ANEXA Nr. 4

INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

	Română	Engleză
Universitatea	Universitatea Babeş-Bolyai din Cluj-Napoca	Babes-Bolyai University, Cluj-Napoca
Facultatea / Structura organizațională de conducere	Facultatea de Matematică și Informatică	Faculty of Mathematics and Computer Science
Departamentul / Unitatea organizațională	Departamentul de Informatică	Department of Computer Science
Poziția în statul de funcții	62	62
Funcția	Lector universitar dr.	Lecturer
Disciplinele din încărcătura postului/ ariile de cercetare, așa cum figurează în statul de funcții	Sisteme de operare (în lb. engleză); Programare logică și funcțională (în lb. engleză); Medii de proiectare și programare (în lb. engleză); Programare pentru dispozitive mobile (în lb. engleză)	Operating systems; Functional and Logical Programming; Systems for Design and Implementation; Mobile Application Programming
Domeniul științific	Informatică	Computer Science
Descrierea postului scos la concurs	Lector, 62, Departamentul Informatică. Postul de lector universitar presupune desfăşurarea de activităţi didactice, de cercetare ştiinţifică şi de îndrumare a studenţilor, precum şi efectuarea de servicii pentru comunitatea academică. Cerințe pentru candidați: Candidații la ocuparea postului vacant de lector universitar trebuie să aibă palmaresul științific în concordanță cu standardele domeniului Informatică și cu disciplinele postului. Este obligatorie deținerea diplomei de doctor în domeniul postului pentru ocuparea căruia se candidează. 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).	Lecturer, 62, Department of Computer Science. The position of lecturer involves teaching activities, scientific research and guidance of the students, as well as activities for the academic community. Requirements for candidates: The candidates must have the scientific record in conformity with the standards of computer science domain and with the required disciplines. It is mandatory for the candidate to have a Phd degree in the field of the position he/she is a candidate for. Also, candidates must prove knowledge of the English language through documents (level C1 or documents attesting to study or research internships for at least 9 months abroad in institutions where communication was made in English).

Atribuții	Activitatea didactică: curs, seminar, laborator, proiecte, consultaţii, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post. Activitatea de cercetare ştiinţifică: •participarea la cel puţin un seminar de cercetare în cadrul facultăţii; •participarea la competiţii pentru obţinerea de granturi de cercetare ştiinţifică; • publicarea, în fiecare perioadă de 3 ani, a cel puţin 3 articole/studii indexate BDI (Mathematical Reviews/MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP); • îndeplinirea cerinţelor din anexa la fisa postului, cel puțin la nivelul calificativului -Satisfăcător-. Activitatea de îndrumare a studenţilor: îndrumare de lucrări de diplomă şi disertaţie, tutore la o formaţie de studenţi, îndrumarea acestora pentru participarea la activitatea cercurilor ştiinţifice şi la concursuri studenţeşti. Servicii pentru comunitatea academică: participare la acţiunile desfăşurate de departament, facultate şi universitate (promovarea admiterii, colaborarea cu mediul economic, etc).	Teaching activity: course, seminar, laboratory, projects, exams, preparation of teaching materials. The scientific research activity: • at least one participation at a research seminar within the faculty; • participation in competitions in order to obtain grants for scientific research; • the publication, every 3 years, of at least 3 papers/studies BDI indexed (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP); • fulfillment of the requirements in the annex to the job description, at least at qualification level -Satisfactory-. The activity of mentoring students: supervision of diploma and dissertation theses, tutoring activities, guidance for students participation in scientific seminars and student contests. Activities for the academic community: participation in the activities performed by the department, faculty and university (promotion of admission, the collaboration with the economic environment, etc).
Perioada de înscriere la concurs	10.04.2025-28.05.2025	10.04.2025-28.05.2025
Data și ora susținerii probei orale	20.06.2025, 9:00	20.06.2025, 9:00 - (EEST GMT+3 - dd.mm.yyyy, hh:mm)
Locul susținerii probei orale (adresa Facultății/ Institutului și sala)	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L338	Department of Computer Science, Teodor Mihali Str, Room L338
Descrierea probei orale	Proba orală constă în prezentarea unui curs, în funcție de natura postului. Comisia de concurs stabilește titlul și îl anunță candidatului/candidaților cu 48 de ore înaintea	The oral test consists of the presentation of a course, depending on the nature of the position. The competition committee determines the title and announces it to the

