

FIȘA DISCIPLINEI

Programare Paralelă si Distribuită

Anul universitar 2025/2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Inteligență Artificială
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare Paralelă și Distribuită				Codul disciplinei		MLE5077			
2.2. Titularul activităților de curs			Lect. dr. Radu Lupșa								
2.3. Titularul activităților de seminar			Lect. dr. Radu Lupșa								
2.4. Anul de studiu		3	2.5. Semestrul		5	2.6. Tipul de evaluare		E	2.7. Regimul disciplinei		Obligativu

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	0/2/1
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					20
Tutoriat (consiliere profesională)					10
Examinări					5
Alte activități					-
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamentele programării, Programare orientată obiect, Structuri de date și algoritmi, Sisteme de operare
4.2. de competențe	Abilități de programare

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Sală cu videoproiector; stații cu medii de dezvoltare pentru C++, Python, Java și C#

6.1. Competențele specifice acumulate¹

Competențe profesionale/ esențiale	• utilizarea bazelor teoretice ale informaticii și a modelelor formale • utilizarea instrumentelor informatice în context interdisciplinar
Competențe transversale	• aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

6.2. Rezultatele învățării

Cunoștințe	Absolventul are cunoștințe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor. Absolventul are cunoștințe legate de programare, matematică, inginerie și tehnologie și are abilitățile necesare pentru a le folosi în crearea de sisteme informatice complexe.
Aptitudini	Absolventul are aptitudinile necesare pentru conceperea programelor de calculator și analiza sistemelor software. Absolventul are abilitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante.
Responsabilități și autonomie	Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Dobândirea conceptelor de bază legate de programarea concurrentă, paralelă și distribuită • Bazele comunicației între procese și <i>thread</i>-uri, pe aceeași mașină și pe mașini distincte • Cunoașterea tehnicilor de bază ale programării paralele • Cunoașterea și utilizarea <i>pattern</i>-urilor de proiectare a aplicațiilor paralele • Cunoașterea și utilizarea <i>framework</i>-urilor pentru dezvoltarea aplicațiilor paralele și distribuite
---------------------------------------	--

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

7.2 Obiectivele specifice	• Arhitecturi paralele și sisteme de programare paralelă • Cunoașterea utilizării tehnicilor de programare paralelă în rezolvarea de probleme • Cunoașterea modului de evaluare a creșterii performanțelor prin paralelizare • Abilitatea de a lucra, independent sau în echipă, în rezolvarea problemelor într-un cadru paralel și/sau distribuit.
----------------------------------	--

8. Conținuturi

8.1 Curs	Metode de predare	Observații
C1. Introducere. Necesitatea paralelismului. Programare concurentă vs paralelă vs distribuită.	Expunere: descriere, explicații, exemple, dezbateri.	
C2. Arhitecturi paralele: pipeline, mașini vectoriale, gomputere grid și cluster.	Expunere: descriere, explicații, exemple, dezbateri.	
C3. <i>Thread</i> -uri. <i>Race conditions</i> , exclusiune mutuală, interblocare (<i>deadlock</i>). Primitive de sincronizare.	Expunere: descriere, explicații, exemple, dezbateri.	
C4. Paralelism producător-consumator. Primitive de nivel coborât (variabile condiționale) și de nivel înalt (<i>futures</i> , cozi producător-consumator).	Expunere: descriere, explicații, exemple, dezbateri.	
C5-C6. Programare asincronă. <i>Futures</i> cu continuări. Corutine.	Expunere: descriere, explicații, exemple, dezbateri.	
C7. Algoritmi paraleli simpli.	Expunere: descriere, explicații, exemple, dezbateri.	
C8. Algoritmi de descompunere recursivă și de explorare paralelă.	Expunere: descriere, explicații, exemple, dezbateri.	
C9. Programare distribuită folosind MPI.	Expunere: descriere, explicații, exemple, dezbateri.	
C10. Descompunere recursivă și explorare paralelă în varianta distribuită.	Expunere: descriere, explicații, exemple, dezbateri.	
C11. Protocoale distribuite. Ceasuri Lamport.	Expunere: descriere, explicații, exemple, dezbateri.	
C12. Memorie partajată distribuită.	Expunere: descriere, explicații, exemple, dezbateri.	
C13. Programarea pe GPGPU. OpenCL.	Expunere: descriere, explicații, exemple, dezbateri.	
C14. Toleranța la pene.	Expunere: descriere, explicații, exemple, dezbateri.	

Bibliografie

- <http://www.cs.ubbcluj.ro/~rlupsa/edu/pdp/>
- Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
- Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation,” Morgan Kaufmann,, 2012 .
- Berna L. Massingill, Timothy G. Mattson, and Beverly A. Sanders, Addison A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004.
- Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
- D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
- V. Niclescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
- D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
- B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
- E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.
- Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed. Alabastra, grupul Microinformatica, Cluj, 2002 .
- OpenMP Tutorials

• MPI Tutorials • OpenCL Tutorials		
8.2 Seminar / laborator	Metode de predare	Observații
L1. Introducere	Dialog, dezbateri, exemple, descoperire dirijată.	
L2-L3. Primitive de sincronizare.	Dialog, dezbateri, exemple, descoperire dirijată.	
L4. Paralelism producător-consumator.	Dialog, dezbateri, exemple, descoperire dirijată.	
L5-L6. Programare asincronă.	Dialog, dezbateri, exemple, descoperire dirijată.	
L7. Algoritmi paraleli simpli.	Dialog, dezbateri, exemple, descoperire dirijată.	
L8. Descompunere recursivă.	Dialog, dezbateri, exemple, descoperire dirijată.	
L9. Explorare paralelă.	Dialog, dezbateri, exemple, descoperire dirijată.	
L10. Programare distribuită folosind MPI.	Dialog, dezbateri, exemple, descoperire dirijată.	
L11 Descompunere recursivă și explorare paralelă utilizând MPI	Dialog, dezbateri, exemple, descoperire dirijată.	
L12. Memorie partajată distribuită	Dialog, dezbateri, exemple, descoperire dirijată.	
L13. OpenCL	Dialog, dezbateri, exemple, descoperire dirijată.	
L14. Finalizarea activităților de laborator.	Dialog, dezbateri, exemple, descoperire dirijată.	
Bibliografie • Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006. • Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004. • Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. • E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994. • Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilly, 2011. • Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002. • http://download.oracle.com/javase/tutorial/ • http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx • http://www.cs.ubbcluj.ro/~rlupsa/edu/pdp/		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

• Cursul urmează recomandările ACM și IEEE pentru studiile de informatică • Cursul face parte din curricula în toate marile universități din țară și de peste hotare • Companiile de software consideră conținutul cursului ca fiind important pentru dobândirea deprinderilor avansate de programare.
--

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Conoaștrerea conceptelor	Examen scris	50%

	de bază		
	Aplicarea cunoștințelor teoretice în rezolvarea de probleme	Proiect de semestru	20%
10.5 Seminar/laborator	Aplicarea cunoștințelor teoretice în rezolvarea de probleme	Evaluarea lucrărilor de laborator	30%
10.6 Standard minim de performanță			
• Minim 12 din 14 prezențe la laborator • Nota minim 5 (din 10) la examenul scris • Media finală minim 5 (din 10).			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

...

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică*".