

## ANEXA Nr. 4

## INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

<b>Denumire câmp</b>	<b>Descriere</b>
<b>Facultatea</b>	<b>Matematică și Informatică</b>
<b>Departamentul</b>	Informatică
<b>Poziția în statul de funcții</b>	102
<b>Funcția</b>	<b>Asistent universitar</b>
<b>Disciplinele din încărcătura postului</b>	Bazele programarii orientate obiect (în lb. română) Inteligența artificială (în lb. română) Metode avansate de programare (în lb. engleză) Programare logică și funcțională (în lb. engleză)
<b>Domeniul științific</b>	<b>Informatică</b>
<b>Descrierea postului scos la concurs</b>	Asistent, 102, Departamentul de Informatică. Postul de asistent universitar presupune desfășurarea de activități didactice, de cercetare științifică și de îndrumare a studenților, precum și efectuarea de servicii pentru comunitatea academică.  Candidații la ocuparea postului vacant de asistent universitar trebuie să aibă palmaresul științific în concordanță cu standardele domeniului Informatică și cu disciplinele postului. De asemenea, candidații trebuie să facă dovada stăpânirii limbii engleze prin documente depuse la dosar (nivel C1 sau documente atestând studii sau stagii de cercetare cumulate de cel puțin 9 luni în străinătate, în instituții de învățământ sau cercetare unde comunicarea s-a făcut în limba engleză).
<b>Atribuții</b>	Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post.  Activitatea de cercetare științifică: <ul style="list-style-type: none"> <li>• participarea la cel puțin un seminar de cercetare în cadrul facultății;</li> <li>• participarea la competiții pentru obținerea de granturi de cercetare științifică;</li> <li>• publicarea, în fiecare perioadă de 3 ani, a cel puțin 3 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP);</li> <li>• îndeplinirea cerințelor din anexa la fișa postului, cel puțin la nivelul calificativului -Satisfăcător-.</li> </ul> Activitatea de îndrumare a studenților: îndrumare de lucrări de diplomă, tutore la o formație de studenți, îndrumarea acestora pentru participarea la activitatea cercurilor științifice și la concursuri studențești.  Servicii pentru comunitatea academică: participare la acțiunile desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).
<b>Data și ora susținerii</b>	13.07.2026 ora 10.00

<b>prelegerii/ probei orale</b>	
<b>Locul susținerii prelegerii/ probei orale (adresa Facultății/ Institutului și sala)</b>	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L439
<b>Descrierea probei orale</b>	<p>Proba orală constă în prezentarea unui proiect de seminar/ laborator/ lucrări practice.</p> <p>Comisia stabilește, pe baza tematicii și bibliografiei de concurs, tema prezentării probei orale și o comunică candidaților cu 48 de ore înaintea susținerii probei prin e-mail și prin afișarea pe pagina web a facultății, cu menționarea datei și orei afișării, sub semnătura președintelui comisiei de concurs.</p> <p>Durata minimă a probei orale susținute de către candidat este de 30 de minute; proba conține în mod obligatoriu și o sesiune de întrebări din partea comisiei și/ sau a publicului.</p> <p>Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.</p>
<b>Data și ora evaluării dosarului candidatului</b>	N/A
<b>Locul evaluării dosarului candidatului (adresa Facultății/ Institutului și sala)</b>	N/A
<b>Descrierea evaluării dosarului candidatului</b>	N/A
<b>Data și ora susținerii probei scrise</b>	13.07.2026 ora 9.00
<b>Locul susținerii probei scrise (adresa Facultății/ Institutului și sala)</b>	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L439
<b>Descrierea probei scrise</b>	Evaluarea cunoștințelor fundamentale în domeniul Informatică prin examen scris din tematica anunțată.
<b>Tematica și bibliografia probelor de concurs</b>	<p><b>Proba 1 – Probă scrisă</b></p> <p><b>Tematică:</b></p> <p><b>A. Fundamentele programării</b>  Subalgoritmi: specificare, testare.  Clase de algoritmi: căutare, sortare, interclasare.  Metode de proiectare a algoritmilor: top-down, rafinare succesivă.  Subprograme, apel și modalități de transmitere a parametrilor (prin valoare și referință).  Tehnici de programare: Backtracking, Divide et impera, Greedy.  Programare modulară: modul, interfață, implementare; concretizare în C/C++, Java, Python.</p> <p><b>B. Programare orientată pe obiecte</b></p>

Clase, obiecte.  
Moștenire, polimorfism.  
Programare bazată pe interfețe.