	prelegerii prin email si prin afisarea pe pagina web a facultatii. Prelegerea este urmată obligatoriu de o sesiune de întrebări din partea comisiei şi/sau a publicului. Durata minimă a probei orale susținute de către candidat este de 30 de minute; proba conține în mod obligatoriu şi o sesiune de întrebări din partea comisiei şi/ sau a publicului. Dacă sunt mai mulţi candidaţi, comisia va decide ordinea în care aceştia vor susţine proba orală.	candidate/candidates 48 hours before the lecture through email and by displaying it on the faculty web site. The lecture is followed by a question session from the committee and/or the audience. The public lecture must have at least 30 minutes, being always followed by a session of questions of the committee members and/or the public. If there are more candidates, the committee will decide the order in which they will present the lecture.
Tematica şi bibliografia probelor de concurs	Proba 1 –Probă orală: susținerea unui curs Tematică: A. Sisteme de operare (în lb. engleză); 1. Unix: introducere. Comenzi Unix şi argumente. Expresii regulare, specificarea fişierelor, specificări generice. Filtre şi editoare de texte. 2. Programare Shell. Procesorul de comenzi sh. Variabile, structuri de control (if, for, while, do, case). Comenzi interne utile în context Shell. Variabile shell remarcabile şi rolul acestora. 3. Windows: introducere. Comenzi şi argumente. Fişiere şi căi; drepturi de access. Fişiere de comenzi bat. 4. Sistemul de operare Unix: procese. Procese sub Unix; structura, API (fork, wait, exec, exit, system, popen). 5. Threaduri POSIX Concepte. API: create, exit, join. Variabile mutex. 6. Sistemul de fisiere Unix; operatii I/O. Legături hard şi legături simbolice. Conceptul de montare. Drepturi de acces la fişiere open, close, read, write, lseek, file lock. 7. Teoria sistemelor de operare. Clasificari. Functii. Arhitectura. 8. Procese. Concepte. Concurenta. Semafoare. Sectiune critica si rezultate inconsistente. Impasul. Planificarea	Examination 1 – Public lecture – supporting a course Subjects: A. Operating systems 1. Unix: Introduction. Unix commands and arguments. Regular Expression, generic specification of files. Filters and text editors. 2. Shell Programming. Sh Processors. Variables, control structures (if, for, while, do, case). Embedded commands. Remarcable shell variables. 3. Windows: introduction. Commands and arguments. Files and paths; Access rights Command bat files. 4. OS Unix: processes. Unix processes; structure, API (fork, wait, exec, exit, system, popen). 5. POSIX Threads Concepts. API: create, exit, join. Mutex variables. 6. Unix File System; I/O operations. Hard and symbolic links. Mounting. File access rights open, close, read, write, lseek, file lock. 7. General Theory of Operating Systems Classifications. Functions Architectures. 8. Processes. Concepts Concurrence. Semaphores. Critical sectioons and race conditions Deadlock. Processes scheduling

