

Universitatea „Lucian Blaga” din Sibiu  
 Facultatea de Științe  
 Catedra de Informatică  
 Domeniul de studii de licență: Informatică  
 Specializarea: Informatică

## PROGRAMA ANALITICĂ

<b>Denumirea disciplinei: Modele formale computaționale</b>
<b>Codul disciplinei: 3906F05O045</b>
<b>Anul de studiu și semestrul în care se studiază disciplina: III/5</b>
<b>Regimul disciplinei (obligatorie O, opțională A sau facultativă L): O</b>
<b>Categoria formativă (fundamentală Fd, de specialitate Sp, generală Gen): Sp</b>
<b>Discipline anterioare cerute *: Limbaje formale și automate, OOP</b>
<b>Forma de evaluare (examen E, verificare V, colocviu C): E</b>
<b>Catedra care coordonează disciplina: Catedra de Informatică</b>
<b>Titularul / titularii disciplinei: Lector drd. Ralf Fabian</b>

\* disciplinele studiate anterior a căror cunoaștere este necesară pentru însușirea disciplinei

<b>Extinderea disciplinei în planul de învățământ *:</b>				
Curs	Seminar	Laborator	Proiect	Total ( $NOAD_{sem}$ )
28		28		56

\* numărul semestrial de ore de activități didactice directe

<b>Bugetul de timp și creditele alocate disciplinei</b>			
$NOAD_{sem}$	$NOSI_{sem}$	$NOT_{sem} = NOAD_{sem} + NOSI_{sem}$	Numărul de credite
56	112	168	6

<b>Obiectivele disciplinei</b>
<p><b>Obiectivele cursului</b></p> <ul style="list-style-type: none"> <li>i) Inițierea studenților în teoria calculabilității;</li> <li>j) Studiarea și aprofundarea diverselor modele formale pentru sisteme de calcul.</li> <li>k) Investigarea computațiilor posibile cu diverse combinații de resurse de calcul.</li> <li>l) Studiarea calculelor posibile cât și a celor imposibile</li> <li>m) Prezentarea de rezultate legate de calcule imposibile pe orice sistem de calcul, indiferent de dimensiune sau putere de calcul.</li> <li>n) Familiarizarea studenților cu gândirea algoritmică;</li> <li>o) Introducerea a mai multor formalizări aparent diferite pentru noțiunea de algoritm;</li> </ul>
<p><b>Obiectivele activităților aplicative (seminar, laborator, proiect)</b></p> <ul style="list-style-type: none"> <li>j) Folosirea metodelor formale în implementări și în a demonstra că exista funcții necalculabile și probleme nedecidabile algoritmic.</li> <li>k) Fixarea deprinderilor de realizare în grup a unui produs program corect (cu limbajele C++,</li> </ul>

C#, Java ), prin parcurgerea tuturor etapelor necesare și reflectarea lor într-o documentație completă.

l) Aprofundarea cunoștințelor de programare ale studenților.

**Conținutul disciplinei** (capitolele cursului / tematica seminarului / lucrărilor practice / etapele proiectului)

**CURS**

Nr. crt.	Tema	Nr.ore	Săptămâna
1.	Scurt istoric. Noțiuni introductive și notații Elemente matematice preliminare. Mulțimi, funcții, relații, grafuri, arbori. Tehnici de demonstrare.	2	1
2.	Modele generale de calculabilitate. Clase abstracte de algoritmi.	2	2
3.	Paradigma stărilor finite. Expresii regulate și automate finite. Proprietăți ale limbajelor regulate.	4	3,4
4.	Gramatici independente de context și automatul push-down. Proprietăți ale limbajelor independente de context	2	5
5.	Proprietăți ale limbajelor dependente de context. Automatul liniar mărginit	2	6
6.	Funcții recursive	2	7
7.	Algoritmi Markov	2	8
8.	Mașina Turing și calculabilitatea. Mașina Turing și funcții recursive. Mașina Turing universală. Problema de oprire	2	9
9.	Problema decidabilității pentru gramatici și alte modele.	4	10, 11
10.	Extensii ale modelelor clasice. Logică Fuzzy. Gramatici și automate Fuzzy. Observații și concluzii	6	12, 13, 14