### **C. Structuri de date**

Tipuri abstracte de date (TAD).  
TAD Mulțime, Colecție, Listă, Stivă, Coadă, Dicționar.  
Specificarea TAD.  
Implementări pentru TAD folosind: vectori, liste înlănțuite, arbori binari.

#### *Bibliografie:*

1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994
2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006
3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000
4. B. Eckel, Thinking in C++, vol I și II, <http://www.mindview.net>
5. B. Eckel, Thinking in Java, <http://www.mindview.net>
6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994
7. The Python language reference.
8. <http://docs.python.org/py3k/reference/index.html>
9. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005

### **Proba 2 –Probă orală: susținerea unui proiect de seminar/ laborator/ lucrări practice**

#### **A. Bazele programării orientate obiect (în lb. română)**

1. Elemente de baza ale limbajului C&C++: Elemente lexicale. Operatori. Conversii ; Tipuri de date. Variabile. Constante ; Domeniu de vizibilitate și durata de viață a variabilelor ; Spații de nume ; Instrucțiuni ; Declarația și definiția funcțiilor ; Supraincercarea funcțiilor. Funcții inline.
- 2.a. Tipuri de date derivate și utilizator și alocare dinamică în C++: Tipurile tablou și structură ; Tipurile pointer și referință ; Alocarea și dealocarea memoriei ; Pointeri la funcții și pointeri void.
- 2.b. Programare modulară în C++: Fișiere header. Biblioteci ; Implementări modulare de tipuri abstracte de date ; Folosirea tipului pointer void pentru obținerea genericității.
3. Metoda programării orientate-obiect în C++ : Clase și obiecte ; Membrii unei clase. Specificatori de acces ; Constructori / destructori ; Diagrame UML pentru clase (membrii, acces).
4. Supraincercarea operatorilor ; Membrii statici ; Funcții Friend.
5. Clase: containere și iteratori : Lista simplu înlănțuită și lista dublu

- inlantuita ; Stiva, Coadă, Tabele de dispersie ; Relatia de asociere/agregare între clase - reprezentare UML
6. Relatia de mostenire: Mostenire simpla. Clase derivate ; Principiul substitutiei ; Suprascrierea metodelor ; Mostenire multipla ; Relatia de specializare/generalizare între clase - reprezentare UML.
7. Clase abstracte si Polimorfism: Metode virtuale ; Legare dinamica ; Mostenire virtuala ; Reutilizare cod (mostenire/compozitie) ; Conversii (upcast/downcast).
8. Proiectare orientata-obiect si proiectare bazata pe interfete: Clase abstracte, interfete ; Reprezentare UML pentru interfete ; Proiectarea orientata-obiect a unei biblioteci de structuri de date
9. Sabloane (Template): Functii template. Clase template ; Reutilizarea codului sursa.
10. Operatii de intrare/iesire: Fluxuri de intrare/ iesire. Ierarhii de clase pentru I/O ; Formatare. Manipulatori ; Lucrul cu fisiere.
11. Tratarea exceptiilor: Notiunea de exceptie in programare ; Tratarea exceptiilor in C++.
12. Biblioteca STL: Clase container si iterator ; Folosirea algoritmilor din STL.
13. Elemente de programare bazata pe evenimente si Interfete grafice: MCF, VCL, QT
14. Sabloane de proiectare creat ionale, structurale, comportamentale.

