

FIȘA DISCIPLINEI
Tehnici avansate de programare orientată obiect

1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918”
1.2. Facultatea	de Științe Exacte și Inginerești
1.3. Departamentul	de Științe Exacte și Inginerești
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii	Programare avansata si baze de date

2. Date despre disciplină

2.1. Denumirea disciplinei	<i>Tehnici avansate de programare orientată obiect</i>		2.2. Cod disciplină	MI102			
2.3. Titularul activității de curs	Rotar Corina						
2.4. Titularul activității de seminar	Rotar Corina						
2.5. Anul de studiu	I	2.6. Semestrul	I	2.7. Tipul de evaluare (E/C/VP)	E	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	O

3. Timpul total estimat

3.1. Numar ore pe saptamana	2	din care: 3.2. curs	2	3.3. seminar/laborator	2
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					12
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					40
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					50
Tutoriat					-
Examinări					12
Alte activități					-

3.7 Total ore studiu individual	114
3.8 Total ore din planul de învățământ	48
3.9 Total ore pe semestru	162
3.10 Numărul de credite	6

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	1. Programare orientata obiect 2. Analiza si proiectarea programelor (recomandata)
4.2. de competențe	-

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<i>Sala dotata cu videoproiector/tabla</i>
--------------------------------	--

5.2. de desfășurarea a seminarului/laboratorului

Laboratoare – calculatoare dotate cu: Visual Studio 2010, acces Internet.

6. Competențe specifice acumulate

Competențe profesionale	<p>Competențe cognitive Capacitatea de:</p> <ul style="list-style-type: none">• analiza si proiectare orientata obiect• identifica tipul de sablon de proiectare in urma studiului de caz• opera cu instrumentele UML in scopul proiectarii software• implementa intr-o maniera eficienta orientata obiect proiectul software <p>Competențe tehnice/profesionale Studentul trebuie sa dovedeasca faptul ca a dobandit un nivel acceptabil de cunostinte si intelegere a subiectului si ca a dobandit abilitatile corecte de analiza, proiectare si implementare bazate pe sabloane de proiectare si pe paradigmele generale ale programarii orientate-obiect.</p> <ul style="list-style-type: none">• consolidarea deprinderilor de analiza si proiectare a sistemelor soft prin invatarea unui proces liniar simplu de analiza si proiectare orientata obiect• abilitatea de a opera cu paradigma programarii generice• studierea unor elemente de baza ale limbajului de modelare UML• deprinderea de a utiliza sabloane de proiectare• aplicarea unor sabloane standard privind alocarea responsabilitatilor pentru obiecte• studierea unor metode de proiectare care s-au aplicat in procesul de construire a unor biblioteci <p>Competențe afective Capacitatea de a cunoaște și înțelege:</p> <ul style="list-style-type: none">• conceptele avansate ale paradigmei orientate obiect ;• principalele tipuri de sabloane de proiectare;• principiile proiectarii orientate obiect <p>Capacitatea de a colabora cu membrii unei echipe de proiectare-programare.</p>
Competențe transversale	Nu e cazul

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

7.1 Obiectivul general al disciplinei	<p>Cursul isi propune:</p> <ul style="list-style-type: none">• studierea capitolelor avansate ale paradigmei orientate obiect• consolidarea deprinderilor de analiza si proiectare a sistemelor software prin <i>analiza si proiectare orientata obiect</i>• abilitatea de a opera cu paradigma programarii generice• studierea unor elemente de baza ale limbajului de modelare UML• deprinderea de a utiliza <i>sabloane de proiectare</i>
7.2 Obiectivele specifice	<ul style="list-style-type: none">• aplicarea unor sabloane standard privind alocarea

	responsabilitatilor pentru obiecte <ul style="list-style-type: none"> • studierea unor metode de proiectare care s-au aplicat in procesul de construire a unor biblioteci
--	--

8. Conținuturi

8.1 Curs	Metode de predare	Observații Nr ore alocate
Tehnologia orientata obiect. Concepte fundamentale OO (recapitulare), Clase abstracte.	Prezentare ppt	2
Interfete si Tratarea exceptiilor	Prezentare ppt	2
Functii template, Clase Template, Clase parametrizate.	Prezentare ppt	2
Programarea generica. Iterators and Algorithms	Prezentare ppt	2
Introducere în analiza și proiectarea orientată obiect. Introducere în limbajul de modelare UML	Prezentare ppt	2
UML: Class Diagram, Sequence Diagram, Collaboration Diagram, State Diagram, Activity Diagram, Exerciții	Prezentare ppt	2
GUI: - Ferestre si componente grafice, Containere, elemente atomice ?	Prezentare ppt	2
Introducere in Design Patterns:Definitii, Creational Design Patterns, Structural Design Patterns, Behavioural Design Patterns	Prezentare ppt	2
Creational Design Patterns> Singleton, Factory, Factory Method, Abstract Factory, Builder, Prototype, Object Pool	Prezentare ppt	2
Behavioural Design Patterns> Sablonul de proiectare Model-View-Controller. Sabloane de proiectare > Iterator, Memento, Observer, Strategy, Template Method, ,Visitor.	Prezentare ppt	2
Structural Design Pattern. Sabloane de proiectare> <u>Adapter</u> , <u>Bridge</u> , <u>Composite</u> , <u>Decorator</u> , <u>Flyweight</u> , <u>Proxy</u> .	Prezentare ppt	2
Concepte avansate Design Patterns :Exemple, Studiu de caz.	Prezentare ppt	2
Implementare > Clase/metode/atribute (implementare conform diagramelor de proiectare), Pachete/ componente	Prezentare ppt	2
Paradigma orientata obiect si bazele de date	Prezentare ppt	2
8.2 Bibliografie <ol style="list-style-type: none"> 1. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns - Elements of Reusable Object-Oriented Software, Massachusetts, Addison-Wesley, 1994. (traducere Teora) 2. Bruce Eckel, Thinking in C++ 3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 4. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2000. 5. Austern 1999] Matthew H. Austern, <i>Generic Programming and the STL: Using and Extending the C++ Standard Template Library</i>, Addison-Wesley, 1999. 6. Bruce Eckel, Thinking in C++, manual online. 7. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 		

8. H. Schildt: C++ manual complet, Teora, 2000.
9. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice.
10. Meyer, B., Object-Oriented Software Construction, Prentice-Hall, 1988.

Seminar-laborator	Nr. ore alocate	
<i>Clase abstracte. Probleme</i>	2	
<i>Interfete si Tratarea exceptiilor. Probleme</i>	2	
<i>Functii template,Clase Template. Probleme</i>	2	
<i>Clase parametrizate. Probleme</i>	2	
<i>Probleme: Programarea generica.</i>	2	
<i>Probleme: Programarea generica. continuare</i>	2	
<i>TAD Container (C++)</i>	2	
<i>Sabloane de proiectare</i> Introduction,Factory Method, Abstract Factory	2	
<i>Sabloane de proiectare</i> Builder, Prototype, Singleton	2	
<i>Sabloane de proiectare</i> Adapter,Bridge, Composite, Decorator	2	
<i>Sabloane de proiectare</i> Facade, Flyweight, Proxy	2	
<i>Sabloane de proiectare</i> Chain of Responsibility, Command, Interpreter, Iterator, Mediator	2	
<i>Sabloane de proiectare:</i> Memento, Observer, State, Strategy, Template Method, Visitor	2	
<i>Predarea proiectelor</i>	2	

Bibliografie

1. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns - Elements of Reusable Object-Oriented Software, Massachusetts, Addison-Wesley, 1994. (traducere Teora)
2. Bruce Eckel, Thinking in C++
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2000.
5. Austern 1999] Matthew H. Austern, *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*, Addison-Wesley, 1999.
6. Bruce Eckel, Thinking in C++, manual online.
7. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997.
8. H. Schildt: C++ manual complet, Teora, 2000.
9. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice.
10. Meyer, B., Object-Oriented Software Construction, Prentice-Hall, 1988.

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

-

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Examen oral</i>	50%
	-	-	-

10.5 Seminar/laborator	<i>Verificare pe parcurs</i>	<i>Proiect individual</i>	50%
	-		-

10.6 Standard minim de performanță:

Implementarea și documentarea de unități de program în limbaje de programare de nivel înalt și folosirea eficientă a mediilor de programare

Observatii: Recuperarea laboratoarelor se poate face in regim de consultații in timpul semestrului. De asemenea, in cazuri bine motivate, recuperarea orelor de laborator se mai poate face prin prezentarea de către student a portofoliului complet de lucrari practice - in ultima saptamana din semestrul II, in orele de consultații ale cadrului didactic titular.

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

.....

.....

.....

Data avizării în catedră

Semnătura director de departament

.....

.....