

MD-2045, CHIȘINĂU, STR. Studenților, 9/7, TEL: 022 50-99-08 [www.utm.md](http://www.utm.md)
**TESTAREA SOFTWARE**
**1. Date despre unitatea de curs/modul**

<b>Facultatea</b>	Calculatoare, Informatică și Microelectronică				
<b>Catedra/departamentul</b>	Ingineria Software și Automatică				
<b>Ciclul de studii</b>	Studii superioare de licență, ciclul I				
<b>Programul de studiu</b>	526.5 Tehnologii Informaționale				
<b>Anul de studiu</b>	<b>Semestrul</b>	<b>Tip de evaluare</b>	<b>Categoria formativă</b>	<b>Categoria de opționalitate</b>	<b>Credite ECTS</b>
III (învățământ cu frecvență); IV (învățământ cu frecvență redusă)	6; 8	E	S – unitate de curs de specialitate	A - unitate de curs obligatorie	5

**2. Timpul total estimat**

Total ore în planul de învățământ	Din care				
	Ore auditoriale		Lucrul individual		
	Curs	Laborator/seminar	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
150	30	30/15	-	45	30
150(f/r)	10	8/4	-	50	78

**3. Precondiții de acces la unitatea de curs/modul**

Conform planului de învățământ	Structuri de date și algoritmi, Matematica discretă, Programarea în limbajul C++, Analiza și proiectarea algoritmilor, Limbaje formale și proiectarea compilatoarelor.
Conform competențelor	Însușirea de către studenți a principiilor de elaborare a algoritmilor, limbajelor de programare, metodelor și tehnicilor de formulare a modelelor de calcul și modelarea diferitor fenomene și procese tehnice.

**4. Condiții de desfășurare a procesului educațional pentru**

Curs	Pentru prezentarea materialului teoretic în sala de curs este nevoie de proiector și calculator. Nu vor fi tolerate întârzierile studenților, precum și convorbirile telefonice în timpul cursului.
Laborator/seminar	Studenții vor perfecta rapoarte conform condițiilor impuse de indicațiile metodice. Termenul de predare a lucrării de laborator – o săptămână după finalizarea acesteia. Pentru predarea cu întârziere a lucrării aceasta se depunceață cu 1pct./săptămână de întârziere.

**5. Competențe specifice acumulate**

Competențe profesionale	<b>C2. Privind aspectele organizaționale și informaționale ale securității.</b> <ul style="list-style-type: none"> <li>✓ Identificarea și definirea conceptelor, teoriilor, metodelor și principiilor de elaborare a documentației pentru testare în baza interviurilor cu clientul și a specificației cerințelor clientului.</li> <li>✓ Explicarea conceptelor, teoriilor și metodelor de planificare și proiectare a scenariilor de testare.</li> <li>✓ Aplicarea conceptelor, teoriilor și metodelor de planificare și proiectare a scenariilor de testare.</li> <li>✓ Însușirea conceptelor, principiilor și metodelor de proiectare a testelor în baza specificației cerințelor software.</li> </ul>
Competențe profesionale	<b>C4. Privind metodele și tehnologiile de dezvoltare a soluțiilor în asigurarea și controlul calității produselor software.</b> <ul style="list-style-type: none"> <li>✓ Înțelegerea etapelor ciclului de viața a unui produs software și rolul asigurării și controlului calității</li> </ul>

	<p>pe parcursul acestor etape;</p> <ul style="list-style-type: none"> <li>✓ Formarea capacităților de a putea aplica metodele și tehnicile proiectare a testelor în baza specificației cerințelor software;</li> <li>✓ Cunoașterea și înțelegerea etapelor ciclului de viață a erorilor software;</li> <li>✓ Înțelegerea etapelor procesului de testare;</li> <li>✓ Familiarizarea cu elementele de management a testării software.</li> </ul>
Competențe transversale	<b>CT2.</b> Identificarea, descrierea și derularea activităților organizate într-o echipă cu dezvoltarea capacităților de comunicare și colaborare, dar și cu asumarea diferitelor roluri prin realizarea proiectului de an cu utilizarea corectă a surselor bibliografice și metodelor specifice, precum și susținerea acestora.

### 6. Obiectivele unității de curs/modulului

Obiectivul general	Studierea metodelor de testare și verificare a programelor, principiilor de bază ale tehnicilor de verificare a corectitudinii programelor, cu menținerea unui echilibru între considerațiile teoretice și cele practice
Obiectivele specifice	Să posede cunoștințe despre metodele de elaborare a sistemelor de testare automată, despre erorile ce apar în procesul de elaborare a produselor soft; despre verificarea formalizată a programelor; noțiunile de logică în procesul de verificare a corectitudinii programelor