	proceselor. 9. Gestiunea memoriei. Arhitectura. Alocari: partitionata, paginata, segmentata. Evacuare temporara. Planificarea operatiilor cu memoria. 10. I/O la nivel fizic. Calale de I/O. Zone tampon. Planificarea operatiilor cu discul 11. Sisteme de fisiere. Concepte. Implementari la nivel de baza. Directoare. Jurnalizare; copy-On_write. Exemple: FAT, EXT3, NTFS 12. Incarcarea sistemelor de operare 13. Nucleul Linux 14. Nucleul Windows Bibliografie: 1. Albing C., Vossen J.P., Newham C. bash Cookbook. O'Reilly, 2007 2. Boian F, Vancea A. Boian R. Bufnea D., Sterca A., Cobarzan C., Cojocar D. Sisteme de operare Ed. Risoprint, 2006. 3. Boian F.M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996. 4. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurentă pe platforme Unix, Windows, Java. Ed. Albastră, grupul Microinformatica, Cluj, 2002. 5. Boian F.M.Servicii web;modele, platforme, aplica 2012 6. Lutz M. Learning Python. O'Reilly, 2009. 7. Raymond E.S. The Art of Unix Programming. Prentice Hall, 2003. 8. Stallings W. Operating Systems: Internal and Design Principles. 6th edition, Prentice Hall, 2009. 9. Tanenbaum A.S. Modern Operating Systems. 3rd edition, Prentice Hall, 2009 10. Ubuntu - The Complete Reference. Richard Petersen, MCGraw-Hill, 2009	9. Memory management Architecture Alocatiopns: partitioned, paging, segmentation. Swapping Memory scheduling 10 Phisical I/O I/O chanels Zone tampon. Disk access scheduling 11 File Systems Concepts Low-level implementations. Directories Jurnalization; copy-on-write Example: FAT, EXT3, NTFS 12. Operating systems booting 13. Linux kernel 14. Windows kernel Bibliography: 1. Albing C., Vossen J.P., Newham C. bash Cookbook. O'Reilly, 2007 2. Boian F, Vancea A. Boian R. Bufnea D., Sterca A., Cobarzan C., Cojocar D. Sisteme de operare Ed. Risoprint, 2006. 3. Boian F.M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996. 4. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurentă pe platforme Unix, Windows, Java. Ed. Albastră, grupul Microinformatica, Cluj, 2002. 5. Boian F.M.Servicii web;modele, platforme, aplica 2012 6. Lutz M. Learning Python. O'Reilly, 2009. 7. Raymond E.S. The Art of Unix Programming. Prentice Hall, 2003. 8. Stallings W. Operating Systems: Internal and Design Principles. 6th edition, Prentice Hall, 2009. 9. Tanenbaum A.S. Modern Operating Systems. 3rd edition, Prentice Hall, 2009 10. Ubuntu - The Complete Reference. Richard Petersen, MCGraw-Hill, 2009 11. Windows 7 User Guide. Microsoft, 2009

	11. Windows 7 User Guide. Microsoft, 2009 B. Programare logică şi funcţională (în lb. engleză) 1. Elemente fundamentale ale limbajului Prolog. Fapte şi reguli în Prolog. Obiective. Strategia de control în Prolog. Variabile și propoziții compuse. Variabile anonime. Reguli de potrivire. Modelul de flux. Secțiuni ale unui program Prolog. 2. Programul Prolog. Domenii predefinite. Obiective interne și externe. Predicte cu aritate multiplă. Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Directive de compilare. Expresii aritmetice și comparații. Operații de intrare/ieșire. Șiruri de caractere 3. Backtracking. Controlul backtracking. Predicatul "fail" și "!"(cut). Utilizarea predicatului "!". Tipuri de ”cut”. Predicatul "not". Liste în Prolog. Recursiunea. Exemple de backtracking în Prolog. Găsirea tuturor soluțiilor în același timp. Exemple de predicate în Prolog. Predicte nedeterministe. 4. Obiecte compuse si functori. Unificarea obiectelor compuse. Argumente de mai multe tipuri; liste eterogene. Comparații pentru obiecte compuse. Backtracking cu cicluri. Exemple de proceduri recursive. Cadrul stivei. Optimizare folosind "recursivitatea cozii". Utilizarea predicatului "cut" pentru a păstra "recursivitatea cozii". 5. Structuri de date recursive. Arbori ca structuri de date. Crearea și parcurgerea unui arbore. Arbori de căutare. Baza de date internă a Prolog. Secțiunea "baza de date". Declararea bazei de date interne. Predicte privind operațiile cu baza de date internă. 6. Probleme avansate de Backtracking în Prolog. Gestionarea fișierelor în Prolog. 7. Programare și limbaje de programare. Programare imperativă vs. programare declarativă. Introducere.	 B. Functional and Logical Programming 1. Basic elements of Prolog. Facts and rules in Prolog. Goals. The control strategy in Prolog. Variables and composed propositions. Anonymous variables. Rules for matching. The flow model. Sections of a Prolog program 2. The Prolog program. Predefined domains. Internal and external goals. Multiple arity predicates. The IF symbol (Prolog) and the IF instruction (other languages). Compiler directives. Arithmetic expressions and comparisons. Input/output operations. Strings 3. Backtracking. The backtracking control. The "fail" and "!"(cut) predicates. Using the "!" predicate. Type of cuts. The "not" predicate. Lists in Prolog. Recursion. Examples for backtracking in Prolog. Finding all solutions in the same time. Examples of predicates in Prolog. Non-deterministic predicates 4. Composed objects and functors. Unifying composed objects. Arguments of multiple types; heterogeneous lists. Comparisons for composed objects. Backtracking with cycles. Examples of recursive procedures. The stack frame. Optimization using the "tail recursion". Using the "cut" predicate in order to keep the "tail recursion". 5. Recursive data structures. Trees as data structures. Creating and traversing a tree. Search trees. The internal database of Prolog. The "database" section. Declaration of the internal database. Predicates concerning operations with the internal database. 6. Advanced issues of Backtracking in Prolog. Files management in Prolog. 7. Programming and programming languages. Imperative programming vs. declarative programming. Introduction. The importance of the functional

