

FIȘA DISCIPLINEI
Anul universitar 2016-2017

Decan,

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică „Gheorghe Asachi” din Iași
1.2 Facultatea	Automatica și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii ¹	Licenta
1.6 Programul de studii	Calculatoare

2. Date despre disciplină

2.1 Denumirea disciplinei	Sisteme distribuite						
2.2 Titularul activităților de curs	Mihai Horia Zaharia						
2.3 Titularul activităților de aplicații	Mihai Horia Zaharia, Adrian Alexandrescu, Gavrița Ionuț						
2.4 Anul de studii ²	3	2.5 Semestrul ³	6	2.6 Tipul de evaluare ⁴	Ex	2.7 Tipul disciplinei ⁵	DID

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

3.1 Număr de ore pe săptămână	4	din care 3.2 curs	2	3.3a sem.	-	3.3b laborator	2	3.3c proiect	-
3.4 Total ore din planul de învățământ ⁶	56	din care 3.5 curs	28	3.6a sem.	-	3.6b laborator	28	3.6c proiect	-
Distribuția fondului de timp ⁷									Nr. ore
Studiul după manual, suport de curs, bibliografie și notițe									16
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren									5
Pregătire seminarii/laboratoare/proiecte, teme, referate și portofolii									10
Tutoriat ⁸									5
Examinări ⁹									4
Alte activități:									-
3.7 Total ore studiu individual ¹⁰	40								
3.8 Total ore pe semestru ¹¹	96								
3.9 Numărul de credite	4								

4. Precondiții (acolo unde este cazul)

4.1 de curriculum ¹²	• CTI.DID.205, CTI.DID.206
4.2 de competențe	• CTI.DID.201

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului ¹³	• Videoprojector + tabla
5.2 de desfășurare a seminarului / laboratorului / proiectului ¹⁴	• Computer și pachetele software coform programei

6. Competențele specifice acumulate¹⁵

Număr de credite alocate disciplinei ¹⁶ :			4	Repartizare credite pe competențe ¹⁷
Competențe profesionale	CP1	Operarea cu fundamente științifice, ingineresti și ale informaticii		0.4
	CP2	Proiectarea componentelor hardware, software și de comunicații		0.8
	CP3	Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor		1.2
	CP4	Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații		0.4
	CP5	Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații		1
	CP6	Proiectarea sistemelor inteligente		-
Competențe transversale	CT1	Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei		0.2
	CT2	Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate		-
	CT3	Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională		-

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Prezentarea principalelor direcții specifice calcului distribuit precum și ale aplicațiilor acestuia. Obținerea de abilități în proiectarea și implementarea unor astfel de sisteme.
7.2 Obiective specifice	<ul style="list-style-type: none"> •

8. Conținuturi

8.1 Curs ¹⁸	Metode de predare ¹⁹	Observații
<p>Curs 1 2h Evoluția principalelor clase de arhitecturi. Modelul cu acces uniform la memorie Modelul cu acces neuniform la memorie Modelul cu acces la memorie cache Modelul sistemelor slab cuplate Modelul Li Modelul Enslow Definiții (sisteme distribuite, calcul distribuit) Avantaje și Dezavantaje Comunicațiile</p> <p>Curs 2 2h Modele specifice Modelul unui sistem oarecare sistemului distribuit ca sistem cu tranziții Când se utilizează modelarea SD Modele de aplicații distribuite Alocarea resurselor (singular și multiple) Problema excluziunii mutuale <ul style="list-style-type: none"> - Abordare centralizată - Abordare distribuite </p> <p>Curs 3 2h Sisteme distribuite de fișiere Introducere Caracteristici Tehnici de cache Atomicitate și ordinea evenimentelor Deziderate specifice asigurării consistenței Modele de memorie specifice arhitecturilor multiprocesor scalabile cu memorie comună distribuită <ul style="list-style-type: none"> - Modelul consistenței procesorului - Consistență relaxată ("Release consistency") Problema coerenței cache-ului <ul style="list-style-type: none"> - Protocol cu scriere imediată ("Write-Through") - Protocol cu scriere întârziată ("Write-Back") - Protocol cu scriere unică ("Write Once") Servicii cu stare versus servicii fără stare Replicarea fișierelor Sistemul NFS(Network File Sistem) primele abordări <ul style="list-style-type: none"> - Problema inconsistenței cache în NFS NFS V4 SNF Sprite Network File system</p> <p>Curs 4 2h Sistemul NFS(Network File Sistem) Sistemul Decorum Sistemul Plan 9 Sistemul AFS Operații pe fișiere și consistența semantică</p>	<p>Expunere combinată cu prelegere dar și comentarii suplimentare la tablă cu suport pe videoprojector Discuții suplimentare la cererea studentului în afara orelor aferente curs sau laborator</p>	