### 7. Conținutul unității de curs/modulului

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
<b>Tematica prelegerilor</b>		
<b>T1. ASPECTELE CUPRINSE ÎN STUDIUL TESTĂRII PROGRAMELOR.</b> Definiția erorii software. Cauzele apariției erorilor. Costul erorilor. Principiile de bază ale testării software. Rolul și scopurile testerului în procesul de dezvoltare software. Axiomele testării. Termeni și definiții ale testării.	6	2
<b>T2. TEHNICI BLACK-BOX DE PROIECTARE A TESTELOR.</b> Testarea funcțională. Metoda cutiei negre. Testarea claselor de echivalență. Testarea valorilor de la limite. Testarea tabelor de decizie. Testarea stărilor și tranzițiilor. Testarea grafului cauză-efect	6	2
<b>T3. TEHNICI WHITE-BOX DE PROIECTARE A TESTELOR.</b> Testarea structurală. Aspecte ale testării structurilor liniare, alternative, iterative. Testarea fluxului de control. Niveluri de acoperire cu teste. Graful fluxului de control. Testarea căilor de bază. Numărul ciclomatic. Testarea fluxului de date. Testarea statică a fluxului de date. Testarea dinamică a fluxului de date	4	2
<b>T4. GESTIONAREA ERORILOR ȘI DEFECTELOR SOFTWARE.</b> Lista cu erorile cel mai des întâlnite în procesul de testare software. Ciclul de viață a defectelor soft. Raportarea eficientă a defectelor..	4	1
<b>T5. ETAPELE PROCESULUI DE TESTARE.</b> Ciclul de testare software(SDLC). Planificare și control. Analiza și proiectarea testelor. Implementarea și executarea testelor. Evaluarea și raportarea defectelor. Activități de încheiere a procesului de testare.	4	1
<b>T6. METODOLOGII ȘI MODELE DE DEZVOLTARE A PRODUSELOR SOFTWARE.</b> Metodologia iterativă. Metodologia ciclică. Metodologia ecluză. Modele: Cascadă, V, Spirală, Agile, Prototipizarea. Reverse engineering.	2	1

<b>T7. NIVELELE DE TESTARE.</b> Testarea unităților. Testarea modulelor. Testarea de integrare. Testarea de sistem. Testarea de acceptare.	4	1
<b>Total prelegeri:</b>	<b>30</b>	<b>10</b>
Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
<b>Tematica seminarelor</b>		
<b>S1.</b> Specificarea cerințelor	4	1
<b>S2.</b> Tehnici de testare black-box	4	1
<b>S3.</b> Tehnici de testare white-box	2	1
<b>S4.</b> Raportarea defectelor software.	2	1
<b>S5.</b> Testarea GUI	2	
<b>S6.</b> Testarea nefuncțională	1	
<b>Total seminare:</b>	<b>15</b>	<b>4</b>
Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
<b>Tematica lucrărilor de laborator</b>		
<b>LL1.</b> Analiza și specificarea cerințelor software	6	2
<b>LL2.</b> Proiectarea testelor pe baza tehnicilor black-box	6	2
<b>LL3.</b> Proiectarea testelor pe baza tehnicilor white -box	4	2
<b>LL4.</b> Descrierea și raportarea defectelor	4	2
<b>LL5.</b> Instrumente de testare	4	
<b>LL6.</b> Instrumente de testare	2	
<b>LL7.</b> Automatizarea testării	4	
<b>Total lucrări de laborator:</b>	<b>30</b>	<b>8</b>

### 8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> <li>1. Bălănescu T.; "Corectitudinea algoritmilor", Editura Tehnică, București,1995.</li> <li>2. Marnie L. Hutcheson, <i>Software Testing Fundamentals: Methods and Metrics</i>, John Wiley &amp; Sons, 2003, 408 p.</li> <li>3. Lee Copeland, <i>A Practitioners Guide to Software Test Design</i>, Artech Hous, 2004, 294 p.</li> <li>4. Ron Patton, <i>Software Testing</i>, Sams Publishing, 2005, 408 p.</li> <li>5. Glenford J. Myers, <i>The Art of Software Testing</i>, John Wiley &amp; Sons, 2004.</li> <li>6. Майерс Г. Надёжность программного обеспечения. – М.: Мир, 1980.</li> <li>7. Грис Д. Наука программирования. – М.: Мир, 1984</li> <li>8. Турский В. Методология программирования. М.: Мир, 1981.</li> <li>9. Жуков Д.Ю. Методы верификации программ. 2001.</li> <li>10. Жуков Д.Ю. Методы верификации программ. 2001.</li> <li>11. Орлов С. Технологии разработки программного обеспечения. Учебник- СПб.: Питер, 2002.- 464с.</li> <li>12. Pocatilu P. Costurile testării software. Editura ASE, București, 2004.</li> </ol>
------------	---

Suplimentare	<ol style="list-style-type: none"> <li>1. Хэзфилд Р. Кирби Л. и др. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. К.: Издательство “ДиаСофт”, 2001. – 736 с.</li> <li>2. Andrew Hunt, David Thomas. <i>Pragmatic Unit Testing in C#</i>.</li> <li>3. Jeff Tian. <i>Software Quality Engineering</i>. John Wiley &amp; Sons, 2005</li> <li>4. Hung Q. Nguyen, <i>Testing Applications on the Web</i>, John Wiley &amp; Sons, 2001.</li> </ol>
--------------	--

### 9. Evaluare

Curentă		Proiect de an	Examen final
Atestarea 1	Atestarea 2		
30%	30%		40%
Standard minim de performanță			
<p>Prezența și activitatea la prelegeri și lucrări de laborator;            Obținerea notei minime de „5” la fiecare dintre atestări și lucrări de laborator;            Obținerea notei minime de „5” la proiectul de an;            Demonstrarea în lucrarea de examinare finală a cunoașterii condițiilor de aplicare a procedeelor de modelare constructivă.</p>			