

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatică
1.3 Departamentul	Departamentul de matematică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licența
1.6 Programul de studiu / Calificarea	Informatică

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Paradigme și tehnici ale programării paralele						
2.2 Titularul activităților de curs	Conf. Dr. Niculescu Virginia						
2.3 Titularul activităților de seminar	Conf. Dr. Niculescu Virginia						
2.4 Anul de studiu	3	2.5 Semestrul	6	2.6. Tipul de evaluare	colocviu	2.7 Regimul disciplinei	Optional

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

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	36	Din care: 3.5 curs	24	3.6 seminar/laborator	12
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					50
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					50
Tutoriat					15
Examinări					14
Alte activități: .....					-
3.7 Total ore studiu individual		110			
3.8 Total ore pe semestru		175			
3.9 Numărul de credite		7			

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

4.1 de curriculum	Fundamentele programării, Structuri de date
4.2 de competențe	Abilități de implementare programe.

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

5.1 De desfășurare a cursului	Sala cu proiector
5.2 De desfășurare a seminarului/laboratorului	laborator cu stații de lucru

### 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a subiectului și că este capabil (a) să exprime aceste cunoștințe; și, de asemenea, că poate folosi aceste cunoștințe în rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă.
<b>Competențe transversale</b>	<ul style="list-style-type: none"> <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 implementa programe paralele.</li> </ul>

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Aprofundarea conceptelor de bază ale programării paralele.</li> <li>- Abilitatea de a lucra independent sau în echipă pentru a rezolva probleme într-un context de programare paralelă.</li> <li>- Cunoaștere și abilități de implementare a paradigmatelor și tehnicilor de bază ale programării paralele.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>- Cunoașterea și folosirea celor mai importante paradigme/tehnici de dezvoltare a programelor paralele.</li> <li>- Implementarea programelor paralele folosind Java, C#, C++, Scala.</li> <li>- Realizarea unui Proiect (aplicație) folosind pentru implementare una dintre variantele specificate anterior</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Programarea paralelă: generalități • Paralelism implicit vs. Paralelism explicit - Modelul Data-parallel - Modelul Message-passing - Modelul Shared-variable - Arhitecturi paralele	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
2. Sabloane - <i>Master-slaves</i> - <i>Task-Farm/Work-Pool</i> Exemple	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
3. Pipeline ca paradigmă de programare paralelă. Exemple	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
4. Modelul PRAM (Parallel Random Access Machine) Rețele computaționale. Exemplificări. Evaluarea performanței programelor paralele	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
1. Scalabilitate. Evaluare teoretică și experimentală.	Expunere, descriere, explicație, exemple, discuții ale unor	

	studii de caz.	
2. Tehnici utilizate in constructia algoritmilor paraleli - Tehnica arborelui binar - Tehnica dublarii recursive - Tehnica par-impar - Tehnica reducerii ciclice par-impar	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
3. - Algoritmi de sortare; Paradigma divide & Impera	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
4. Exemple de algoritmi paraleli din teoria grafurilor. - Tehnica parrcursului Eulerian	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
5. Algoritmi paraleli din algebra liniara	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
6. Programare paralela in medii cu memorie partajata. <i>Java Threads, Executors, Futures, Scala, OpenMP</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7. Programare paralela in medii cu memorie partajata si distribuita. MPI + threads, MPI + OpenMP	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
8. Colocviu: Presentare proiecte	Discutii, analize, evaluari.	

#### Bibliografie

1. A Pattern Language for Parallel Programming. Berna L. Massingill, Timothy G. Mattson, and Bever Sanders, Addison Wesley Software Patterns Series, 2004.
2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012.
3. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
6. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
7. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
8. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
9. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.

8.2 Seminar / laborator	Metode de predare	Observații
1. Alegerea si specificarea unei probleme corespunzatoare proiectului	Dialog, dezbateri, explicatie, exemple.	
2. Proiectarea si implementarea solutiei secventiale	Dialog, dezbateri, explicatie, exemple.	
3. Proiectarea solutie paralela. Alegerea unui	Dialog, dezbateri,	

sablon de programare paralela adecvat. Evaluare teoretica a solutiei paralele	explicatie,exemple.	
4. Implementarea solutiei paralele.	Dialog, dezbatere, explicatie,exemple.	
5. Testare si validare	Dialog, dezbatere, explicatie,exemple.	
6. Presentari ale proiectelor studentilor	Dialog, dezbatere, explicatie,exemple.	
Bibliografie		
1. Parallel Programming in the .NET [ <a href="https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx">https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx</a> ] 2. Scala Programming Language, [ <a href="http://www.scala-lang.org">http://www.scala-lang.org</a> ] 3. Java Programming Language [ <a href="http://www.oracle.com/us/technologies/java/overview">www.oracle.com/us/technologies/java/overview</a> ] 4. OpenMP[ <a href="http://openmp.org/wp/">http://openmp.org/wp/</a> ] 5. MPI[ <a href="http://www.mpi-forum.org/">http://www.mpi-forum.org/</a> ]		

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

Cursul respecta Recomandarile IEEE and ACM Curricula pentru studii in Computer Science; Cursuri cu tematica similara exista in programele de studii ale majoritatii universitatilor din tara si strainatate.

Cursul defineste dobandirea unor abilitati care constituie avantaje evidentiate de potentialele firme angajatoare din domeniu.

### 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	In scopul evaluarii obtinerii competentelor specifice, fiecare student va dezvolta un proiect prin care va rezolva o problema cu specificatii date, folosind programare paralela(se va alege unul dintre limbajele discutate). Proiectul va avea pe langa implementare si o documentatie care va descrie metoda de paralelizare si evaluarea performantei.	colocviu	Proiect: -specificatia problemei si implementarea secventiala – 15%
10.5 Laborator			- proiectare paralela si evaluare teoretica – 25%  -Implementare si testare -40%  -prezentare proiect si intrebari legate de deciziile de proiectare si implementare - 20%
10.6 Standard minim de performanță			
Minim 5 la media finala.			

Data completării

.....

Titular de curs

.....Niculescu Virginia.....

Titular de laborator

.....Niculescu Virginia.....

Data avizării în departament

.....

Director de departament

.....