FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Sapientia" din municipiul Cluj-Napoca
1.2. Facultatea	Științe Tehnice și Umaniste din Târgu Mureș
1.3. Domeniul de studii	Calculatoare și tehnologia informației
1.4. Ciclul de studii	Licență
1.5. Programul de studiu	Calculatoare
1.6. Calificarea	Inginer

2. Date despre disciplină

2.0. Departamentul				Matematică - Informatică			
2.1 Denumirea disciplinei				Programare logică și programare funcțională			
2.2 Titularul disciplinei / activităților de curs				Lector dr. Márton Gyöngyvér			
2.3 Titularul (ii) activităților de	seminar			-			
	laborator			Lector dr. Márton Gyöngyvér			
	proiect			-			
2.4 Anul de studiu	2	2.5 Semestrul	4	2.6. Tipul de evaluare	C	2.7 Regimul disciplinei	DI

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/ 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
a) Studiul după manual, suport de curs, bibliografie și notițe	8
b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	4
c) Pregătire laboratoare, teme, referate, portofolii și eseuri	5
d) Tutoriat	-
e) Examinări	2
f) Alte activități:	-

3.7 Total ore studiu individual	19
3.8 Total ore pe semestru	75
3.9 Numărul de credite	3

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	•
4.2 de competențe	•

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	• Sală de curs cu tablă și videoproiector
5.2 De desfășurare a seminarului/laboratorului/proiectului	• Sală de laborator dotată corespunzător: calculatoare, rețea, legătură la Internet, videoproiector, tablă

6. Competențele specifice acumulate

Competențe profesionale	• C2. Proiectarea componentelor hardware, software și de comunicații • C2.1 Descrierea structurii și funcționării componentelor hardware, software și de comunicații • C2.2 Explicarea rolului, interacțiunii și funcționării componentelor sistemelor hardware, software și de comunicații • C2.3 Construirea unor componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii • C2.4 Evaluarea caracteristicilor funcționale și nefuncționale ale componentelor hardware, software și de comunicații, pe baza unor metrici • C2.5 Implementarea componentelor sistemelor hardware, software și de comunicație
Competențe transversale	•

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Familiarizarea cu paradigma programării funcționale. Folosirea noțiunilor teoretice învățate în aplicații concrete. Formarea și dezvoltarea gândirii pe baza logicii programării funcționale. • Studenții vor fi capabili să dezvolte aplicații software, folosind paradigma programării funcționale.
7.2 Obiectivele specifice	• Să deprindă studentul cu paradigmele programării funcționale: - să definească, să explice, și să aplice conceptele fundamentale ale programării funcționale într-un limbaj de programare, de ex. Haskell • Să înțeleagă bazele teoretice ale programării funcționale, • Să inducă ideea utilizării acestor paradigme în funcție de necesitățile aplicațiilor.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Introducere, specificații, concepte introductive și generale legate de programarea funcțională în limbajul de programare Haskell	Expunerea interactivă Dezbaterea Problematizarea	**4 ore**
Tipuri de date: simple, complexe. Variabile, expresii în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**2 ore**
Funcții și operatori. Structuri de control, recursivitate în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**2 ore**
Liste, funcții pe liste. Funcții din biblioteca standard în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**4 ore**
Operații de tip mulțime, expresii lambda. Funcții de ordin superior în Haskell	Expunerea interactivă Dezbaterea	**2 ore**

	Problematizarea	
Date infinite, evaluarea leneșă. Structuri de date si algoritmi în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**4 ore**
Operații I/O în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**4 ore**
Eficiența algoritmilor	Expunerea interactivă Dezbaterea Problematizarea	**2 ore**
Noțiuni legate de baze de date în Haskell	Expunerea interactivă Dezbaterea Problematizarea	**2 ore**
Test de evaluare a cunoștințelor		**2 ore**

Bibliografie obligatorie

1. Pagina Web al titularului disciplinei: https://ms.sapientia.ro/~mgyongyi/funk_log.html

Bibliografie opțională

1. Simon Thompson, Haskell: The Craft of Functional Programming (3rd Edition), Addison Wesley, 2011.
2. Miran Lipovaca, Learn You a Haskell for Great Good!: A beginner's guide, 2011.
3. Graham Hutton, Programming in Haskell, Cambridge University Press, 2nd edition, 2016.
4. Bryan O'Sullivan, John Goerzen, Don Stewart, Real World Haskell, O'Reilly Media, 2008.
5. Horváth Zoltán: Programnyelvek/ A funkcionális programozási nyelvek elemei
6. Mihai Gontineac, Programare funcțională: o întroducere utilizând limbajul Haskell, Ed. Alexandru Myller, 2006.
7. Csörnyei Zoltán, Lambda-kalkulus, Typotex, Budapest, 2007.