	Importanța programării funcționale ca o nouă metodologie de programare. Istoria și prezentarea LISP. 8. Elemente de bază în LISP. Structuri de date dinamice. Reguli sintactice şi semantice. Clasificarea funcţiilor în LISP. Funcții primitive în LISP. Predicte de bază în LISP. 9. Predicte pentru liste; pentru numere. Funcții logice şi aritmetice. Definirea funcțiilor utilizatorului. Forma condițională. Metoda de colectare a variabilelor. 10. Gestionarea simbolurilor. Alte funcţii de accesare a listelor. OBLIST şi ALIST. Funcții distructive. Comparații. Alte funcții interesante. 11. Mecanisme de definire. Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA, expresii LABEL. Generatoare, argumente funcționale. Funcțiile MAP. Forme iterative. 12. Alte elemente în LISP. Structuri de date. Macrodefiniții. Argumente opționale. Bibliografie: 1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp şi Prolog. Aplicaţii în Inteligenţa Artificială, Editura Albastră, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp şi Prolog, Editura Albastră, ClujNapoca, 2003 3. http://www.ifcomputer.com/PrologCourse, Lecture on Prolog 4. http://www.lpa.co.uk, Logic Programming 5. FIELD A., Functional Programming, Addison Wesley, New York, 1988. 6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984. C. Medii de proiectare și programare (în lb. engleză)	programming as a new programming methodology. History and presentation of LISP 8. Basic elements in Lisp. Dynamic data structures. Syntactic and semantic rules. Functions' classification in Lisp. Primitive functions in Lisp. Basic predicates in Lisp. 9. Predicates for lists; for numbers. Logic and arithmetic functions. Defining user functions. The conditional form. The collecting variable method. 10. Symbols' managing. Other functions for lists' accessing. OBLIST and ALIST. Destructive functions. Comparisons. Other interesting functions. 11. Definitional mechanisms. The EVAL form. Functional forms; the functions FUNCALL and APPLY. LAMBDA expressions, LABEL expressions. Generators, functional arguments. MAP functions. Iterative forms. 12. Other elements in Lisp. Data structures. Macrodefinitions. Optional arguments Bibliography: 1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp şi Prolog. Aplicaţii în Inteligenţa Artificială, Editura Albastră, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp şi Prolog, Editura Albastră, ClujNapoca, 2003 3. http://www.ifcomputer.com/PrologCourse, Lecture on Prolog 4. http://www.lpa.co.uk, Logic Programming 5. FIELD A., Functional Programming, Addison Wesley, New York, 1988. 6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.

