

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai din Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematică și Informatică</b>
1.3 Departamentul	<b>Departamentul de Matematică și Informatică al Liniei Maghiare</b>
1.4 Domeniul de studii	<b>Informatică</b>
1.5 Ciclul de studii	<b>Postuniversitar</b>
1.6 Programul de studiu / Calificarea	<b>Program postuniversitar de informatică și dezvoltare software (în limba maghiară)</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Fundamentele Programării și Algoritmicii</b>						
2.2 Titularul activităților de curs	<b>Dr. Kolumbán Sándor</b>						
2.3 Titularul activităților de seminar	<b>Dr. Kolumbán Sándor</b>						
2.4 Anul de studiu	<b>1</b>	2.5 Semestrul	<b>1</b>	2.6. Tipul de evaluare	<b>E</b>	2.7 Regimul disciplinei	<b>Obligatorie</b>
Codul disciplinei	<b>MLM5121</b>						

### 3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	5	Din care: 3.2 curs	2	3.3 seminar/laborator	3
3.4 Total ore din planul de învățământ	50	Din care: 3.5 curs	20	3.6 seminar/laborator	30
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					6
Examinări					4
Alte activități: .....					-
3.7 Total ore studiu individual	75				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	-
4.2 de competențe	-

## 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	Sală cu proiector
5.2 De desfășurare a seminarului/laboratorului	Laborator echipat cu calculatoare cu Java

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>C1.1. Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</p> <p>C1.2. Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază.</p> <p>C1.3. Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.</p> <p>C4.1. Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice.</p> <p>C3.2. Identificarea și explicarea modelelor informatice de baza adecvate domeniului de aplicare.</p> <p>C4.2. Interpretarea de modele matematice și informatice (formale).</p>
<b>Competențe transversal</b>	<p>CT1. Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul științific și didactic, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională.</p> <p>CT3. Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"><li>• Să cunoască conceptele de bază a programării procedurale.</li><li>• Să învețe elementele de bază a limbajului Java (din afara elementelor legate de programare orientată obiect)</li></ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"><li>• Înțelegerea conceptelor de bază ale programării</li><li>• Folosirea instrumentelor de bază pentru scrierea programelor</li><li>• Învățarea elementară a limbajului Java</li><li>• Cunoașterea instrumentelor de dezvoltare, execuție și depanare</li><li>• Înțelegerea considerațiilor stilistice pentru scrierea programelor</li></ul>

## 8. Conținuturi

8.1	Curs	Metode de predare	Observații
1	Introducere la programare procedurală. Construcții și tipuri elementare in Java. Variabile și atribuire. Inștrucțiuni. Citire și afișare simplă pe ecran.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
2	Elementele codului de sursă. Semantica și sintactica limbii. Secvențe, joncțiune, cicluri. Structuri de control complexe, expresii logice complexe.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
3	Colecții și indexare. Algoritmuri simple bazate pe colecții. Folosirea fișierelor. Șiruri cu simbol de sfârșit. Enumerare, acumulare, găsirea valorilor extreme.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
4	Tipuri avansate. Funcții. Ciclul de viață a variabilelor. Bazele refactorării. Precedența operatorilor. Efectele secundare ale funcțiilor. Valorile stânga și dreapta (lvalue, rvalue). Ordinul de evaluare.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
5	Mașina de stat. Segmentarea programelor. Comentarii și documentare.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
6	Recurzivitate directă și indirectă. Probleme tipice: ciclul infinit, stack overflow. Exemple.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
7	Algoritmi folosind tipuri elementare. Algoritmuri de căutare: <ul style="list-style-type: none"> <li>- cautare lineară</li> </ul> Algoritmuri de sortare: <ul style="list-style-type: none"> <li>- BubbleSort</li> <li>- SelectionSort</li> <li>- InsertionSort</li> </ul> Complexitate lor de timp și spațiu.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
8	Metoda divizării. Algoritmuri de căutare: <ul style="list-style-type: none"> <li>- cautare binară</li> </ul> Algoritmuri de sortare: <ul style="list-style-type: none"> <li>- MergeSort</li> <li>- QuickSort (numai interfață)</li> </ul> Complexitate lor de timp și spațiu. Sortare stabilă și instabilă.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	
9	Backtracking.	<ul style="list-style-type: none"> <li>- Explicatie</li> <li>- Exemple</li> <li>- Conversatie</li> </ul>	

10	Recapitulare.		
----	---------------	--	--

Bibliografie:

1. Lovász László, Algoritmusok bonyolultsága, <http://web.cs.elte.hu/~kiralyl/Algbony.pdf>
2. Fóthi Ákos, Bevezetés a programozáshoz, <http://compalg.inf.elte.hu/~tony/KedvencKonyvek/InfoKonyvtar/04-Bevezetes%20a%20programozasba/Bevezetes%20a%20programozasba-Konyv.pdf>
3. Király Zoltán, Algoritmuselmélet, <http://web.cs.elte.hu/~kiralyl/Algoritmusok.pdf>
4. Fleiner Tamás, A számítástudomány alapjai, <http://www.cs.bme.hu/~fleiner/jegyzet/NESZ.pdf>

8.2	Seminar / laborator	Metode de predare	Observații
1	Compilarea și executarea programelor predate. Modificarea programelor furnizate.		
2	Probleme standard de programare cu operațiuni elementare.		
3	Folosirea colecțiilor, citire și afișare folosind fișiere.		
4	Evaluarea și depanarea programelor furnizate.		
5	Mașină de state pentru JSON.		
6	Turnul din Hanoi. Implementare și măsurare.		
7	Implementarea și măsurarea algoritmilor predate.		
8	Implementarea și măsurarea algoritmilor predate.		
9	Implementarea algoritmului cu regine, sodoku și no-3-in line.		
10	Recapitulare.		

Bibliografie:

1. Lovász László, Algoritmusok bonyolultsága, <http://web.cs.elte.hu/~kiralyl/Algbony.pdf>
2. Drótos Márton, Algoritmuselmélet feladatgyűjtemény, <http://www.cs.bme.hu/~drotos/algfgy.pdf>

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

- Cursul respectă curricula IEEE și ACM pentru domeniul Informatică.
- Cursul există în programele de studiu ale universităților importante din România și din străinătate.
- Conținutul disciplinei este considerat de majoritatea companiilor software ca fiind deosebit de important pentru obținerea unor abilități medii de programare.

## 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunostintele acumulate	Examen scris	50%
10.5 Seminar/laborator	Test scurt pe seminar	Examen practic	15%
	Teme de laborator	Evaluare continuă	35%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"><li>Fiecare cursant trebuie să demonstreze că a atins un nivel acceptabil de cunoștințe și înțelegere a domeniului, că este capabil să prezinte aceste cunoștințe într-o manieră coerentă și că are abilitatea de a stabili anumite conexiuni și de a folosi aceste cunoștințe în rezolvarea diferitelor probleme.</li><li>Promovarea este condiționată de nota minimă 5 la activitatea de laborator, și examenul scris.</li></ul>			

Data completării

30.08.2020

Semnătura titularului de curs

Dr. Kolumbán Sándor

Semnătura titularului de seminar

Dr. Kolumbán Sándor

Data avizării în departament

Semnătura directorului de departament

Conf.univ.dr. András Szilárd Károly