

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematica și Informatica</b>
1.3 Departamentul	<b>Departamentul de Informatica</b>
1.4 Domeniul de studii	<b>Informatica</b>
1.5 Ciclul de studii	<b>Licenta</b>
1.6 Programul de studiu / Calificarea	<b>Informatica</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Programare paralelă și distribuită</b>						
2.2 Titularul activităților de curs	<b>Conf. univ.dr. Niculescu Virginia</b>						
2.3 Titularul activităților de seminar/laborator	<b>Conf. univ.dr. Niculescu Virginia</b>						
2.4 Anul de studiu	<b>3</b>	2.5 Semestrul	<b>5</b>	2.6. Tipul de evaluare	<b>E</b>	2.7 Regimul disciplinei	<b>Obligativu</b>

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					5
Examinări					6
Alte activități: .....					-
3.7 Total ore studiu individual	58				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> <li>Fundamentele programării, Programare orientată obiect</li> <li>Structuri de date și algoritmi</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li>Abilități de programare Java și C++</li> </ul>

### 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> <li>Sala cu proiector</li> </ul>
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>laborator cu stații de lucru</li> </ul>

### 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a domeniului și că este capabil</p> <p>(a) să exprime aceste cunoștințe;</p> <p>și, de asemenea, că poate folosi aceste cunoștințe în</p> <p>(b) rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă și distribuită.</p> <p><b>C6:</b>  C6.1 Identificarea conceptelor și modelelor de bază pentru sisteme de calcul și rețele de calculatoare.  C6.4 Efectuarea de măsurători de performanță pentru timpi de răspuns, consum de resurse.  C6.5 Realizarea unor proiecte de rețele de calculatoare/arhitecturi paralele și distribuite</p>
<b>Competențe transversale</b>	<p><b>CT1</b> 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</p> <p><b>CT3</b> 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</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Însușirea principalelor concepte și mecanisme cu care se operează în contextul programării paralele, concurente și distribuite.</li> <li>- Prezentarea bazelor de intercomunicare între procese și threaduri, aflate pe aceeași mașină sau pe mașini aflate la distanță.</li> <li>- Însușirea de cunoștințe de bază ale programării paralele, concurente și distribuite</li> <li>- Cunoașterea și înțelegerea paradigmatelor și tehnicilor de bază ale programării paralele.</li> <li>- Înțelegerea și folosirea unor șabloane de proiectare pentru dezvoltarea aplicațiilor paralele.</li> <li>- Deprinderea folosirii unor framework-uri pentru dezvoltarea aplicațiilor paralele și distribuite.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Cunoașterea elementelor de bază ale arhitecturilor paralele și ale sistemelor de programare paralelă</li> <li>• Abilitatea de a aplica tehnici specifice programării paralele în rezolvarea problemelor.</li> <li>• Abilitatea de a evalua creșterea de performanță obținută prin folosirea paralelizării.</li> <li>• Abilitatea de a lucra independent sau în echipă pentru a rezolva probleme într-un context de programare paralelă și/sau distribuită.</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
C1 Introducere generală în programarea paralelă și distribuită - necesitatea folosirii paralelismului; - programare paralelă vs. progr. distribuită, vs. Progr. Concurență - niveluri de folosire a paralelismului	Expunere	
C2. Arhitecturi paralele – <i>Taxonomii (Flynn)</i> SMP, MPP, interconnection networks Mașini vectoriale Sisteme grid și clustere, Supercalculatoare, acceleratoare	Expuneri: concepte, exemple, studii de caz	
C3. Tipuri și modele de paralelism • Paralelism implicit vs. Paralelism explicit - Modelul Data-parallel - Modelul Message-passing (distributed memory) Modelul Shared-memory Procese versus fire de execuție • gestiunea proceselor	Expuneri: concepte, exemple, studii de caz	
C4. Message Passing parallel programming Programare paralelă în medii cu memorie distribuită. MPI	Expuneri: concepte, exemple, studii de caz	