	1. Automatizarea construcției, gestionarea dependențelor; sisteme de control al versiunilor 2. JDBC 3. Containere de inversiune a controlului 4. Arhitectura client-server 5. Apel de procedură la distanță 6. Maparea relațională a obiectelor 7. Maparea relațională a obiectelor 8. Integrarea aplicațiilor întreprinderii 9. Integrarea aplicațiilor întreprinderii 10. Servicii web 11. Aplicații web 12. Socluri web 13. Securitatea web 14. Baze de date NoSql Bibliografie: 1. Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015. 2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design and Unified Process, Berlin, Prentice Hall, 2002. 3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 4. Hohpe, G., Woolf, B., Enterprise integration patterns, Addison-Wesley, 2003. 5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 6. ***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/ 7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 8. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015. 9. Spring http://projects.spring.io/spring-framework	C. Systems for Design and Implementation 1. Build automation, dependency management; version control systems 2. JDBC 3. Inversion of control containers 4. The client server architecture 5. Remote procedure call 6. Object relational mapping 7. Object relational mapping 8. Enterprise application integration 9. Enterprise application integration 10. Web services 11. Web applications 12. Web sockets 13. Web security 14. NoSql databases Bibliography: 1. Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015. 2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design and Unified Process, Berlin, Prentice Hall, 2002. 3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 4. Hohpe, G., Woolf, B., Enterprise integration patterns, Addison-Wesley, 2003. 5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 6. ***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/ 7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 8. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015.

	D. Programare pentru dispozitive mobile (în lb. engleză) 1. Instrumente Android de bază. Android Studio. Ciclul de viață al activităților/fragmentelor. Interfețe utilizator. 2. Liste și restul resurselor. Vizualizări. Procesare în fundal. Rețele. 3. Master-details și resurse rest. Mai multe vizualizări: NavigationDrawer. OkHttp, JsonReader, JsonWriter. ContentProviders. 4. Persistență locală. Preferințe și fișiere. Baze de date: SQLite, Room, Realm. 5. Securizarea aplicațiilor mobile. Modelul de securitate Android. Token-uri web JSON. OAuth 2.0 6. Sincronizarea datelor. WebSockets. Servicii locale de sincronizare. LoaderManagers. 7. Programare reactivă. Realm - bază de date în timp real. Rx - programare reactivă. Coroutine. 8. Servicii de sistem și senzori. Servicii. Procese. Senzori. 9. Animații. ValueAnimator. ObjectAnimator. Cadrul tranzițiilor. 10. Servicii Firebase. Autentificare. Bază de date. Configurare la distanță. 11. Monetizare. Reclame. Facturare în aplicație. Firebase. 12. Conștientizare și apropiere. Anticipare și reacție. Apropiat. Web fizic. 13. Testați-vă aplicația. Junit. Mockito. UI Automator, Expresso. Laborator de testare Firebase. Testarea performanței. 14. Simularea examenului și discuții. Exemplu de cerințe pentru examen. Simularea examenului live. Bibliografie: 1. Android Development.	9. Spring http://projects.spring.io/spring-framework D. Mobile Application Programming 1. Base Android tooling. Android Studio. Activity/Fragment lifecycle. User interfaces. 2. Lists and rest resources. Views. Background processing. Networking. 3. Master-details and rest resources. More views: NavigationDrawer. OkHttp, JsonReader, JsonWriter. ContentProviders. 4. Local persistence. Preferences and Files. Databases: SQLite, Room, Realm. 5. Securing mobile apps. Android security model. JSON Web Tokens. OAuth 2.0 6. Synchronizing data. WebSockets. Local synchronization services. LoaderManagers. 7. Reactive programming. Realm - real-time database. Rx - reactive programming. Coroutines. 8. System services and sensors. Services. Processes. Sensors. 9. Animations. ValueAnimator. ObjectAnimator. Transitions framework. 10. Firebase Services. Authentication. Database. Remote Config. 11. Monetize. Ads. In-app billing. Firebase. 12. Awareness and nearby. Anticipate and react. Nearby. Physical Web. 13. Test your app. Junit. Mockito. UI Automator, Expresso. Firebase test lab. Performance testing. 14. Exam simulation and discussions. Sample exam requirement. Live exam simulation. Bibliography: 1. Android Development. http://developer.android.com/index.html

