

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN ORADEA
1.2 Facultatea	FACULTATEA DE INGINERIE ELECTRICĂ ȘI TEHNOLOGIA INFORMAȚIEI
1.3 Departamentul	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.4 Domeniul de studii	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.5 Ciclul de studii	LICENȚĂ
1.6 Programul de studii/Calificarea	TEHNOLOGIA INFORMAȚIEI /INGINER

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare paralelă						
2.2 Titularul activităților de curs	Prof.univ.dr.ing. Zmaranda Doina						
2.3 Titularul activităților de laborator/proiect	Drd. Artenie Andrada						
2.4 Anul de studiu	IV	2.5 Semestrul	7	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	DS

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 laborator /proiect	1/1
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 laborator /proiect	14/14
Distribuția fondului de timp ore					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					2
Examinări					6
Alte activități.....					
3.7 Total ore studiu individual	44				
3.9 Total ore pe semestru	100				
3.10 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de competențe	Cunoașterea unui limbaj de programare orientat pe obiecte

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	- cursul se desfășoară față în față (sală de curs dotată cu calculator și videoproiector); prezentare bazată pe slide-uri
5.2. de desfășurare a laboratorului /proiectului	- laboratorul desfășoară față în față (sală de laborator dotată cu calculatoare și platforma .NET /Visual Studio) - prezenta obligatorie la toate laboratoarele 100%; - recuperarea laboratoarelor se face conform Regulamentului privind activitatea profesională a studenților în baza Sistemului European de Credite Transferabile (ECTS); se pot recupera pe parcursul semestrului maxim 30% din totalul activităților

6a. Competențele specifice acumulate	
Competențe profesionale	CP2. Proiectarea componentelor hardware, software și de comunicații CP3. Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor CP5. Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații
Competențe transversale	

6b. Rezultatele așteptate ale învățării	
Cunoștințe	Studentul cunoaște și identifică conceptele teoretice și mecanismele care stau la baza programării concurente/paralele și utilizează medii de programare (framework-uri) și limbaje/biblioteci specifice pentru implementarea de aplicații concrete care impun concurență și/sau paralelism (Visual Studio și .NET, biblioteca TPL)
Aptitudini	Studentul alege și utilizează mecanismele concurenței, conceptele fundamentale și abordările consacrate privind structurarea aplicațiilor concurente; specifică cerințe, analizează, elaborează, dezvoltă și testează aplicații practice specifice implementărilor bazate pe abordări concurente și paralelism;
Responsabilitate și autonomie	Studentul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale; studentul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei

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

7.1 Obiectivul general al disciplinei	În contextul diversității aplicațiilor software existente, cu necesități stringente de performanță, cursul abordează aspectele specifice din domeniul proiectării și dezvoltării aplicațiilor concurente
7.2 Obiectivele specifice	- Cursul își propune prezentarea conceptelor teoretice și a mecanismelor care stau la baza programării concurente prezentând conceptele fundamentale ale programării concurente precum și structura generală a aplicațiilor concurente. Exemplificările sunt realizate utilizând platforma .NET și C# respectiv TPL .NET, fără a restrânge însă generalitatea conceptelor prezentate - Laboratorul familiarizează studenții cu aspecte practice privind proiectarea și implementarea de aplicații concurente utilizând platforma .NET (Task Parallel Library și programare asincronă)

8. Conținuturi

8.1.Curs	Metode de predare	Observații
Procesare concurentă și paralelă. Identificarea paralelismului potențial în procesul de proiectare al aplicației. Legea lui Amdahl. Legea lui Gustafson.	Expunere liberă, cu prezentarea cursului pe slide-uri, față în față; discuții libere;	2
Gestionarea thread-urilor concurente. Crearea thread-urilor. Proprietăți. Creare. Tratarea terminării anormale/corecte a firelor de execuție.		2
Sincronizarea thread-urilor pentru partajarea datelor comune. Excluderea mutuală. Sincronizarea comunicării între thread-uri.		4

Utilizarea unui ThreadPool în .NET. Execuția în ThreadPool.		2
Șabloane utilizate în programarea concurentă: șablonul producător/consumator, șablonul Pipelining.		2
Conceptele de thread-safety și afinitatea la thread. Responsivitatea aplicațiilor : exemplu UI.		2
API-uri pentru multithreading – .NET Parallel Library. Scalabilitatea aplicațiilor concurente. Avantajele utilizării API-urilor pentru multithreading.		2
Construcții pentru asigurarea paralelismului task-urilor. Task Parallel Library (TPL). Clasa Parallel. Parallel LINQ (PLINQ).		4
Colecții concurente în .NET.		2
Colecții producător/consumator în .NET.		2
Modelul de programare asincronă - async/await.		4

