

FIȘA DISCIPLINEI

Metode inteligente și aplicațiile lor în ingineria software

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Matematică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Matematică – Informatică (în limba engleză)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Metode inteligente și aplicațiile lor în ingineria software				Codul disciplinei		MLE7040			
2.2. Titularul activităților de curs			Lect. Dr. Oneț-Marian Zsuzsanna								
2.3. Titularul activităților de seminar			Lect. Dr. Oneț-Marian Zsuzsanna								
2.4. Anul de studiu		3	2.5. Semestrul		6	2.6. Tipul de evaluare		E	2.7. Regimul disciplinei		Opțional

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	1S + 1L
3.4. Total ore din planul de învățământ	48	din care: 3.5. curs	24	3.6 seminar/laborator/proiect	24
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					22
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					30
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					30
Tutoriat (consiliere profesională)					15
Examinări					5
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				102	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Algoritmi si Programare, Bazele programarii orientate obiect, Metode avansate de programarea aplicatiilor software
4.2. de competențe	Competențe bune de programare în Python

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	

6. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/esențiale	elaborarea și analiza unor algoritmi pentru rezolvarea problemelor
Competențe transversale	- aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul didactico-științific, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională - desfășurarea eficientă și eficace a activităților organizate în echipă - utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată, atât în limba română, cât și într-o limbă de circulație internațională

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Scopul acestui curs este de a familiariza studenții cu metode inteligente de căutare și algoritmi de clustering, precum și cu modul în care acestea pot fi utilizate pentru a rezolva diverse probleme din domeniul ingineriei software.
7.2 Obiectivele specifice	- Să înțeleagă și să recunoască componentele unei probleme de căutare/optimizare. - Să învețe despre diferite tipuri de testare software și să înțeleagă cum pot fi descrise activitățile de testare software ca probleme de căutare. - Să recunoască unele dintre cele mai cunoscute bad smells din codul sursă. - Să înțeleagă algoritmi de clustering și modul în care aceștia pot fi utilizați cu date din ingineria software

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Organizarea cursului. Inginerie software bazată pe căutare. Componentele unei probleme de optimizare.	- Expunere interactivă - Explicație - Conversație - Demonstrație didactică - Studiu de caz	
2. Hill climbing, Simulated Annealing, Tabu Search		
3. Algoritmi genetici		
4. Aplicații ale algoritmilor de căutare pentru mutation testing.		
5. Aplicații ale algoritmilor de căutare pentru unit testing		
6. Aplicații ale algoritmilor de căutare pentru fuzz testing		
7. Aplicații ale algoritmilor de căutare pentru testare de integrare		
8. Aplicații ale algoritmilor de căutare pentru testarea de regresie		
9. Algoritmi de clustering și aplicațiile lor în testarea software		

10. Bad smells și refactorizare		
11. Metode inteligente pentru refactorizare software		
12. Abordări bazate pe embedding-ul de cod		
Bibliografie 1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001 2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Computing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012 3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989 4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018		
8.2 Seminar / laborator	Metode de predare	Observații
1. Organizarea seminarului.	• Conversație • Dialog • Studiu de caz	• Seminarul va fi organizat o dată la 2 săptămâni, cu o durată de 2 ore.
2. Alegerea temei pentru raport I		
3. Prezentare raport I		
4. Prezentare raport I + alegerea temei pentru raport II		
5. Prezentare raport II		
6. Prezentare raport II		
Bibliografie 1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001 2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Computing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012 3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989 4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018		
8.3 Laborator	Metode de predare	Observații
1. Biblioteci python și implementarea algoritmului Hill climbing	• Conversation • Dialog • Case studies	Laboratorul va fi organizat o dată la 2 săptămâni, cu o durată de 2 ore. În cadrul fiecărui laborator, studenții vor lucra în echipe de 2-3 persoane și vor trebui să rezolve probleme simple în Python, legate de temele discutate la curs.
2. Mutation testing în Python		
3. Unit / fuzz testing în Python		
4. Algoritmi de clustering		
5. Refactorizare software		
6. Metode inteligente pentru refactorizare software		
Bibliografie 1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001 2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Computing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012 3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989 4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018		

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

Conținutul disciplinei este în concordanță cu discipline similare din alte universități din România și din străinătate, precum și cu cerințele pe care le-ar putea avea potențialii angajatori din domeniul ingineriei software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe de bază despre algoritmi de căutare și clustering și despre aplicațiile acestora în ingineria software	Examen scris în sesiune	40%
10.5 Seminar	Capacitatea de a recunoaște și de a înțelege conceptele discutate într-un articol de cercetare. Capacitatea de a înțelege un algoritm nou de căutare	Prezentarea unui raport despre un algoritm de căutare și despre conținutul unui articol de cercetare.	30%
10.6 Laborator	Corectitudinea și completitudinea proiectelor de laborator realizare în echipe de 2-3 persoane.	Observarea continuă pe parcursul laboratoarelor. Media notelor obținute la cele 6 proiecte de laborator.	30%
10.7 Standard minim de performanță			
- Fiecare student trebuie să demonstreze că a dobândit un nivel acceptabil de cunoștințe și înțelegere a domeniului și că este capabil să exprime coerent aceste cunoștințe. - Nota la examenul scris trebuie să fie mai mare de 5, iar media finală trebuie să fie, de asemenea, mai mare de 5. - Participarea la cel puțin 4 laboratoare și 3 seminare este obligatorie, în caz contrar studentul nu va avea dreptul să participe la examenul scris. - Cel puțin o temă de laborator (din cele șase) trebuie rezolvat pentru a putea participa la examenul scris.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru Dezvoltare durabilă - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".

Data completării:
15.04.2025

Semnătura titularului de curs
Lect. dr. Zsuzsanna ONETȚ-MARIAN

Semnătura titularului de seminar
Lect. dr. Zsuzsanna ONETȚ-MARIAN

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA