

**FIȘA DISCIPLINEI**  
**Programare orientată pe obiecte**  
 UP.01.S.02.O.13.16  
 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 / licențiat în Informatică

**2. Date despre disciplină**

2.1	Denumirea disciplinei	<b>Programare orientată pe obiecte</b>									
2.2	Titularul activităților de curs	Prof. univ. dr. Tudor BĂLĂNESCU									
2.3	Titularul activităților de laborator	Lect. univ. dr. Ionuț DINCĂ									
2.4	Anul de studii	1	2.5	Semestrul	2	2.6	Tipul de evaluare	Examen	2.7	Regimul disciplinei	O

**3. Timpul total estimat**

3.1	Număr de ore pe săptămână	4	3.2	din care curs	2	3.3	laborator	2
3.4	Total ore din planul de inv.	56	3.5	din care curs	28	3.6	laborator	28
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								38
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								22
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								22
Tutoriat								8
Examinări								4
Alte activități .....								
3.7	Total ore studiu individual			94				
3.8	Total ore pe semestru			150				
3.9	Număr de credite			6				

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

4.1	De curriculum	-
4.2	De competențe	capacitate de sinteză

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

5.1	De desfășurare a cursului	Sală dotată cu videoproiector și ecran.
5.2	De desfășurare a seminarului/laboratorului	Laboratorul disciplinei (S121): calculatoare, rețea, internet, videoproiector, software specific disciplinei: Visual Studio 2012, .NET framework, Java.

**6. Competențe specifice acumulate**

Competențe profesionale	<ul style="list-style-type: none"> <li>• Aplicarea conceptelor, metodelor și principiilor generale de proiectare a unui sistem software folosind metodologia programării pe obiecte.</li> <li>• Utilizarea șablonelor de proiectare în vederea obținerii unui sistem software extensibil, flexibil și robust cu aplicații în diverse limbaje de programare.</li> <li>• Înțelegerea și aplicarea principiilor fundamentale de proiectare orientată pe obiecte</li> </ul>
Competențe transversale	<ul style="list-style-type: none"> <li>• Dezvoltarea capacității de a se integra și de a lucra în echipă.</li> <li>• Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor și de adaptare la cerințele unei societăți dinamice.</li> <li>• Dezvoltarea capacității organizatorice și a autonomiei în derularea de activități în cadrul anumitor proiecte.</li> </ul>

**7. Obiectivele disciplinei**

7.1	Obiectivul general al disciplinei	Disciplina are ca obiectiv general însușirea de către studenți a cunoștințelor de bază privind programarea orientată pe obiecte.
7.2	Obiectivele specifice	<p><i>Obiective cognitive:</i> dezvoltarea unui cadru unitar de programare bazată pe obiecte, cu raportări la câteva din cele mai utilizate limbaje de programare orientată pe obiecte, precum C++, Java, C#, Eiffel; familiarizarea cu pachetele software destinate verificării automate a programelor.</p> <p><i>Obiective procedurale:</i> oferă studenților care intenționează să se implice după absolvire în activitatea companiilor software prilejul de a-și însuși conceptele fundamentale și tehnicile</p>

	de programare orientate spre obiect. <i>Obiective atitudinale:</i> Rigurozitate în implementarea de aplicații orientate pe obiecte.		
<b>8. Conținuturi</b>			
<b>8.1. Curs</b>		Metode de predare	Observații Resurse folosite
1	Concepte fundamentale ale programării orientate pe obiecte. Obiecte și clase, ierarhii de clase, relație de subtip, verificarea statică a tipurilor. Metode și mesaje, multimetode. Moștenire simplă, moștenire multiplă. Transmiterea mesajelor. Legare statică (static binding) și legare dinamică (dynamic, late binding). Coerența stărilor, covarianța metodelor. (4 ore)	Prelegere Dezbatere	Calculator Videoproiector
2	Programare în C++. Încapsularea datelor în C++: Tipuri class. Constructori și destructori. Funcții și clase prietene. Supradefinirea operatorilor: supradefinire prin funcții membre, supradefinire prin funcții prietene. Clasele istream, ostream, iostream: supraîncărcarea operatorilor << și >>. Conversii de tip definite de programator. (4 ore)		
3	Clase derivate: ierarhii de tipuri; downcast și upcast. Redefinirea funcțiilor membre; metode virtuale. Polimorfism de ierarhie. Polimorfism generic. Clase virtuale, Funcții virtuale. Clase abstracte. Operații de intrare-ieșire în C++. (4 ore)		
4	Programare în Java. Tratarea excepțiilor. Interfețe grafice. Event Delegation Model. Model View Controller. Observer-Observable. (4 ore)		
5	Programare paralelă și concurentă: fire de executare. Serializarea obiectelor. Invocarea la distanță a metodelor (RMI). Programarea în rețea. Applet-uri și servlet-uri. (4 ore)		
6	Programare în C#. Platforma Microsoft.net. JIT compiling versus interpretare și compilare. Clase, ierarhii de clase, supraîncărcarea operatorilor. Tipuri valoare, tipuri referință. Tratarea excepțiilor. Implementarea interfetelor. Attribute, spații de nume. (4 ore)		
7	Delegare, tratarea evenimentelor. Proprietăți, indexuri, attribute. Fire de executare. Meta date, reflectare (4 ore)		
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. T. Bălănescu, Ș. Mocanu: Interfețe grafice în Java, Editura Fundației România de Măine, 2005.</li> <li>2. F. M. Boian, R. F. Boian: Tehnologii fundamentale Java pentru aplicații Web, Editura Albastră, 2004.</li> <li>3. D. Gries, P. Gries: Multimedia Introduction to Programming Using Java, Springer, 2005.</li> <li>4. Ș. Tănasă, C. Olaru, Ș. Andrei: Java, de la 0 la expert, Polirom, 2003.</li> <li>5. Herbert Schildt: C++, manual complet. Teora, 1997.</li> <li>6. Octavian Catrina, Iuliana Cojocaru: Turbo C++, Teora, 1993.</li> <li>7. John Hunt: Java and Object Orientation- an introduction, Springer, 1998.</li> <li>8. Meyer, B.: Object Oriented Software Construction, Prentice Hall, 1997, second edition.</li> <li>9. Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon D. Reid, Morgan Skinner, Eric White: Beginning Visual C Sharp 2008, Wiley Publishing, Inc, 2008.</li> <li>10. T. Bălănescu, Ș. Gavrilă, H. Georgescu, M. Gheorghe, L. Sofonea, I. Văduva: "Programarea în limbajele Pascal și Turbo Pascal", Editura Tehnică, vol. 1 și vol. 2, 1992.</li> <li>11. T. Bălănescu: Corectitudinea algoritmilor, Editura Tehnică, 1995.</li> </ol>			
<b>8.2. Aplicații – Seminar / Laborator</b>		Metode de predare	Observații Resurse folosite
1	Liste eterogene. Programe extensibile. Implementarea unui sistem de rezervare camere hotel. (4 ore)	Studii de caz, Proiecte	Calculatorul, videoproiectorul , platforme software: Java, .NET Framework, IDE: Eclipse, Visual Studio
2	Clase de matrici. Clase de numere complexe. Supraîncărcarea operatorilor. (4 ore)		
3	Tratarea excepțiilor Interfețe grafice și Event Delegation Model. (4 ore)		
4	Interfață grafică pentru un calculator de buzunar, bazat pe interfețele Action Listener, WindowListener și utilizând șablonul Model View Controller. (4 ore)		
5	Utilizarea șablonului Observer-Observable pentru implementarea unui ceas analogic, pentru implementarea unor interfețe. Programare cu fire de executare în Java: procese producător-consumator. Programare în rețea pentru implementarea unei aplicații client-server de verificarea a unor date de înregistrare (sign-in). (4 ore)		
6	Interfețe grafice (GUI) C# pentru jocuri (ghicirea unui număr). (4 ore)		
7	Implementarea unor protocoale de sincronizare între procese producător-consumator. (4 ore)		
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. T. Bălănescu, Ș. Mocanu: <i>Interfețe grafice în Java</i>, Editura Fundației România de Măine, 2005.</li> </ol>			

2. F. M. Boian, R. F. Boian: *Tehnologii fundamentale Java pentru aplicații Web*, Editura Albastră, 2004.
3. D. Gries, P. Gries: *Multimedia Introduction to Programming Using Java*, Springer, 2005.
4. Ș. Tănasă, C. Olaru, Ș. Andrei: *Java, de la 0 la expert*, Polirom, 2003.
5. Herbert Schildt: *C++, manual complet*. Teora, 1997.
6. Octavian Catrina, Iuliana Cojocaru: *Turbo C++*, Teora, 1993.
7. John Hunt: *Java and Object Orientation- an introduction*, Springer, 1998.
8. Meyer, B.: *Object Oriented Software Construction*, Prentice Hall, 1997, second edition.
9. Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon D. Reid, Morgan Skinner, Eric White: *Beginning Visual C Sharp 2008*, Wiley Publishing, Inc, 2008.

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

Competențele dobândite la disciplină permit absolvenților să utilizeze în practică principiile programării orientate pe obiecte în domenii ca proiectarea și programarea sistemelor software.

**3. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Test de evaluare finală	Probă scrisă	50%
10.5 Seminar/ Laborator	Activitate laborator Lucrări periodice de control Tema de casă	Înregistrare activitate Verificare rezultate Verificare soluții	30% 10% 10%
10.6 Standard minim de performanță	- nota 5 la prezență, activitate seminar, lucrări periodice de control și tema de casă; - nota 5 la proba scrisă finală.		

Data completării  
20.09.2020

Titular de curs  
Prof. univ. dr. Tudor BĂLĂNESCU

Titular de laborator  
Lect. univ. dr. Ionuț DINCĂ

Data avizării în Departament

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

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