8.2 Seminar – nu este cazul	Metode de predare	Observații
8.3 laborator		
Limbajul de programare Haskell • Noțiuni introductive • Tipuri de date: simple, complexe • Variabile, expresii	Exercițiul Conversația euristică Modelarea Documentarea	**2 ore**
• Funcții și operatori • Structuri de control, recursivitate,	Exercițiul Conversația euristică Modelarea Documentarea	**2 ore**
• Liste, funcții pe liste, • Funcții din biblioteca standrad	Exercițiul Conversația euristică Modelarea Documentarea	**2 ore**
• Operații de tip mulțime, expresii lambda • Funcții de ordin superior	Exercițiul Conversația euristică Modelarea Documentarea	**4 ore**
• Date infinite, evaluarea leneșă • Structuri de date si algoritmi • Operații I/O (tastatura, monitor)	Exercițiul Conversația euristică Modelarea Documentarea	**4 ore**
• Operații I/O (fișiere de tip text, binar) • Moduli	Exercițiul Conversația euristică	**4 ore**

	Modelarea Documentarea	
• Eficiența algoritmilor • Noțiuni legate de baze de date	Exercițiul Conversația euristică Modelarea Documentarea	**2 ore**
Test de evaluare a cunoștințelor		**4 ore**

Bibliografie

1. Simon Thompson, Haskell: The Craft of Functional Programming (3rd Edition), Addison Wesley, 2011.
2. Miran Lipovaca, Learn You a Haskell for Great Good!: A beginner's guide, 2011.
3. Graham Hutton, Programming in Haskell, Cambridge University Press, 2nd edition, 2016.
4. Bryan O'Sullivan, John Goerzen, Don Stewart, Real World Haskell, O'Reilly Media, 2008.
5. Horváth Zoltán: Programnyelvek/ A funkcionális programozási nyelvek elemei
6. Mihai Gontineac, Programare funcțională: o întroducere utilizând limbajul Haskell, Ed. Alexandru Myller, 2006.
7. Csörnyei Zoltán, Lambda-kalkulus, Ed. Typotex, Budapest, 2007
8. Pagina Web al titularului disciplinei: http://www.ms.sapientia.ro/~mgyongyi

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

- Conținutul disciplinei este în concordanță cu cerințele planului de învățământ, cu conținutul disciplinei predate la alte Universități din țară (Universitatea Tehnică din Cluj-Napoca, Universitatea Politehnică din Timișoara, Universitatea Transilvania din Brașov) și de peste hotare (Universitatea Tehnică și de Științe Economice din Budapesta)
- În sprijinul obiectivelor de business ale companiilor de a dezvolta produse software robuste si minimizarea erorilor, cursul pune accent pe corectitudinea dezvoltării programelor.
- Conținutul cursului este coroborat cu tendințele din industrie de adoptare a tehnicilor de programare funcțională.

10. Evaluare

A. Condiții de îndeplinit pentru prezentarea la evaluare:

Efectuarea tuturor lucrărilor de laborator și promovarea testului de evaluare a cunoștințelor la laborator

B. Criterii, metode și ponderi în evaluare:

Tip activitate		10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs		înțelegerea problemei, abilitatea de a analiza funcționarea și de a efectua calcule într-un circuit electronic	Evaluare scrisă (finală în presesiune)	50%
10.5	Seminar	-	-	-
	Laborator	înțelegerea problemei, abilitatea de a analiza funcționarea unui circuit electronic	Evaluare pe parcurs prin test de evaluare scris	50%
	Proiect	-	-	-

10.6 Standard minim de performanță
Cunoașterea conceptelor fundamentale ale programării funcționale, la nivel teoretic și practic (programare în limbajul Haskell). Abilitatea de a dezvolta cod in Haskell. Capacitatea de a respecta termenele limita.

	Semnătura titularului disciplinei:	Semnătura titularului /rilor de aplicații:
	Lector dr. Márton Gyöngyvér	Lector dr. Márton Gyöngyvér
Data:	Semnătura directorului de departament:	Semnătura coordonatorului programului de studii:
16. 09. 2019	Conf. dr.ing. Kátai Zoltán	Ș.l.dr.ing. Szabó László Zsolt