*Bibliografie:*

1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley Publ., Massachusetts, 1983.
2. R. Andonie, I. Garbacea, Algoritmi fundamentali. O perspectiva C++, Editura Libris,
3. Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicabile, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994.
5. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995.
6. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996.
7. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996.
8. Dan Roman, Ingineria programarii obiectuale, Editura Albastra, Cluj\_Napoca, 1996.
9. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998.
10. Bruce Eckel, Thinking in C++, [www.bruceeckel.com](http://www.bruceeckel.com)

**B. Inteligență artificială (în lb. română)**

1. Introducere în IA
2. Rezolvarea problemelor prin căutare: Tipuri de probleme, Strategii de

căutare (SC)

a. Neinformate (nI)

b. Informate: Best-first search, Greedy best-first search, A\*, IDA\*

c. Căutare locală: Căutare locală simplă (Căutare tabu, Hill climbing - Simulated annealing), Căutare locală în fascicol (beam local search), Algoritmi evolutivi), Optimizare bazată pe comportamentul de grup (Particle swarm optimization), Optimizare bazată pe furnici (Ant colony optimisation)

d. Căutare adversială: Arbori AND-OR, Minimax - Taieturi alpha-beta

3. Sisteme inteligente

a. Sisteme bazate pe cunoștințe (SBC): Reprezentarea cunoștințelor certe și incerte, Inferența pe baza cunoștințelor, Sisteme bazate pe reguli (Sisteme care manipulează elemente de certitudine și Sisteme care manipulează elemente de incertitudine)

b. Sisteme care învață singure (inteligentă computațională):

- Tipuri de probleme de învățare (supervizată și nesupervizată) - Metrici de evaluare a procesului de învățare; Algoritmi de învățare (Algoritmi de învățare supervizată - Metoda celor mai mici pătrate; Gradient descrescător; Regresie logistică; Rețele neuronale artificiale (clasice); Rețele neuronale artificiale (convolutive); Cel mai apropiat vecin; Arbori de decizie; Mașini cu suport vectorial)

- Algoritmi de învățare nesupervizată aglomerativă și divizivă: K-means și K-medoids

- Algoritmi de învățare prin întărire: Q-learning și Rețele neuroanle de tip deep

4. Sisteme hibride

*Bibliografie:*

1. S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995

2. C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011

3. M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998 4. A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001

5. T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997

6. James Kennedy, Russel Eberhart, Particle Swarm Optimisation, Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, 1995

7. Marco Dorigo, Christian Blum, Ant colony optimization theory: A survey, Theoretical Computer Science 344 (2005) 243 – 27

8. H.F. Pop, G. Șerban, Inteligență artificială, Cluj Napoca, 2004

9. D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003

10. C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006 11. I.

Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016

<https://www.deeplearningbook.org/>

### **C. Metode avansate de programare (în lb. engleză)**

1. Introducere în platforma Java: platformă, sintaxa limbajului, tipuri de date primitive, matrice, clase, interfețe, pachete, enumerări, suprascriere, supraîncărcare, excepții
2. Colecții și tipuri generice: clase anonime, polimorfism, conversie
3. IO, NIO: fluxuri binare și orientate pe caractere, fișiere, canale și tamponare
4. Programare funcțională: expresii lambda, fluxuri
5. GUI: componente Java FX, gestionarea evenimentelor
6. Concurență: thread-uri, executori, futures, gestionarea excepțiilor
7. Concurență: metode sincronizate vs. asincrone, metode de apel invers, anulare
8. GUI (continuare):FXML, CSS. Metaprogramare: reflecție, serializare
9. Introducere în C# și .Net
10. Colecții în C#
11. Operații IO în C#
12. GUI în C#
13. Introducere în Scala
14. Introducere în Rust

#### *Bibliografie:*

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Specificația limbajului Java™ Java SE 7 Edition.
2. Eckel, B., Thinking in Java, ediția a 4-a, Prentice Hall, 2006
3. Eckel, B.: Gândirea în modele cu Java, 2004. MindView, Inc
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994
5. \*\*\*, Tutorial Java, 2013. <http://download.oracle.com/javase/tutorial/>
6. Joseph Albahari și Ben Albahari, C# 4.0 în câteva cuvinte, ediția a patra, O'Reilly, 2022
7. \*\*\*, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>

### **D. Programare logica si functionala (în lb. engleză)**

1. Recursivitate în Pseudocod
2. Liste în Prolog
3. Arbori în Prolog. Gestiunea listelor (eterogene) în Prolog.
4. Backtracking în Prolog
5. Prelucrarea listelor în LISP
5. Programare recursivă în Lisp
6. Funcții MAP în LISP.

