 2019
8. Kathy Sierra, Bert Bates: Head First Java, 2nd Edition, O'Reilly Media, 2009
9. Goetz B., Peierls T., Bloch J., Bowbeer j., Holmes D., Lea D.: Java Concurrency in Practice, Addison-Wesley Professional, 2006.
10. Burns B.: Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services, O'Reilly, 2018

Further reading

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

Notes

Modified by dr inż. Tomasz Gratkowski (last modification: 21-07-2021 10:03)