# FIŞA DISCIPLINEI

**1. Date despre program**

| | |
|---|---|
| 1.1 Instituția de învățământ superior | Universitatea Transilvania din Brașov |
| 1.2 Facultatea | Matematică și informatică |
| 1.3 Departamentul | Matematică și informatică |
| 1.4Domeniul de studii deMASTERAT[1)] | Matematică și științe ale naturii |
| 1.5Ciclul de studii[2)] | Masterat |
| 1.6 Programul de studii/ Calificarea | Tehnologii moderne în ingineria sistemelor software / Informatician |

**2.Date despredisciplină**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2.1 Denumirea disciplinei | PROGRAMARE FUNCȚIONALĂ NESECVENȚIALĂ | | | | | | | |
| 2.2 Titularul activităților de curs | VASILESCU ANCA | | | | | | | |
| 2.3 Titularul activităților de seminar/ laborator/ proiect | VASILESCU ANCA | | | | | | | |
| 2.4 Anul de studiu | 2 | 2.5 Semestrul | 3 | 2.6 Tipul de evaluare | E | 2.7 Regimul disciplinei | Conținut[3)] | DAP |
| | | | | | | | Obligativitate[3)] | Di |

**3.Timpultotal 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/ proiect | **0/2/0** |
| 3.4 Total ore din planul de învățământ | 56 | din care: 3.5 curs | 28 | 3.6 seminar/laborator/ proiect | 0/28/0 |

| | |
|---|---|
| Distribuția fondului de timp | ore |
| Studiul după manual, suport de curs, bibliografie şi notițe | 30 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren | 40 |
| Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii şi eseuri | 30 |
| Tutoriat | 14 |
| Examinări | 5 |
| Alte activități..... | |
| **3.7 Total ore de activitate a studentului** | 119 |

| | |
|---|---|
| **3.8 Total ore pe semestru** | 175 |
| **3.9 Numărul de credite[5)]** | 7 |

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

| | |
|---|---|
| 4.1 de curriculum | • Concepte specifice pentru programarea imperativă și programarea funcțională<br>• Repere cu privire la dezvoltarea aplicațiilor software şi a sistemelor informatice<br>• Noțiuni de scriere academică, de etică științifică și profesională |
| 4.2 de competențe | • Competențe generale și de specialitate conform programului de studiu de Licență absolvit<br>• Abilități de cercetare la nivelul unui absolvent de ciclu de licență |

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

| | |
|---|---|
| 5.1 de desfăşurare a cursului | |
| 5.2 de desfăşurare a seminarului/ laboratorului/ proiectului | • Sală de laborator cu resurse educaționale şi informaționale specifice – calculatoare, conexiune de rețea, internet |

**6. Competențe specifice acumulate (conform grilei de competențe din planul de învățământ)**

| | |
|---|---|
| Competențe profesionale | CP 1. Programarea în limbaje de nivel înalt<br>RI 1.2. Absolventul poate să utilizeze biblioteci și framework-uri pentru îmbunătățirea performațelor și a funcționalității aplicațiilor software<br>RI 1.4. Absolventul poate să aplice principiile și metodele programării funcționale și nesecvențiale folosind tehnologii specifice<br>CP 2. Dezvoltarea și întreținerea aplicațiilor informatice<br>RI 2.1 Absolventul poate să elaboreze proiecte și lucrări informatice folosind limbaje specifice<br>RI 2.3. Absolventul poate să lucreze în echipă pentru a dezvolta aplicații informatice folosind limbaje specifice |
| Competențe transversale | CT 1. Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și norelor de etică profesională<br>RI 1.2. Absolvenții vor fi capabili să desfășoare activități creatoare, să se dezvolte profesional și să abordeze noi domenii, adaptându-se cerințelor nou apărute<br>CT 3. Utilizare 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<br>RI 3.Absolvenții vor utiliza instrumente și tehnici eficiente pentru a comunica informații complexe într-un mod clar și concis |

**7. Obiectivele disciplinei(reieşind din competențele specifice acumulate)**

| | |
|---|---|
| 7.1 Obiectivul general al disciplinei | Utilizarea bazelor teoretice şi aplicative ale informaticii pentru transmiterea conceptelor specifice programării funcționale orientată către paralelism și concurență, în vederea utilizării eficiente în activitatea viitorului informatician aacestor caracteristici la nivelul unui limbaj de programare funcțională modern |
| 7.2 Obiectivele specifice | • Coroborarea elementelor specifice programării funcționale cu cele ale programării nesecvențiale (paralelă / concurentă / distribuită) pentru dezvoltarea unor aplicații moderne, care să răspundă optim cerințelor<br>• Identificarea situațiilor problemă care se pot rezolva eficient prin aplicații funcționale nesecvențiale<br>• Identificarea și valorificarea în dezvoltarea de proiecte a aspectelor funcționale și nesecvențiale la nivelul limbajelor de programare nespecifice |

