

## ANEXA Nr. 4

## INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

Denumire câmp	Descriere
Facultatea	Matematică și Informatică
Departamentul	Informatică
Poziția în statul de funcții	94
Funcția	Asistent universitar
Disciplinele din încercătura postului/ ariile de cercetare, așa cum figurează în statul de funcții	Metode avansate de programare Verificarea și validarea sistemelor soft
Domeniul științific	Informatică
Descrierea postului scos la concurs	<p>Asistent, 94, 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.</p> <p>Candidații la ocuparea postului vacant de asistent universitar trebuie să aibă palmaresul științific în concordanță cu standardele domeniului Informatică și cu disciplinele postului.</p>
Atribuții	<p>Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post.</p> <p>Activitatea de cercetare științifică:</p> <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 4 ani, a cel puțin 5 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP) din care cel puțin 2 indexate ISI sau SCOPUS sau în volumele unor conferințe internaționale relevante (ACM, IEEE, AMS, EMS).</li> </ul> <p>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.</p> <p>Servicii pentru comunitatea academică: participare la acțiunile desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).</p>
Data și ora susținerii probei orale	28.01.2020, ora 12:00
Locul susținerii probei orale	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C512

<p><b>Probele de concurs, data, ora și locul de susținere a acestora</b></p>	<p>Pentru postul de ASISTENT UNIVERSITAR, concursul constă în:</p> <ol style="list-style-type: none"> <li>1. evaluarea dosarului individual;</li> <li>2. susținerea unei probe orale;</li> <li>3. susținerea unei probe scrise.</li> </ol> <p>Proba orală constă în prezentarea unui proiect de seminar/ laborator/ lucrări practice. 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 afișarea la avizierul și pagina web a facultății, cu menționarea datei și orei afișării, sub semnătura președintelui comisiei de concurs. 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>Proba 1 – Probă scrisă: 28.01.2020, ora 8:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C510.</p> <p>Proba 2 – Probă orală: susținerea unui proiect de seminar/ laborator/ lucrări practice – 28.01.2020, ora 12:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C512. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.</p> <p><b>Dosarul individual, proba orală și proba scrisă contează în proporții egale la nota finală</b> acordată în referatul individual de apreciere întocmit de fiecare membru al comisiei de concurs;</p>
<p><b>Tematica și bibliografia probelor de concurs</b></p>	<p>Proba 1 – Probă scrisă</p> <p>Tematică:</p> <p>A. Metode avansate de programare</p> <ol style="list-style-type: none"> <li>1. Introducere în platforma Java: platformă, sintaxa limbajului, tipuri de date primitive, tablouri, clase, interfețe, pachete, tipul enum, suprascrierea, supraîncărcarea, excepții</li> <li>2. Colecții și tipuri generice: clase anonime, polimorfism, casting</li> <li>3. Intrări - Ieșiri (IO,NIO): stream (flux) pe octeți și pe caractere, files, canal de comunicare(channel), zonă tampon (buffer)</li> <li>4. Java 8: expresii lambda, fluxuri (streams)</li> <li>5. GUI: Componentele Java FX, tratarea evenimentelor (curs introductiv)</li> <li>6. GUI (cont) –FXML, CSS, MVC pattern</li> <li>7. Concurență: fire de execuție, sincronizare, taskuri asincrone - Java</li> <li>8. Metaprogramming: reflecție, serializare -Java</li> <li>9. XML: schema, documents - Java</li> <li>10. Introducere în platforma .Net. Limbajul C#. Colecții în C#.</li> <li>11. LINQ</li> </ol>

12. I/O. Concurență: fire de execuție, sincronizare, taskuri asincrone. (C#)
13. GUI – C#.

#### Bibliografie

1. JAMES GOSLING, BILL JOY, GUY STEELE, GILAD BRACHA, ALEX BUCKLEY. The Java™ Language Specification Java SE 7 Edition.
2. ECKEL, B., Thinking in Java, 4th edition, Prentice Hall, 2006
3. ECKEL, B., Thinking in Patterns with 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. \*\*\*, The Java Tutorial, 2013  
<http://download.oracle.com/javase/tutorial/>

#### B. Fundamentele programării, Programare orientată pe obiecte, Structuri de date

##### 1. Fundamentele programării

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.

##### 2. Programare orientată pe obiecte

Clase, obiecte.

Moștenire, polimorfism.

Programare bazată pe interfețe.

##### 3. 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. <http://docs.python.org/py3k/reference/index.html>
8. 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

Tematică: Verificarea și validarea sistemelor soft

1. Verificarea și validarea sistemelor soft. Inspectarea programelor
2. Testarea programelor (1): Conceptul de testare a programelor. Criterii de testare. Testarea black-box.
3. Testarea programelor (2): Criterii de testare. Testarea white-box.
4. Niveluri de testare. Tipuri de testare
5. Testarea aplicațiilor Web. Selenium Web Driver
6. Execuție simbolică
7. Verificarea modelelor
8. Teoria corectitudinii programelor (I): Evoluția conceptului de corectitudine. Contribuțiile lui Floyd, Hoare
9. Teoria corectitudinii programelor (II): Contribuțiile lui Dijkstra
10. Asigurarea calității produselor soft. Controlul calității.
11. Abilități în testare și atribuții ale testerului

Bibliografie

1. FRENTIU, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. PRESSMAN, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
3. L. CRISPIN, J. GRECORY, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. PEZZAND, M. YOUNG, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. NAIK, P. TRIPATHY, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. KATOEN, Principles of Model Checking, MIT Press, May 2008
7. R. PATTON, Software Testing, Sams Publishing, 2005
8. GLENFORD J. MYERS, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. BERNSTEIN, Practical software testing, Springer, 2002
10. BALANESCU T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995.
11. MORGAN, C., Programing from Specifications, Prentice Hall, NewYork, 1990.
12. Dromey G., Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989.
13. HOARE, C.A.R., An axiomatic basis for computer

	<p>programming, CACM, 12(1969), pg.576-580, 583.</p> <p>14. DIJKSTRA, E., A constructive approach to the problem of program correctness, BIT, 8(1968), pg.174-186.</p> <p>15. DIJKSTRA, E., Guarded commands, nondeterminacy and formal derivation of programs, CACM, 18(1975), 8, pg.453-457.</p>
<p><b>Descrierea procedurii de concurs</b></p>	<p>Comisia de concurs evaluează candidații ținând cont de următoarele criterii:</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>Nota finală a fiecărui candidat se calculează ca medie aritmetică a notelor obținute la criteriile de mai sus.</p> <p>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 6 (șase) la fiecare criteriu;</li> <li>• nota finală cel puțin 7 (șapte) dată de fiecare referent;</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 asupra concursului votului secret al membrilor comisiei. În urma exercitării votului secret, președintele constată rezultatul votului, îl comunică membrilor comisiei și îl menționează în încheierea raportului asupra concursului, cu precizarea numărului de voturi "pentru", respectiv "contra", votul fiind menținut secret. În cazul în care votul "pentru" nu este acordat de majoritatea membrilor comisiei, postul scos la concurs nu este ocupat de niciun candidat. Raportul asupra concursului este semnat de fiecare dintre membrii comisiei de concurs și de către președintele comisiei.</p>

**Director department,**  
**Prof. dr. Anca Andreica**