FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**UNIVERSITATEA DIN ORADEA**
1.2 Facultatea	**FACULTATEA DE INGINERIE ELECTRICĂ ŞI TEHNOLOGIA INFORMAŢIEI**
1.3 Departamentul	**CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI**
1.4 Domeniul de studii	**CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI**
1.5 Ciclul de studii	**LICENŢĂ**
1.6 Programul de studii/Calificarea	**TEHNOLOGIA INFORMAŢIEI /INGINER**

2. Date despre disciplină

2.1 Denumirea disciplinei			**PROGRAMARE PARALELĂ**				
2.2 Titularul activităţilor de curs			Prof.univ.dr.ing. Zmaranda Doina				
2.3 Titularul activităţilor de laborator/proiect			Drd. Criste Evelin-Medeea				
2.4 Anul de studiu	**IV**	2.5 Semestrul	**7**	2.6 Tipul de evaluare	**Ex**	2.7 Regimul disciplinei	**(I)**

(I) Impusă; (O) Opțională; (F) Facultativă

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

3.1 Număr de ore pe săptămână	**4**	din care: 3.2 curs	**2**	3.3 laborator /proiect	**2**
3.4 Total ore din planul de învăţământ	**56**	din care: 3.5 curs	**28**	3.6 laborator /proiect	**28**
Distribuţia fondului de timp ore					ore
Studiul după manual, suport de curs, bibliografie şi notiţe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri					14
Tutoriat					2
Examinări					6
Alte activităţi.....					

3.7 Total ore studiu individual	**44**
3.9 Total ore pe semestru	**100**
3.10 Numărul de credite	**4**

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	
4.2 de competenţe	Cunoaşterea unui limbaj de programare orientat pe obiecte

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	- cursul se desfăsoară față în față (sală de curs dotată cu calculator şi videoproiector); prezentare bazată pe slide-uri
5.2. de desfăşurare a laboratorului /proiectului	- laboratorul se desfășoară față în față' - sală de laborator dotată cu calculatoare conectate în rețea și software adecvat - 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 activităţilor

6.1. Competenţele specifice acumulate	
Competenţe profesionale	**CP2.** Proiectarea componentelor hardware, software şi de comunicaţii **CP3.** Soluţionarea problemelor folosind instrumentele ştiinţei şi ingineriei calculatoarelor **CP5.** Proiectarea, gestionarea ciclului de viaţă, integrarea şi integritatea sistemelor hardware, software şi de comunicaţii
Competenţe transversale	

6.2. Rezultatele aşteptate ale învăţării	
Cunoștințe	Studentul cunoaște și identifică conceptele teoretice şi mecanismele care stau la baza abordărilor și programării concurente și paralele; studentul știe să utilizeze 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 Task Parallel Library)
Aptitudini	Studentul alege şi utilizează mecanismele concurenţei, conceptele fundamentale și abordările consacrate privind structurarea aplicaţiilor 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 (reieşind din grila competenţelor specifice acumulate)

7.1 Obiectivul general al disciplinei	În contextul diversităţii aplicaţiilor software existente, cu necesităţi stringente de performanţă și responsivitate, cursul abordează aspectele specifice din domeniul proiectării şi dezvoltării aplicaţiilor concurente
7.2 Obiectivele specifice	▪ Cursul îşi propune prezentarea conceptelor teoretice şi a mecanismelor care stau la baza programării concurente/paralelismului prezentând conceptele fundamentale ale programării concurente precum şi structura generală a aplicaţiilor concurente. Exemplificările sunt realizate utilizând platforma .NET şi C# respectiv TPL .NET, fără a restrânge însă generalitatea conceptelor prezentate ▪ Laboratorul familiarizează studenţii cu aspecte practice privind proiectarea şi implementarea de aplicaţii concurente utilizând platforma .NET (Task Parallel Library şi programare asincronă)

8. Conţinuturi

8.1.Curs	Metode de predare	Observaţii
Procesare concurentă şi paralelă. Identificarea paralelismului potenţial în procesul de proiectare al aplicaţiei. 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 execuţie.		2
Sincronizarea thread-urilor pentru partajarea datelor datelor comune. Excluderea mutuală. Sincronizarea comunicării între thread-uri.		4
Utilizarea unui ThreadPool în .NET. Execuţia în ThreadPool.		2

Şabloane utilizate în programarea concurentă: şablonul producător/consumator, şablonul Pipelining.		2
Conceptele de thread-safety şi afinitatea la thread. Responsivitatea aplicaţiilor : exemplu UI.		2
API-uri pentru multithreading – .NET Parallel Library. Scalabilitatea aplicaţiilor 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

