

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Matematică-Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Matematică Informatică – limba de studiu română

### 2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Algoritmi și Programare					
(en)		Algorithms and Programming					
2.2 Titularul activităților de curs			Lect. Dr. Ionescu Vlad-Sebastian				
2.3 Titularul activităților de seminar			Lect. Dr. Ionescu Vlad-Sebastian				
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6. Tipul de evaluare	C	2.7 Regimul disciplinei	Obligatorie
2.8 Codul disciplinei		MLR5005					

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

3.1 Număr de ore pe săptămână	6	Din care: 3.2 curs	2	3.3 seminar/laborator	2 sem 2 lab
3.4 Total ore din planul de învățământ	84	Din care: 3.5 curs	28	3.6 seminar/laborator	56
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					8
Examinări					18
Alte activități: .....					
3.7 Total ore studiu individual		66			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

### 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, plus proiector
5.2 De desfășurare a seminarului/laboratorului	• Laboratoare echipate cu medii de dezvoltare Python

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<ul style="list-style-type: none"> <li>• C1.1 Descrierea adecvată a paradigmatelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</li> <li>• C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</li> <li>• 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</li> <li>• C1.4 Testarea unor aplicații pe baza unor planuri de test</li> <li>• C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente</li> </ul>
<b>Competențe transversale</b>	<ul style="list-style-type: none"> <li>• CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</li> <li>• 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</li> </ul>

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Să cunoască conecpele de baza ale ingineriei software (proiectare, implementare și întreținere) și să învețe limbajul de programare Python.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Să cunoască conecpele de baza ale programarii</li> <li>• Să cunoască conecpele de baza ale ingineriei software</li> <li>• Să folosească instrumente de baza pentru construirea programelor</li> <li>• Să învețe limbajul Python si instrumente de dezvoltare pentru programarea, executia si depanarea programelor Python.</li> <li>• Să-și însușeasca un stil de programare conform celor mai bune recomandări practice.</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
<b>1. Introducere în procese de dezvoltare software</b> <ul style="list-style-type: none"> <li>• Ce este programarea: algoritm, program, elemente de bază Python, interpretor Python, roluri în ingineria software</li> <li>• Cum scriem programe: enunț problemă, cerințe, proces de dezvoltare dirijat de funcționalități</li> <li>• Exemple: calculator</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
<b>2. Programare procedurală</b> <ul style="list-style-type: none"> <li>• Tipuri structurate: liste, tuple, dicționare</li> <li>• Funcții: cazuri de testare, definiție, variabile, apel, transmiterea parametrilor, funcții anonime</li> <li>• Cum scriem funcții: programare dirijată de teste, refactorizări</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	

<b>3. Programare modulara</b> <ul style="list-style-type: none"> <li>• Ce este un modul: modul Python, domeniul variabilelor, pachete, module standard, distribuire module</li> <li>• Cum organizam codul sursa: responsabilitati, single responsibility principle, separation of concerns, dependency, coupling, cohesion</li> <li>• Eclipse+PyDev</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
<b>4. Tipuri definite de utilizator</b> <ul style="list-style-type: none"> <li>• Cum definim tipuri noi</li> <li>• Incapsulare, ascunderea informatiei, tipuri abstracte de date</li> <li>• Introducere programarea orientata obiect –clasa implementarea unui TAD</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>5. Programarea orientata obiect</b> <ul style="list-style-type: none"> <li>• Mecanismele programarii orientate obiect</li> <li>• Dezideratele proiectarii obiectuale <ul style="list-style-type: none"> <li>○ Cuplare redusa</li> <li>○ Coeziune ridicata</li> <li>○ Abstractizare corespunzatoare</li> <li>○ Complexitate gestionabila</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>6. Programarea orientata obiect</b> <ul style="list-style-type: none"> <li>• Principii si sabloane de proiectare: <ul style="list-style-type: none"> <li>○ SOLID: single responsibility, open-closed....</li> <li>○ DDD: entity, value object, repository, service</li> <li>○ IoC, app coordinator, dependency injection</li> <li>○ GOF: singleton, strategy</li> </ul> </li> <li>• Problema: program cu operatii CRUD pe entitati de un tip dat</li> <li>• Arhitectura stratificata: UI, Domeniu, Infrastructura</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>7. Proiectarea programelor</b> <ul style="list-style-type: none"> <li>• Diagrame UML</li> <li>• Sabloane Grasp</li> <li>• Exceptii</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>8. Persistenta și procesarea datelor</b> <ul style="list-style-type: none"> <li>• DTO</li> <li>• Filter, lambda</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>9. Testarea si inspectarea programelor</b> <ul style="list-style-type: none"> <li>• Black box testing, white box testing</li> <li>• Unit testing, integration testing</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> </ul>	

