

FIȘA DISCIPLINEI

Programare paralelă , distribuită

1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918”
1.2. Facultatea	Științe Exacte și Inginerești
1.3. Departamentul	Științe Exacte și Inginerești
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Licență
1.6. Programul de studii	Informatica

2. Date despre disciplină

2.1. Denumirea disciplinei	<i>Programare paralelă , distribuită</i>		2.2. Cod disciplină	INFO 301			
2.3. Titularul activității de curs	Domșa Ovidiu						
2.4. Titularul activității de seminar	Incze Arpad						
2.5. Anul de studiu	III	2.6. Semestrul	I	2.7. Tipul de evaluare (E/C/VP)	E	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	O

3. Timpul total estimat

3.1. Numar ore pe saptamana	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					50
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					50
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					50
Tutoriat					-
Examinări					28
Alte activități					-

3.7 Total ore studiu individual	108
3.8 Total ore din planul de învățământ	56
3.9 Total ore pe semestru	164
3.10 Numărul de credite	6

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<i>Programare imperativa si procedurala Programare orientată pe obiecte</i>
4.2. de competențe	<i>Algoritmi si structuri de date</i>

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<i>Sala dotata cu videoproiector/tabla</i>
--------------------------------	--

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> - C6.1. Identificarea conceptelor si modelelor de baza pentru sisteme de calcul si retele de calculatoare. Aprofundarea conceptelor de progmmare avansată în Java folosind Tread.. - C6.2. Identificarea si explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor si a retelelor. - C6.3. Utilizarea tehnicilor pentru instalarea, configurarea si administrarea sistemelor si retelelor. Identificarea problemelor abordate cu tehnicile și algoritmi studiați; - C6.4. Efectuarea de măsurători de performanță pentru timpi de răspuns, consum de resurse; stabilirea drepturilor de acces. Studentul absolvent al acestui curs va fi capabil să utilizeze în practică noțiunile de fire de execuție și procesare paralela. - C5.4. Evaluarea calitatii diferitelor sisteme de gestiune a bazelor de date din punctul de vedere al structurii, funcționalității si extensibilitatii.
Competențe transversale	<p>CT3 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> <p>Competențe cognitive: dobândirea de cunoștințe fundamentale și specifice privind conceptul de programare paralelă și concurență, capacitatea de identificare a aplicabilității algoritmilor studiați în probleme reale; înțelegerea necesității utilizării firelor de execuție în abordarea eficientă a problemelor: dobândirea de cunoștințe fundamentale privind conceptul de programare folosind fire de execuție;</p> <p>Competențe afectiv valorice: dezvoltarea capacității de analiză și înțelegere a unei probleme reale de complexitate ridicată și abordarea acesteia din perspectivă multiprocesor în mod eficient.</p> <p>Spirit de echipă (încurajarea studentului de a lucra în echipe de proiectare, analiză și programare); conștientizarea importanței cunoașterii și aprofundării metodelor studiate.</p>

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> - <i>Dezvoltarea gândirii algoritmice și a abilităților de a elabora algoritmi complecși.</i> - <i>Însușirea instrumentelor de bază în elaborarea algoritmilor paraleli.</i>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> - <i>Cunoașterea tipologiei specifice programării Tread-urilor in Java..</i> - <i>Deprinderea studenților cu utilizarea unui limbaj de programare evoluat în vederea implementării algoritmilor studiați.</i>

8. Conținuturi

Curs	Metode de predare	Observații
Principii generale de programare paralelă. Materiale introductive.	<i>Prelegere, discuții, exemplificări</i>	
Paralelismul din perspectiva arhitecturii calculatoarelor.	<i>Prelegere, discuții, exemplificări</i>	
Calcul paralel. Modele de utilizare a memoriei.	<i>Prelegere, discuții, exemplificări</i>	
Comparatie privind modalitatile de abordare a utilizarii in paralel a resurselor unui system. Pocesoarele si rezolvarea problemelor non deterministe.	<i>Prelegere, discuții, exemplificări</i>	4 ore
Design si analiza a algorimilor paraleli.	<i>Prelegere, discuții, exemplificări</i>	
Algoritmi numerici și ne numerici in perspectiva programarii paralele.	<i>Prelegere, discuții, exemplificări</i>	4 ore
Utiizarea memoriei in rezolvarea problemelor de programare paralelă si concurentă.	<i>Prelegere, discuții, exemplificări</i>	
Aplicatii vizuale. Utilizarea programarii paralele.	<i>Prelegere, discuții, exemplificări</i>	
Aplicații. Exemple.	<i>Prelegere, discuții, exemplificări</i>	
Sincronizarea proceselor. Aplicații.	<i>Prelegere, discuții, exemplificări</i>	
Interconectarea rețelelor.	<i>Prelegere, discuții, exemplificări</i>	
Analiza, evaluare, asigurarea feed-back.	<i>Prelegere, discuții, exemplificări</i>	
Bibliografie		
A. Grama, G. Karypis, V. Kumara and A. Gupta, "Introduction to Parallel Computing," 2nd Ed., Addison Wesley, 2003.		
Seminar-laborator		
Java Threads. Limbajul Java recapitulare Example cu Threads.	Lucrare practică de laborator	4 ore
Probleme - Utilizarea paralelă a memoriei - Probleme single processor si multiprocesor.	Lucrare practică de laborator	4 ore
Threads, clase, instanțieri, condiții, restricții.	Lucrare practică de laborator	
Soluții Tehnice	Lucrare practică de laborator	4 ore
Sincronizarea si excluderea. Prezentare generală	Lucrare practică de laborator	
Facilitatile programarii concurente in Java	Lucrare practică de laborator	
Object-oriented design patters for concurrency.	Lucrare practică de laborator	
Proiecte propuneri design, solutii	Lucrare practică de laborator	
Concurrent Programs	Lucrare practică de laborator	
Parallel computing and message passing	Lucrare practică de laborator	
Aplicații practice. Exemple de probleme practice	Lucrare practică de laborator	
Bibliografie		
1. Ernest SCHEIBER , PROGRAMARE CONCURENTA SI PARALEL-DISTRIBUITA IN JAVA, Ed. Albastra ,2007.		
2. A.Grama, G. Karypis, V. Kumara and A. Gupta, "Introduction to Parallel Computing," 2nd Ed., Addison Wesley, 2003.		

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

- *Programare aplicata in practica*

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Examen scris</i>	30%
	-	-	-
10.5 Seminar/laborator	<i>Verificare pe parcurs</i>	<i>Portofoliu de lucrări practice de laborator</i>	30%
	<i>-Proiect</i>	<i>Proiect practic</i>	- 40%
10.6 Standard minim de performanță: 5.00			

Data completării

.....

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în catedră

.....

Semnătura director de departament

.....