1. https://learn.microsoft.com/en-us/dotnet/standard/parallel-programming/
2. https://dotnettutorials.net/lesson/introduction-to-parallel-and-asynchronous-programming-in-csharp/
3. https://dev.to/soham_galande/handling-concurrency-in-c-a-guide-to-multithreading-and-parallelism-446
4. https://www.c-sharpcorner.com/article/parallel-programming-in-c-sharp-the-complete-guide-to-task-parallel-library-tpl-fo
5. https://www.geeksforgeeks.org/c-sharp/task-parallel-library-in-c/
6. https://blog.stackademic.com/mastering-tpl-best-practices-for-task-parallel-library-76f707ad96b5
7. https://clouddevs.com/dot-net/task-parallel-library/
8. https://medium.com/@stanislousvanderputt/asynchronous-programming-and-task-parallel-library-for-responsive-and-efficient-development-net-ab2f280d9c1b
9. https://dev.to/soham_galande/handling-concurrency-in-c-a-guide-to-multithreading-and-parallelism-446
10. Christian Nagel – C# and .NET 2021 Edition, Wiley & Sons, ISBN: 978-1-119-79720-3, 2021
11. Riccardo Terrell, Concurrency in .NET. Modern patterns of concurrent and parallel programming, ISBN 9781617292996, Manning Publications, 2018
12. Stephen Cleary - *Concurrency in C# Cookbook Paperback* – 2014, ISBN-13: 860-1402245338 ISBN-10: 1449367569 Edition: 1st
13. Rodney Ringler - *C# Multithreaded and Parallel Programming Paperback* – 2014, ISBN-10: 184968832X, ISBN-13: 978-1849688321
14. Zaharie Dorin, **Zmaranda Doina** - *Dezvoltarea aplicaţiilor software utilizând platforma .NET*, Editura ASE Bucureşti, ISBN 978-606-505-547-6, 506pg., 2012

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

Bibliografie

1. https://www.geeksforgeeks.org/c-sharp/task-parallel-library-in-c/
2. https://blog.stackademic.com/mastering-tpl-best-practices-for-task-parallel-library-76f707ad96b5
3. https://clouddevs.com/dot-net/task-parallel-library/
4. https://medium.com/@stanislousvanderputt/asynchronous-programming-and-task-parallel-library-for-responsive-and-efficient-development-net-ab2f280d9c1b
5. https://dev.to/soham_galande/handling-concurrency-in-c-a-guide-to-multithreading-and-parallelism-446

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

- O disciplină similară se regăseşte şi în curricula specializării de Calculatoare din cadrul Universităţii Politehnica Timişoara. Cunoaşterea conceptelor de bază ale programării paralele / concurente, prezentate în cadrul acestei discipline, reprezintă o competenţă 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 cunoaşterea noţiunilor 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 cunoaşterea amănunţită a tuturor subiectelor (10 puncte)	**Examen scris -** evaluarea se face față în față Studenţii primesc spre rezolvare fiecare un subiect cu întrebări care testează însuşirea 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 aplicaţiile propuse, explicarea modalităţii de implementare şi a arhitecturii acestora - pentru nota 10, finalizarea corectă și completă tuturor aplicaţiilor propuse	**Aplicaţie practică -** evaluarea se face față în față La fiecare laborator studenţii sunt evaluaţi 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 studenţii finalizează şi predau toate aplicațiile practice propuse în cadrul laboratorului. Media dintre nota primita pentru aplicaţiile practice şi 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 Conditii de participare la examenul final: Nota finală laborator ≥ 5 Conditii de promovare: **Nota disciplină** ≥ 5			
10.6 Standard minim de performanţă			
• cunoaşterea şi înţelegerea conceptelor de bază specifice proiectării şi implementării unei aplicaţii concurente • cunoaşterea şi înțelegerea structurii generale a aplicaţiilor concurente şi familiarizarea cu şabloanele de proiectare specifice domeniului • cunoaşterea aspectelor specifice proiectării şi implementării unei aplicaţii concurente: mecanismele concurenței, conceptele fundamentale, structurarea aplicaţiilor concurente precum şi a modalității de aplicare a conceptelor teoretice în cadrul dezvoltării unei aplicații practice, concrete • utilizarea unui API pentru multithreading			

Data completării 02.09.2026	Semnătura titularului de curs, Prof. dr. ing. Zmaranda Doina dzmaranda@uoradea.ro	Semnătura titularului de laborator Drd. Criste Evelin-Medeea

Data avizării în departament 17.09.2026	Semnătura directorului de departament, Conf.univ. dr. Elisa Valentina Moisi emoisi@uoradea.ro

Data avizării în consiliul facultății 18.09.2026	Semnătură Decan, Conf. univ. dr. ing. Eugen Gergely egergely@uoradea.ro