<ul style="list-style-type: none"> <li>• Program inspection: coding style, refactoring</li> </ul>	<ul style="list-style-type: none"> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>10. Complexitatea Algoritmilor, recursivitate, căutare</b> <ul style="list-style-type: none"> <li>• Recursivitate directa si indirecta</li> <li>• Exemple</li> <li>• Căutare secvențială</li> <li>• Căutare binară</li> <li>• Complexitatea algoritmilor</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>11 Algoritmi de sortare</b> <ul style="list-style-type: none"> <li>• BubbleSort</li> <li>• SelectionSort</li> <li>• InsertionSort</li> <li>• QuickSort</li> <li>• MergeSort</li> <li>• Cmplexitatea algoritmilor</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>12. Metode de rezolvare a problemelor</b> <b>Metoda divizarii</b> <ul style="list-style-type: none"> <li>• Descriere Metoda</li> <li>• Exemple</li> </ul> <b>Metoda Backtracking</b> <ul style="list-style-type: none"> <li>• Algoritmul Backtracking</li> <li>• Extensii ale algoritmului</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demosntrație didactică</li> </ul>	
<b>13. Programare dinamică</b> <ul style="list-style-type: none"> <li>• Descrierea metodei</li> <li>• Exemple</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Conversație</li> </ul>	
<b>14. Colocviu - Evaluare</b>	<ul style="list-style-type: none"> <li>•</li> </ul>	
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. Kent Beck. <i>Test Driven Development: By Example</i>. Addison-Wesley Longman, 2002. See also Test-driven development. <a href="http://en.wikipedia.org/wiki/Test-driven_development">http://en.wikipedia.org/wiki/Test-driven_development</a></li> <li>2. Martin Fowler. <i>Refactoring. Improving the Design of Existing Code</i>. Addison-Wesley, 1999. See also <a href="http://refactoring.com/catalog/index.html">http://refactoring.com/catalog/index.html</a></li> <li>3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. <i>Introduction to Algorithms 3rd ed.</i>, 2009.</li> <li>4. Craig Larman. <i>Applying UML and Patterns. An Introduction to Object Oriented Analysis and Design</i>, 2004.</li> <li>5. <i>The Python language reference</i>. <a href="http://docs.python.org/py3k/reference/index.html">http://docs.python.org/py3k/reference/index.html</a></li> <li>6. <i>The Python standard library</i>. <a href="http://docs.python.org/py3k/library/index.html">http://docs.python.org/py3k/library/index.html</a></li> </ol> <p><i>The Python tutorial</i>. <a href="http://docs.python.org/tutorial/index.html">http://docs.python.org/tutorial/index.html</a></p>		
8.2 Seminar / laborator	Metode de predare	Observații
1. Programe Python	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
2. Programare procedurală	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> </ul>	

	<ul style="list-style-type: none"> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
3. Programare modulară	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
4. Tipuri definite de utilizator	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
5. Principii de proiectare	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
6. POO	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
7. Proiectare	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
8. Testare și inspectare	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
9. Recursivitate. Complexitatea algoritmilor	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
10. Metoda injumătățirii. Algoritmi de căutare	<ul style="list-style-type: none"> <li>• Expunere</li> </ul>	

	<ul style="list-style-type: none"> <li>• interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
11. Algoritmi de sortare	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
12. Backtracking	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
13. Pregătirea examenului practic	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	
14: Pregătirea examenului scris	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicație</li> <li>• Conversație</li> <li>• Exemple</li> <li>• Demonstrație didactică</li> </ul>	

#### Bibliografie

1. Kent Beck. *Test Driven Development: By Example*. Addison-Wesley Longman, 2002. See also Test-driven development. [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)
  2. Martin Fowler. *Refactoring. Improving the Design of Existing Code*. Addison-Wesley, 1999. See also <http://refactoring.com/catalog/index.html>
  3. Frentiu, M., H.F. Pop, Serban G., *Programming Fundamentals*, Cluj University Press, 2006
  4. *The Python language reference*. <http://docs.python.org/py3k/reference/index.html>
  5. *The Python standard library*. <http://docs.python.org/py3k/library/index.html>
- The Python tutorial*. <http://docs.python.org/tutorial/index.html>

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

- Cursul respecta recomandările IEEE și ACM legate de Curricula pentru specializarea Informatică.
- Cursul face parte din programul de studiu de la majoritatea universităților importante din România și din străinătate.
- Conținutul cursului este considerat de companiile soft ca fiind important pentru un nivel mediu de cunoștințe în programare.

## 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Corectitudinea și completitudinea cunoștințelor acumulate. Capacitatea de a proiecta și implementa programe scrise în limbajul Python	Examen scris	30%
10.5 Seminar/laborator	Abilitatea de a scrie și depăna un program Python	Examen practic	40%
	Programele scrise în timpul semestrului	Documentație	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"><li>• Minimum 5 la fiecare proba.</li></ul>			

Data completării

06.05.2019

Semnătura titularului de curs

Lect. Dr. Ionescu Vlad-Sebastian

Semnătura titularului de seminar

Lect. Dr. Ionescu Vlad-Sebastian

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Anca Andreica