#### *Bibliografie:*

1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012

	<p>2. POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, ClujNapoca, 2003</p> <p>3. <a href="http://www.ifcomputer.com/PrologCourse">http://www.ifcomputer.com/PrologCourse</a>, Lecture on Prolog</p> <p>4. <a href="http://www.lpa.co.uk">http://www.lpa.co.uk</a>, Logic Programming</p> <p>5. FIELD A., Functional Programming, Addison Wesley, New York, 1988.</p> <p>6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.</p>
<p><b>Descrierea procedurii de concurs</b></p>	<p>Comisia de concurs evaluează candidații ținând sub următoarele aspecte:</p> <ul style="list-style-type: none"> <li>• Conținutul dosarului individual;</li> <li>• Proba orală (proba 1)</li> <li>• Proba scrisă (proba 2).</li> </ul> <p>În evaluarea activității științifice se va ține cont de calitatea publicațiilor și contribuțiile candidaților în raport cu exigențele prevăzute în norma didactică.</p> <p>În cazul tuturor posturilor, durata minimă a probei orale/ prelegerii susținute de către candidat este de 30 de minute; proba conține în mod obligatoriu și o sesiune de întrebări din partea comisiei.</p> <p>În cazul unui post cu încărcătură integrală într-o singură limbă străină, probele de concurs vor fi susținute în limba respectivă în fața comisiei de concurs; în cazul unui post cu încărcătură în mai multe limbi, probele de concurs vor fi susținute în limbile respective, conform unei proceduri stabilite prin anunțul de concurs.</p> <p>Nota finală a fiecărui candidat se calculează ca medie aritmetică a notelor obținute la criteriile de mai sus. Fiecare membru al comisiei (inclusiv președintele) întocmește un referat individual de apreciere care propune o notă finală pentru fiecare candidat.</p> <p>Candidații eligibili pentru ocuparea postului scos la concurs trebuie să obțină:</p> <ul style="list-style-type: none"> <li>• cel puțin nota 7 (șapte) la fiecare probă;</li> <li>• media generală cel puțin 8,50 (opt și 50%).</li> </ul> <p>Președintele comisiei de concurs întocmește un raport asupra concursului în care prezintă notele finale atribuite candidaților de către membrii comisiei și indică media generală obținută de fiecare candidat, calculată ca medie aritmetică a notelor finale din referatele individuale. Media generală astfel obținută reprezintă rezultatul concursului pentru fiecare candidat. Pe baza mediei generale, comisia de concurs decide ierarhia candidaților și nominalizează candidatul eligibil care a întrunit cel mai bun rezultat în concurs. Președintele comisiei de concurs supune raportul de sinteză asupra concursului votului deschis al membrilor comisiei. În urma exercitării votului, președintele constată rezultatul votului, îl comunică membrilor comisiei și îl menționează în încheierea raportului asupra concursului, cu precizarea numărului de voturi "pentru", respectiv "contra". În cazul în care votul "pentru" nu este acordat de majoritatea membrilor comisiei, postul scos la concurs nu este ocupat de niciun candidat. Ierarhia candidaților stabilită prin media generală nu poate fi modificată prin votul comisiei. Raportul asupra concursului este semnat de fiecare dintre membrii comisiei de concurs și de către președintele comisiei.</p>

<b>Perioada de comunicare a rezultatelor</b>	<i>În aceeași zi cu încheierea lucrărilor comisiei.</i>
<b>Perioada de depunere a contestațiilor</b>	<i>În termen de 3 zile lucrătoare de la comunicarea rezultatului.</i>
<b>Salariul minim de încadrare a postului la momentul angajării</b>	<i>7417 lei</i>

**Director Departament,  
Conf. dr. Adrian STERCA**

