

# FIȘA DISCIPLINEI

## Verificarea și Validarea Sistemelor de Programe Anul universitar 2020-2021

### 1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Științe, Educație Fizică și Informatică
1.3	Departamentul	Matematică-Informatică
1.4	Domeniul de studii	Informatică
1.5	Ciclul de studii	Licență
1.6	Programul de studii / Calificarea	Informatică / Informatică

### 2. Date despre disciplină

2.1	Denumirea disciplinei	<b>Verificarea și validarea sistemelor de programe</b>									
2.2	Titularul activităților de curs	Lect. univ. dr. Tudose Cristina Vasilica									
2.3	Titularul activităților de laborator	Lect. univ. dr. Tudose Cristina Vasilica									
2.4	Anul de studii	III	2.5	Semestrul	I	2.6	Tipul de evaluare	E	2.7	Regimul disciplinei	O

### 3. Timpul total estimat

3.1	Număr de ore pe săptămână	3	3.2	din care curs	2	3.3	laborator	1
3.4	Total ore din planul de învățământ	42	3.5	din care curs	28	3.6	laborator	14
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								25
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								20
Tutorat								2
Examinări								6
Alte activități								-
3.7	Total ore studiu individual				83			
3.8	Total ore pe semestru				125			
3.9	Număr de credite				5			

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

4.1	De curriculum	Elemente fundamentale de programare orientată pe obiecte.
4.2	De competențe	Capacitate de analiză și sinteză, abilități de programare.

### 5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală cu o capacitate de minim 100 locuri dotată cu videoproiector și ecran de proiecție, o tablă.
5.2	De desfășurare a laboratorului/proiectului	Sala de laborator dotată cu calculatoare, capacitate maximă 18 studenți/laborator.

### 6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> <li>Dezvoltarea și întreținerea aplicațiilor informatice;</li> <li>Utilizarea instrumentelor informatice in context interdisciplinar;</li> <li>Utilizarea bazelor teoretice ale informaticii si a modelelor formale.</li> </ul>
Competențe transversale	<ul style="list-style-type: none"> <li>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>Desfășurarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse;</li> <li>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

7.1	Obiectivul general al disciplinei	Disciplina are ca obiectiv general însușirea de către studenți a cunoștințelor de bază, metodelor și tehnicilor privind tehnicile automate de testare și validare a sistemelor de programe, a modalităților de implementare și de aplicare adecvată la situații concrete a acestor metode și tehnici.
7.2		<i>Obiective cognitive:</i>

Obiectivele specifice	<p>► Cunoașterea principiilor fundamentale ale verificării unui sistem de programe.</p> <p>► Cunoașterea principiilor fundamentale corespunzătoare principalelor metode de testare a sistemelor de programe.</p> <p><i>Obiective procedurale:</i></p> <p>► Formarea deprinderilor de a utiliza metodele de testare ingineresti, folosite pe scara larga in industria de software, cat si abordarile formalizate, folosite in special pentru sistemele safety-critical.</p> <p><i>Obiective atitudinale:</i></p> <p>► Rigurozitate în folosirea metodelor de verificare și validare a sistemelor de programe.</p>
-----------------------	--

## 8. Conținuturi

8.1. Curs		Nr. ore	Metode de predare	Observații Resurse folosite
1.	Metode functionale de testare a sistemelor; clase de echivalenta, valori de frontiera, impartirea in categorii. Exemple	4	Prelegere Dezbateri Explicația Descrierea Exemplificarea Problematizarea Exercițiul	Calculator Videoprojector
2.	Metode structurale de testare a sistemelor; nivele de acoperire a grafului asociat programului. Exemple	4		
3.	Testarea bazată pe mutanți. Exemple	4		
4.	Specificatii algebrice - Automatele finite ca limbaj de specificare	4		
5.	Testarea sistemelor pe baza specificatiilor formale - metode bazate pe masini cu stari finite (W, Wp, UIO, DS)	4		
6.	Logica temporală. CTL, LTL. Exemple	4		
7.	Model checking. Exemple	4		

### Bibliografie:

- M. Huth, M. Ryan, Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, Second Edition 2004.
- M. Holcombe, F. Ipate, Correct Systems: building business process solutions, Springer Verlag, 1998.
- D. MacKenzie, Mechanizing Proof: computing, risk and trust, MIT Press, 2001.
- J. Spivey, The Z Notation: a Reference Manual, Prentice Hall, 1989.
- C. Baier and J-P Katoen, Principles of Model Checking, MIT Press, 2008
- P. Ammann and J. Offutt, Introduction to software testing, Cambridge University Press, 2016

8.2. Aplicații – Laborator		Nr. ore	Metode de predare	Observații Resurse folosite
1	Testare unitara.Prezentare JUnit - aplicatii	4	Studiul de caz Lucrul în grup Problematizarea	Calculator Videoprojector
2	Folosirea de instrumente automate pentru construirea grafului de control al fluxului, pentru generarea acoperirii codului, pentru generare mutanti si rulare teste pe mutanti (CFGGenerator, EclEmma, PitClipse, mujava)	4		
3	Specificarea proprietatilor de tip safety si liveness in CTL, LTL	2		
4	Verificarea modelelor folosind NuSMV, Spin. Exemple	4		

### Bibliografie:

- M. Huth, M. Ryan, Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, Second Edition 2004. ISBN: 0 521 54310.
- M. Holcombe, F. Ipate, Correct Systems: building business process solutions, Springer Verlag, 1998.
- D. MacKenzie, Mechanizing Proof: computing, risk and trust, MIT Press, 2001.
- J. Spivey, The Z Notation: a Reference Manual, Prentice Hall, 1989.
- C. Baier and J-P Katoen, Principles of Model Checking, MIT Press, 2008
- P. Ammann and J. Offutt, Introduction to software testing, Cambridge University Press, 2016
- JUnit 5 user guide - <https://junit.org/junit5/docs/current/user-guide/#overview>
- muJava Home Page (gmu.edu) <https://cs.gmu.edu/~offutt/mujava/>
- NuSMV home page (fbk.eu) <https://nusmv.fbk.eu/>
- Spin - Formal Verification (spinroot.com) <http://spinroot.com/spin/whatispin.html>

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

Competențele dobândite în cadrul disciplinei permit absolvenților să utilizeze eficient modelele specifice tehnicilor automate de testare și validare a sistemelor de programe în rezolvarea cerințelor aferente practicii și cercetării în domeniul informaticii.

## 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	a) examen	a) scris - verificare cunoștințe teoretice și elemente de implementare	50%
10.5 Laborator	a) activitate b) proiect	a) înregistrare activitate b) verificare proiect	20% 30%
10.6 Standard minim de performanță	* Nota minimă 5 la activitățile de laborator și nota minimă 5 la examenul final. * Set de cunoștințe minimale pentru promovarea examenului final: cunoașterea principiilor de bază ale verificării și validării unui sistem de programe folosind metodele prezentate.		

Data completării  
20.09.2020

Titular de curs  
Lect. univ. dr. Tudose Cristina

Titular de laborator  
Lect. univ. dr. Tudose Cristina

Data avizării în Departament  
23.09.2020

Director Departament (prestator)  
Conf.univ.dr. Doru CONSTANTIN

Director Departament (beneficiar)  
Conf.univ.dr. Doru CONSTANTIN