Bibliografie

- <https://learn.microsoft.com/en-us/dotnet/standard/parallel-programming/>
- <https://dotnettutorials.net/lesson/introduction-to-parallel-and-asynchronous-programming-in-csharp/>
- <https://www.bytehide.com/blog/step-by-step-learn-parallel-programming-with-c-and-net-2024>
- https://dev.to/soham_galande/handling-concurrency-in-c-a-guide-to-multithreading-and-parallelism-446
- <https://wirefuture.com/post/mastering-net-a-deep-dive-into-task-parallel-library>
- Christian Nagel – C# and .NET 2021 Edition, Wiley & Sons, ISBN: 978-1-119-79720-3, 2021
- Riccardo Terrell, Concurrency in .NET. Modern patterns of concurrent and parallel programming, ISBN 9781617292996, Manning Publications, 2018
- Stephen Cleary - *Concurrency in C# Cookbook Paperback* – 2014, ISBN-13: 860-1402245338 ISBN-10: 1449367569 Edition: 1st
- Rodney Ringler - *C# Multithreaded and Parallel Programming Paperback* – 2014, ISBN-10: 184968832X, ISBN-13: 978-1849688321
- Zaharie Dorin, **Zmaranda Doina** - *Dezvoltarea aplicațiilor software utilizând platforma .NET*, Editura ASE București, ISBN 978-606-505-547-6, 506pg., 2012
- https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/_layouts/15/onedrive.aspx?isAscending=true&id=%2Fpersonal%2Frodica%5Fzmaranda%5Fdidactic%5Fuoradea%5Fro%2FDocuments%2FPP%2F%60Curs%5FPP&sortField=LinkFilename&view=0

8.2. Laborator	Metode de predare	Observații
Conceptele de bază ale concurenței în .NET: fire de execuție. Crearea firelor de și pornirea firelor de execuție	Studentii primesc temele pentru laborator cu cel puțin o săptămână înainte, și le studiază. La începutul laboratorului se discută și se analizează modalitățile de rezolvare a aplicațiilor propuse, folosind videoproiectorul. Pe urmă, studenții realizează partea practică a lucrării (problemele propuse) sub îndrumarea cadrului didactic.	2
Utilizarea unui ThreadPool. Întreruperea thread-urilor		2
Sincronizarea thread-urilor și excludere mutuală. Utilizarea de date în comun		2
Notificări între thread-uri. Șablonul producător/consumator		2
Afinitatea la thread		2
Utilizarea paralelismului datelor		4
TPL (Task Parallel Library) – Crearea, execuția și întreruperea task-urilor		2
TPL (Task Parallel Library) – Clasa Parallel și PLINQ		2
Colecții concurente în .NET. Colecții de tipul producător/consumator.		2
Modelul de programare asincronă		4
Incheierea situației la laborator, verificări și recuperare		4

Bibliografie

1. https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/_layouts/15/onedrive.aspx?isAscending=true&id=%2Fpersonal%2Frodica%5Fzmaranda%5Fdidactic%5Fuoradea%5Fro%2FDocuments%2FPP%2FLaborator%5FPP&sortField=LinkFilename&view=0

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

- O disciplină similară se regăsește și în curricula specializării de Calculatoare din cadrul Universității Politehnica Timișoara. Cunoașterea conceptelor de bază ale programării paralele / concurente, prezentate în cadrul acestei discipline, reprezintă o competență importantă, solicitată de companiile de software

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- pentru nota 5 este necesară cunoașterea noțiunilor fundamentale solicitate în subiecte, fără a prezenta detalii asupra acestora, care să însumeze în total 5 puncte (4 puncte + 1 punct din oficiu) - pentru nota 10, este necesară cunoașterea amănunțită a tuturor subiectelor (10 puncte)	Examen scris - evaluarea se face față în față Studentii primesc spre rezolvare fiecare un subiect cu întrebări care testează însușirea conceptelor teoretice prezentate în cadrul cursului. Fiecare întrebare are un punctaj; în total întrebările însumează un punctaj de 9 de puncte; se acordă 1 punct din oficiu.	40 %
10.5 Laborator	- pentru nota 5, realizarea unui procent de minim 50% din aplicațiile propuse, explicarea modalității de implementare și a arhitecturii acestora - pentru nota 10, finalizarea corectă și completă tuturor aplicațiilor propuse	Aplicație practică - evaluarea se face față în față La fiecare laborator studenții sunt evaluați pe baza activității (răspunsuri la întrebări, propuneri de implementare, etc.), evaluări care se concretizează la finalul laboratorului într-o notă pe activitatea de laborator din cursul semestrului. De asemenea, până în ultima oră de laborator studenții finalizează și predau toate aplicațiile practice propuse în cadrul laboratorului. Media dintre nota primită pentru aplicațiile practice și nota de la activitatea de la laborator va reprezenta nota finală de laborator	60%

Calcul **Nota disciplină**: 0.60 Nota finală laborator + 0.40 Nota examen scris

Condiții de participare la examenul final: Nota finală laborator ≥ 5

Condiții de promovare: **Nota disciplină** ≥ 5

10.6 Standard minim de performanță

- cunoașterea și înțelegerea conceptelor de bază specifice proiectării și implementării unei aplicații concurente
- cunoașterea și înțelegerea structurii generale a aplicațiilor concurente și familiarizarea cu șabloanele de proiectare specifice domeniului
- cunoașterea aspectelor specifice proiectării și implementării unei aplicații concurente: mecanismele concurenței, conceptele fundamentale, structurarea aplicațiilor concurente precum și a modalității de aplicare a conceptelor teoretice în cadrul dezvoltării unei aplicații practice, concrete
- utilizarea unui API pentru multithreading

Data completării
02.09.2025

Semnătura titularului de curs,
Prof. dr. ing. Zmaranda Doina
dzmaranda@uoradea.ro

Semnătura titularului de laborator
Drd. Artenie Andrada
andradaartenie@uoradea.ro

Data avizării în departament

Semnătura directorului de departament,
Conf.univ. dr. Elisa Valentina Moisi
emoisi@uoradea.ro

Data avizării în consiliul facultății

Semnătură Decan,
Conf. univ. dr. ing. Eugen Gergely
egergely@uoradea.ro