

FIȘA DISCIPLINEI

1. Date despre program

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

2. Date despre disciplină

2.1 Denumirea disciplinei	Multi-paradigm programming languages						
2.2 Titularul activităților de curs	Conf. Dr. Niculescu Virginia						
2.3 Titularul activităților de seminar	Conf. Dr. Niculescu Virginia						
2.4 Anul de studiu	3	2.5 Semestrul	6	2.6. Tipul de evaluare	colocviu	2.7 Regimul disciplinei	Optional

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	11a b+ 1pr
3.4 Total ore din planul de învățământ	48	Din care: 3.5 curs	24	3.6 seminar/laborator	24
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					27
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					35
Tutoriat					4
Examinări					10
Alte activități:					-
3.7 Total ore studiu individual			77		
3.8 Total ore pe semestru			125		
3.9 Numărul de credite			5		

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele programării, Programare Orientată Obiect, Programare funcțională, Java, C++
4.2 de competențe	Abilități de implementare programe și abilități de abstractizare

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	Sala cu proiector
5.2 De desfășurare a seminarului/laboratorului	laborator cu stații de lucru

6. Competențele specifice acumulate

C.4.1 Definierea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice

C.4.2 Interpretarea de modele matematice și informatice (formale)

C.4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale

C 4.4 Utilizarea simulării pentru studiul comportamentului modelelor realizate și evaluarea performanțelor

C 4.5 Încorporarea de modele formale în aplicații specifice din diverse domenii

Competențe profesionale	<ul style="list-style-type: none"> Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a subiectului și că este capabil (a) să exprime aceste cunoștințe; și, de asemenea, că poate folosi aceste cunoștințe în rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă.
Competențe transversale	<ul style="list-style-type: none"> 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 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

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> Aprofundarea conceptelor de baza ale programării Abilitatea de a lucra independent sau în echipă pentru a rezolva probleme într-un context de programare paralelă. Cunoaștere și abilități de implementare a paradigmatelor și tehnicilor de baza ale programării paralele.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> Introducere în programare în limbajul Scala și în alte limbaje multiparadigma Aprofundarea cunoștințelor de programare în Java, C#, C++, Scala. -

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Paradigme de programare: generalități	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
2. Bazele programării în Scala	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
3. Programare orientată obiect în Scala	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
4. Programare funcțională în Scala	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
5. Concurența în Scala	Expunere, descriere, explicație, exemple, discuții ale unor	

	studii de caz.	
6. Paradigme de programare paralela in Scala	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7. Introducere in Scala coroutines	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
8. Actor programming in Scala	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
9. Metaprogramming	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
10. Dataflow programming	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
11. Prezentare referate si proiecte	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
12. Prezentare referate si proiecte	Discutii, analize, evaluari.	

Bibliografie

1. Michael Scott. Programming Language Pragmatics. 4th ed. Morgan Kaufmann, 2015
2. Martin Odersky, Lex Spoon, Bill Venners. Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition
3. Paul Chiusano and Runar Bjarnason Functional Programming in Scala, Mannon, 2014.
4. Bjarne Stroustrup: The C++ Programming Language Special Edition, Addison-Wesley, 2000
5. Andrei Alexandrescu. Modern C++ Design: Generic Programming and Design Patterns Applied Addison-Wesley Professional.2001
6. Georgy Pashkov Multi-Paradigm Programming with Modern C++. Packt Publishing.2020
7. David Vandevorde, Douglas Gregor, Nicolai M. Josuttis. C++ Templates: The Complete Guide, 2nd Edition. Addison-Wesley Professional. 2017 Chapter: C++ Metaprogramming.
8. C.D. Marlin. Coroutines. A Programming Methodology, a Language Design and an Implementation. Springer-Verlag Berlin Heidelberg

8.2 Seminar / laborator	Metode de predare	Observații
1. Exemple simple in Scala	Dialog, dezbateri, explicatie,exemple.	
2. Tema de programare in Scala 1	Dialog, dezbateri, explicatie,exemple.	
3. Tema de programare in Scala 2	Dialog, dezbateri, explicatie,exemple.	
4. Tema de programare in Scala 3	Dialog, dezbateri, explicatie,exemple.	
5. Prezentari studenti	Dialog, dezbateri, explicatie,exemple.	
6. Prezentari ale proiectelor studentilor	Dialog, dezbateri,	

explicatie,exemple.

Bibliografie

1. PMichel Schinz and Philipp Haller. A Scala Tutorial for Java Programmers. <https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html>
2. Scala Tutorial. <https://www.tutorialspoint.com/scala/index.htm>

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

Cursul respecta Recomandarile IEEE and ACM Curricula pentru studii in Computer Science;
Cursuri cu tematica similara exista in programele de studii ale majoritatii universitatilor din tara si strainatate.

Cursul defineste dobandirea unor abilitati care constituie avantaje evidentiata de potentialele firme angajatoare din domeniu.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Teme de laborator	colocviu	Laborator: 40%
10.5 Laborator	Referat care prezinta un limbaj multiparadigma si un mini-proiect in limbajul prezentat		Referat+Proiect= 60%
10.6 Standard minim de performanță			
Minim 5 la media finala.			

Data completării

.

Titular de curs

.....Niculescu Virginia.....

Titular de laborator

.....Niculescu Virginia.....

Data avizării în departament

.....

Director de departament

.....