**8. Conținuturi**

| 8.1 Curs | Metode de predare | Număr de ore | Observații |
|---|---|---|---|
| Programare imperativă față de programare funcțională. Programare secvențială față de programare distribuită. Caracterizare. Avantaje. Dezavantaje | Curs interactiv<br>Prelegere<br>Dialog<br>Dezbatere | 2 | |
| Elemente de programare funcțională. Haskell - limbaj de programare funcțională pur. | | 8 | |
| Suportul pentru programare funcțională al altor limbaje de programare (Python, C#, F#). Abordare comparativă. | | 4 | |
| Sisteme distribuite. Algoritmi distribuiți. Elemente de programare nesecvențială (paralelă, concurentă, distribuită) | | 4 | |
| Dezvoltarea aplicațiilor nesecvențiale cu limbaje de programare funcționale. Caracterizare. Avantaje. Dezavantaje | | 2 | |
| Suportul limbajului *Haskell* pentru paralelism și concurență. | | 4 | |

| | | | |
|---|---|---|---|
| Suportul altor limbaje pentru dezvoltarea aplicațiilor funcționale distribuite – Python, C#, F# | | | |
| Specificul proiectelor software funcționale nesecvențiale. Elemente de paralelism, concurență, caracter distribuit în proiecte abordate funcțional. | | 4 | |
| 8.2 Seminar/ laborator/ proiect | Metode de predare-învățare | Număr de ore | Observații |
| Organizarea activităților de laborator. Identificarea nivelului de cunoștințe anterioare în domeniu ale studenților | Studiu de caz | 2 | |
| Specificul unui limbaj de programare funcțional pur – Haskell. Elemente de limbaj. Dezvoltarea de aplicații funcționale elementare înHaskell | Asaltul de idei | 8 | |
| Comparație cu elementele de programare funcțională oferite de alte limbaje de programare – Python, C#, F# | Rezolvare de probleme | 4 | |
| Specificul programării nesecvențiale. Suportul limbajelor de programare de nivel înalt pentru paralelism, concurență, abordare distribuită. | Lucru în grup | 4 | |
| Abordarea funcțională a comunicării și sincronizării în sisteme paralele, concurente, distribuite. | Problematizare | 6 | |
| Dezvoltarea de proiecte software funcționale nesecvențiale. | Proiect | 4 | |

Bibliografie


1. Alexander GRANIN, *Functional Design and Architecture: Examples in Haskell*, Manning Pub, 2024, 1st edhttps://books.google.ro/books?id=XYMoEQAAQBAJ
2. Enrico BUONANNO, *Functional Programming in C#, Second Edition*, Manning Pub, 2022, 2nd ed.
3. Wojciech TUREK et al., *Special issue on Parallel and distributed computing based on the functional programming paradigm*, Concurrency and Computation Practice and Experience, August 2018
4. Vitaly BRAGILEVSKY, *Haskell in Depth*, Manning Pub, 2021
5. John REPPY, *Concurrency and Parallelism in Functional Programming Languages*, Spring Lectures, 2017, https://www.classes.cs.uchicago.edu
6. Graham HUTTON, *Programming in Haskell*, Cambridge University Press, 2nd edition, 2016
7. Alejandro Serano MENA, *Beginning Haskell. A Project Based Approach*, Apress, 2014
8. Simon MARLOW, *Parallel and Concurrent Programming in Haskell*, O'Reilly Media Inc., 2013
9. Bryan O'SULLIVAN et al., *Real World Haskell*, O'Reilly Media Inc., 2009
10. Mihai GONTINEAC, *Programare Funcțională - O introducere utilizând limbajul Haskell*, Ed. Al. Myller, Iaşi, 2006
11. F.W. BURTON, *Functional Programming for Concurrent and Distributed Computing*, The Computer Journal, Vol. 30, No. 5, 1987, 437-450


**9. Coroborarea conținuturilor disciplinei cu aşteptările reprezentanților comunităților epistemice, ale asociaților profesionale şi ale angajatorilor reprezentativi din domeniul aferent programului**

| |
|---|
| Conținuturile sunt în acord cu direcțiile de învățare şi cercetare propuse de asociațiile profesionale reprezentative - IEEE, ACM. |

**10.Evaluare**

| Tip de activitate | 10.1 Criterii de evaluare | 10.2 Metode de evaluare | 10.3 Pondere din nota finală |
|---|---|---|---|
| 10.4 Curs | • Atingerea obiectivelor educaționale | Evaluare finală prin probă scrisă | 25% |

| | | | |
|---|---|---|---|
| 10.5 Laborator | specifice ale disciplinei<br>• Acumularea competențelor vizate prin conținutul disciplinei | Evaluare formativă prin rapoarte tehnice, rapoarte de cercetare, proiecte, aplicații software, activități participative | 75% |
| 10.6 Standard minim de performanță | | | |
| • prezentarea la termene a activităților incluse în evaluarea formativă<br>• utilizarea corectă a structurilor elementare specifice limbajului de programare funcțională (*Haskell, Python*), pentru implementarea aspectelor nesecvențiale | | | |

Prezenta Fișă de disciplină a fost avizată în ședința de Consiliu de departament din data de 26.09.2024 și aprobată în ședința de Consiliu al facultății din data de 26.09.2024.

| | |
|---|---|
| **Conf. Dr. Ion Gabriel STAN,**<br>**.....**<br>**Decan** | **Conf. Dr. Nicușor MINCULETE,**<br>**.....**<br>**Director de departament** |
| **Lector Dr. Anca VASILESCU,**<br>**.....**<br>**Titular de curs** | **Lector Dr. Anca VASILESCU,**<br>**.....**<br>**Titular de laborator** |

Notă:

[1)] Domeniul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat (se completează conform cu Nomenclatorul domeniilor şi al specializărilor/ programelor de studii universitare în vigoare);

[2)] Ciclul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat;

[3)] Regimul disciplinei (conținut) - se alege una din variantele**: DF** (disciplină fundamentală)/ **DD** (disciplină din domeniu)/ **DS** (disciplină de specialitate)/ **DC** (disciplină complementară) - pentru nivelul de licență; **DAP**(disciplină de aprofundare)/ **DSI** (disciplină de sinteză)/ **DCA** (disciplină de cunoaştere avansată) - pentru nivelul de masterat;

[4)] Regimul disciplinei (obligativitate) - se alege una din variantele: **DI**(disciplină obligatorie)/ **DO** (disciplină opțională)/ **DFac**(disciplină facultativă);

5)Un credit este echivalent cu 25 – 30 de ore de studiu (activități didactice şi studiu individual).