

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 Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclu de studii	Licență
1.6 Programul de studiu / Calificarea	Informatică - engleză

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	4	Din care: 3.2 curs	2	3.3 seminar/laborator	1la b+ 1pr
3.4 Total ore din planul de învățământ	48	Din care: 3.5 curs	24	3.6 seminar/laborator	24
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					28
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					40
Tutoriat					15
Examinări					14
Alte activități:					-
3.7 Total ore studiu individual		127			
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

Competențe profesionale	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.
Competențe transversale	<ul style="list-style-type: none"> - Abilitatea de a aplica tehnici specifice programării paralele în rezolvarea problemelor. - Abilitatea de a evalua creșterea de performanță obținută prin folosirea paralelizării. - Abilitatea de a implementa programe paralele.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

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

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 – tendințe actuale	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
2. Pipeline ca paradigmă de programare paralelă. Exemple – implementări concrete	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
3. Streaming programming Java streams. <i>Comparatie cu sabloanele pipeline si divide&impera</i>	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
4. Sabloane - <i>Task-Farm/Work-Pool/ Thread-Pool</i> Exemple; Java Executors; Fork-Join Pool; Work-stealing; Futures	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
5. Modelul PRAM (Parallel Random Access Machine)	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	

6. Rețele computationale. Exemplificari. Evaluarea performantei programelor paralele Scalabilitate. Evaluare teoretica si experimentală. - Tehnica reducerii ciclice par-impair	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7. Descompunerea geometrica a domeniului de date Algoritmi paraleli din algebra liniara. Exemple de algoritmi paraleli din teoria grafurilor. - Tehnica parcursurului Eulerian	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
8. Tehnici utilizate in constructia algoritmilor paraleli - Tehnica dublării recursive - Tehnica par-impair Algoritmi de sortare; (merge sort, quicksort); odd-even sort, counting sort; Bitonic sort	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
9. Modelul BSP	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
10. Modelul Actor	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
11. Hybrid approaches: MPI + threads, MPI + OpenMP	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
12. Colocviu: Prezentare 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 sablon de programare paralela adecvat. Evaluare teoretica a solutiei paralele	Dialog, dezbateri, explicatie,exemple.	
4. Implementarea solutiei paralele.	Dialog, dezbateri, explicatie,exemple.	
5. Testare si validare	Dialog, dezbateri, explicatie,exemple.	
6. Presentari ale proiectelor studentilor	Dialog, dezbateri, explicatie,exemple.	
Bibliografie <ol style="list-style-type: none"> 1. Parallel Programming in the .NET [https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx] 2. Scala Programming Language, [http://www.scala-lang.org] 3. Java Programming Language [www.oracle.com/us/technologies/java/overview] 4. OpenMP[http://openmp.org/wp/] 5. 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 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 definește dobândirea unor abilitati care constituie avantaje evidentiare 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/tehnologiile 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

.....

.....