**LABORATOR**

Nr. crt.	Tema	Nr.ore	Săptămâna
1.	Recapitulare conceptelor și noțiunilor de baza pentru mecanismele care vor fi studiate. Familiarizarea cu mediul de dezvoltare. Stabilirea echipelor pentru temele de implementare. Împărțirea proiectelor.	2	1
2.	Expresii regulate și Automatul finit. Gramatici independente de context.	4	2,3
3.	Mașina Turing ca mecanism de specificare a mulțimilor de elemente. Modalități de implementare	2	4
4.	Mașina Turing ca mecanism de calcul.	2	5
5.	Codificarea mașinii Turing pe banda de intrare	2	6
6.	Operații cu mașini Turing. Compunerea. Mașini Turing cu n benzi	2	7
7.	Funcții recursive. Implementarea funcțiilor și operatorilor de compunere	2	8
8.	Implementarea unui framework pentru executarea algoritmilor	2	9

	Markov		
9.	Dezvoltarea operațiilor pentru compunerea algoritmilor Markov	2	10
10.	Exemple de aplicații cu logica Fuzzy. Modele formale cu logică Fuzzy	6	11, 12
11.	Susținerea și predarea proiectului	4	13, 14

#### **Descrierea metodelor de predare**

- Pentru predare se va folosi prelegerea, dezbaterea, învățarea prin cooperare, explicația, tematizarea.
- lucrări de laborator disponibile studenților, la cerere pe suport magnetic.
- Sunt valabile regulamentele oficiale ale universității în legătură cu prezenta studenților la activitățile didactice și cu cazurile de copiat și plagiat.
- Promovarea examenului este condiționată de predarea completă a lucrărilor de proiect.
- Prezenta la orele de proiect este obligatorie.

#### **Descrierea formelor și metodelor de evaluare a cunoștințelor**

Nota finală se va stabili după cum urmează:

- |         |                      |      |
|---------|----------------------|------|
| xxxii)  | Proiecte de semestru | 60 % |
| xxxiii) | Examen final         | 40 % |

Evaluarea proiectului de semestru constă în:

- predarea și susținerea în ultima săptămâna din semestru a programului și documentația realizată, pe care se va acorda o notă. Nu se accepta întârzieri;
- se va pune accent deosebit pe scrierea cât mai clară a documentației complete și la timp.

#### **Bibliografie obligatorie**

11. Arthur Fleck - *Formal Models of Computation, The ultimate limits of computing*, World Scientific Press, 2001, ISBN 978-981-02-4500-9
12. Ralf Fabian, *Limbaje formale Teorie. Exemple. Probleme*, Editura Universității „Lucian Blaga”, Sibiu, 2006, pag. 123, ISBN 973-739-225-6
13. Calude Cristian, *Complexitatea calculului*, North-Holland, 1988.
14. Hopcroft, J.E., Motwani, R. and Ullman, J.D. - *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley 2007.
15. Creanga I., Reischer C., Simovici D. *Introducerea algebrică în informatică*, vol. 1 și vol. 2 , 1974

#### **Bibliografie opțională**

1. Emil M. Popa, *Formal Syntax and Semantics of Programming Language*, Editura „Alma Mater”, Sibiu, 2004.
2. Aho, A.V., Ullman, J.D., *The Theory of Parsing Translations and Compiling*, vol.1: Parsing, vol.2: Compiling, New Jersey, Prentice-Hall, inc., 1972.
3. Teodor Rus, *Mecanisme formale pentru specificarea limbajelor*, Ed. Academie Române, București, 1983.
4. Ernest G. Manes, Michael A. Arbib. *Algebraic Approaches to program semantics* – Springer – Verlag New York Berlin Heidelberg London Paris Tokyo – 1986
5. Salomaa A., *Formal Languages*, New York, Academic Press, 1973.
6. Păun Gh., *Mecanisme generative ale proceselor economice*, Ed Tehnică, București, 1988.
7. Păun Gh., *Probleme actuale in teoria limbajelor formale*, Editura Academiei, 1983.

8. Păun Gh., *Gramatici contextuale*, București, 1982

**Data elaborării:**

**Titularul / titularii disciplinei,  
Lector drd. Ralf Fabian**