	http://developer.android.com/index.html 2. React Native. https://facebook.github.io/react-native/ 3. Flutter. https://flutter.io/docs 4. Vogella. Android Development Tutorials. http://www.vogella.com/android.html	2. React Native. https://facebook.github.io/react-native/ 3. Flutter. https://flutter.io/docs 4. Vogella. Android Development Tutorials. http://www.vogella.com/android.html
Descrierea procedurii de concurs	Comisia de concurs evaluează candidaţii ţinând cont de următoarele criterii: • Conținutul dosarului individual (pondere 80%); • Susținerea unui curs (proba 1) (pondere 20%). Nota finală a fiecărui candidat se calculează ca medie ponderată a notelor obţinute la criteriile de mai sus. Fiecare membru al comisiei (inclusiv preşedintele) întocmește un referat individual de apreciere care propune o notă finală pentru fiecare candidat. În evaluarea activității științifice se va ține cont de calitatea publicațiilor și contribuțiile candidaților în raport cu exigențele prevăzute în norma didactică sau de cercetare. Candidații eligibili pentru ocuparea postului scos la concurs trebuie să obțină: • cel puțin nota 7 (șapte) la fiecare criteriu; • media generală cel puțin 8,50 (opt și 50%). Preşedintele comisiei de concurs întocmeşte un raport de sinteza 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	The committee evaluates the candidates taking into account the following criteria: • the content of the candidate file (80%); • public lecture (examination 1) (20%). The final grade of every candidate will be computed as the weighted average of grades obtained for the above criteria. Each member of the committee (including the president) writes an individual report and proposes a final grade for each candidate. When evaluating the scientific results, the committee will consider the quality of the publications and contributions of the candidates having as reference the requirements specified by the didactic and research norm. Eligible candidates must obtain: • at least grade 7 (seven) for each criterion; • the overall average at least 8.50 (eight and 50%). The committee president will write a synthesis report containing the final grades assigned to the candidates by the committee members and indicates the overall average obtained by each candidate, calculated as the arithmetic mean of the individual final grades. The overall average thus obtained represents the result of the competition for each candidate. Based on the general average, the committee decides the hierarchy of the candidates and nominates the eligible candidate who has

	întrunit cel mai bun rezultat în concurs. Președintele comisiei de concurs supune raportul 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.	qualified the best in the competition. The committee members will vote the final report of the committee president; the voting will be open. As a result of the voting, the president acknowledges the result of the vote, communicates it to the committee members and mentions it in the conclusion of the competition report, together with an indication of the number of votes "for" and "against". In the case when the vote "for" is not granted by the majority of the committee members, the position is not taken by any candidate. The hierarchy of the candidates as it results from the overall average grades obtained by each candidate can not be altered by the committee's vote. The competition report is signed by each committee member and by the committee president.
Perioada de comunicare a rezultatelor	19.06.2025-20.06.2025	19.06.2025-20.06.2025
Perioada de depunere a contestațiilor	23.06.2025-25.06.2025	23.06.2025-25.06.2025
Salariul minim de încadrare a postului la momentul angajării	7699	7699
Lista completă a documentelor pe care candidații trebuie să le includă în dosarul de concurs	https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata	https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata
Adresa la care trebuie trimis dosarul de concurs	Registratura Universității "Babeș-Bolyai", (camera P20), str. M. Kogălniceanu nr. 1, Cluj-Napoca	Main building of Babes-Bolyai University, (Room P20) M. Kogălniceanu Street, 1, Cluj-Napoca