C5. Programare paralela in medii cu memorie partajata - C++ Threads, Java Threads	Expuneri: concepte, exemple, studii de caz	
C6. Concurenta – concepte Race-conditions, critical section, mutual exclusion, deadlock	Expuneri: concepte, exemple, studii de caz	
C7. Concurenta – concepte Sincronizare: monitoare, semafoare, variabile conditionale	Expuneri: concepte, exemple, studii de caz	
C8. Programare paralela asincrona	Expuneri: concepte, exemple, studii de caz	
C9. Programare paralela in medii cu memorie partajata. - OpenMP	Expuneri: concepte, exemple, studii de caz	
C10 . Evaluarea performantei programelor paralele: complexitate-timp, complexitate-procesor, acceleratie, eficienta, cost; granularitate, scalabilitate	Expuneri: concepte, exemple, studii de caz	
C11. <i>Data parallel programming</i> : Cadrul general de dezvoltare a aplicațiilor GPU Arhitectură; platforma NVIDIA API de programare; modelul CUDA	Expuneri: concepte, exemple, studii de caz	
C12- Sabloane de proiectare pentru programarea paralela <i>Master-slaves; Task-Farm/Work-Pool; Divide &amp; Impera; Pipeline</i>	Expuneri: concepte, exemple, studii de caz	
C13 Sabloane de proiectare pentru programarea distribuita	Expuneri: concepte, exemple, studii de caz	
C14. Analiza comparativa generala a noilor concepte/tehnici/principii/sabloane introduse	studii de caz	
Bibliografie		
<ol style="list-style-type: none"> <li>1. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.</li> <li>2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012 .</li> <li>3. <a href="#">F. Buschmann</a>, <a href="#">K. Henney</a>, <a href="#">D. C. Schmidt</a>. Pattern-Oriented Software Architecture Volume 4: A Pattern Language for</li> <li>4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.</li> <li>5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.</li> <li>6. B. L. Massingill, T.G. Mattson, and B. A. Sanders, A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004.</li> <li>7. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.</li> <li>8. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.</li> <li>9. Distributed Computing Volume 4, Wiley. 2007.</li> <li>10. M. Richards. Software Architecture Patterns. Understanding Common Architecture. Patterns and When to Use Them 2015 O'Reilly Media.</li> <li>11. <a href="#">D. Schmidt</a> (Author), <a href="#">M. Stal</a> (Author), <a href="#">H. Rohnert</a> (Author), <a href="#">F. Buschmann</a>. Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects Volume 2. Wiley, 2000.</li> <li>12. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002</li> <li>13. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.</li> <li>14. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed. Albastra, grupul Microinformatica, Cluj, 2002.</li> </ol> <p>***, Tutoriale OpenMP ***, Tutoriale MPI ***, Tutoriale CUDA</p>		
8.2 Laborator		
L1 –L2 . Threads vs. Processes	Discutii, exemplificare, evaluare	
L2-L4 MPI - exemple	Discutii, exemplificare, evaluare	
L5. Programare distribuita MPI	Discutii, exemplificare, evaluare	
L6- L8. Concurenta – conditionari, sincronizari cu exemplificari C, Java	Evaluare	
L9. Probleme de sincronizare	Discutii, exemplificare, evaluare	
L10. Aplicatii folosind programare concurenta- multithreading	Discutii, exemplificare, evaluare	
L11- L12 Proiect Client-Server	Discutii, exemplificare, evaluare	
L13-L14 Proiect CUDA	Evaluare	
Bibliografie aditionala		
<ol style="list-style-type: none"> <li>1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.</li> </ol>		

2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
5. J. Sanders, E. Kandrot. CUDA by Example. An Introduction to General-Purpose GPU Programming. Addison-Wesley. 2010.
6. A. WILLIAMS. C++ Concurrency in Action. PRACTICAL MULTITHREADING. MANNING, 2012.
7. \*\*\*, Tutoriale Java <http://download.oracle.com/javase/tutorial/>
8. \*\*\*, Tutoriale C# <http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx>
9. \*\*\*, Tutorial C++11
10. \*\*\*, OpenMP[<http://openmp.org/wp/>]
11. \*\*\*, MPI[<http://www.mpi-forum.org/>]

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

- Cursul respecta recomandările curriculei ACM și IEEE pentru studiile de informatică.
- Cursul apare în planurile de învățământ a celor mai importante universități din țară și străinătate.
- Firmele de soft consideră conținutul cursului important pentru dobândirea unor abilități 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	Cunoașterea conceptelor și mecanismelor de bază	Examen teoretic	30%
10.5 Laborator	Folosirea conceptelor și mecanismelor introduse la curs pentru rezolvarea unor probleme concrete	Programe de laborator - Evaluarea calitatii și completitudinii temelor primite -Evaluarea înțelegerii aplicării conceptelor și mecanismelor folosite	40%
		Teste practice	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> <li>○ Media finală trebuie să fie cel puțin 5 (pe o scară de la 1 la 10).</li> <li>○ Promovarea este condiționată de nota minimă 5 la activitatea de laborator și examenul teoretic.</li> </ul>			

Data completării

.....

Titular de curs

Niculescu Virginia

Titular de laborator

Niculescu Virginia

Data avizării în departament

.....

Director de departament

.....