<p>Curs 5 2h Echilibrarea Incarcarii Problematika Algoritmul generic al unei echilibrări a încărcării Echilibrarea statică a încărcării Cel mai apropiat vecin Round and Robin Asincron Round and Robin Global Metode de planificare dinamică Abordarea centralizată Abordarea descentralizată (rr) Etapile specifice realizării echilibrării dinamice (SID, DASUD) Echilibrarea prin predicția încărcărilor</p> <p>Curs 6 2h Memoria virtuală distribuită DSM Memoria distribuita simulata Memorie comuna distribuita realizata software (Virtual shared memory) Protocol de tip "Single Writer" Protocol de tip "Multiple Writers" Modele de consistenta a memoriei pentru memoria comuna distribuita implementata software (SDSM) Eager Release Consistency Lazy Release Consistency Entry Release Consistency Algoritmul LI Sincronizare și sisteme tranzacționale Sicronizarea în sistemele distribuite\ Ceasuri logice Abordarea Lamport Ceasuri fizice Algoritmul Cristian Algoritmul Berkeley Algoritm descentralizat Tranzacții atomice Model tranzactional Primitivele tranzacțiilor Proprietățile tranzacțiilor Folosirea spațiului privat de lucru Metoda jurnalului posibilei execuții ("writeahead log") Protocol cu execuția în două etape Controlul concurenței în sistemele tranzacționale</p> <p>Curs 7 2h Cluster De ce calcul paralel Paradoxul calculului paralel Justificare apariției conceptului de cluster Cluster management software Atributele folosite la clasificarea cluster-elor Suportul pentru disponibilitate tehnici pentru asigurarea disponibilității, folosite la cluster-e. - Redundanța izolată - Preluarea (failover) - Scheme de refacere (recovery schemes) - Imagine unică de system Arhitectura unui cluster Relațiile dintre componentele software/hardware, la un nod dintr-un cluster</p>		
---	--	--

<p>Gestiunea job-urilor în cluster-e Tratarea eterogeneității Clustere Beowulf Cluster IBM</p> <ul style="list-style-type: none"> - WebSphere Performance Advisor: <p>Proiectul NOW de la Universitatea Berkeley Windows server Windows server Advanced Server</p> <p>Curs 8 2h GRID Descrierea Arhitecturii Grid Descrierea nivelelor arhitecturii Grid</p> <ul style="list-style-type: none"> - Nivelul Fabric (Fabric) - Nivelul Conectivitate (Connectivity) - Nivelul Resurse (Resource) - Nivelul Coordonare (Collective) <p>Clasificări ale Tehnologiilor Grid</p> <ul style="list-style-type: none"> - Cluster Grid - Campus Grid - Global Grid <p>Problema programarii Tehnologii utilizate</p> <ul style="list-style-type: none"> - MPICH-G2: - CoG Kits, GridPort: - GDMP, Data Grid Tools, SRB: - Condor-G: - Legion: - Cactus: <p>Calcul de mare incarcare si Condor Abordari OOP Portaluri Organizatie virtuala</p> <ul style="list-style-type: none"> - Structura unei organizatii virtuale - Virtual vs. Organic structure - VO-urile si infrastructura - Relatii de incredere - Delegarea simpla - Delegarea specializata sau restrictionata - Separararea responsabilitatilor <p>Cursul 9 2h SOA Ce este sip e c se bazeaza Fundamentele SOA Abstractizarea serviciului Agentul executabilii si stratul de mapare Gestionarul (Handler) serviciilor Solicitant si furnizor Tranziția la SOA Implementari SOA</p> <ul style="list-style-type: none"> - Obiecte distribuite CORBA, J2EE, COM/DCOM. - Middleware orientat spre mesaje (MOM) - WebSphere MQ, Tibco Rendezvous. - Monitoare CICS, IMS, Encinia, Tuxedo. - Platforme B2B precum ebXML, RosettaNet. - Servicii Web - WebSphere MQ <p>Probleme pana la aparitia norilor</p>		
--	--	--

<p>Solutii</p> <p>IaaS - Infrastructure as a Service</p> <p>Strategii posibile de rezolvare</p> <p>Modelul Traditional (on-premise) versus IaaS</p> <p>Ce este noruletul</p> <p>Caracteristicile de baza</p> <p>Componente CC</p> <p>Tipuri de Nori</p> <p>Software as a Service (SaaS)</p> <p>Tipuri de software care conduc la modelul SaaS</p> <p>Aplicatii SaaS si exemple</p> <p>Caracteristicile cheie ale SaaS</p> <p>Platform as a Service (PaaS)</p> <p>Infrastructure as a Service (IaaS or HaaS)</p> <p>Disponibilitate si Fiabilitate in IaaS</p> <p>Gestiune si Interoperabilitate in IaaS</p> <p>Performante si optimizari in IaaS</p> <p>Accesibilitate si portabilitate in IaaS</p> <p>Tipuri de virtualizare</p> <p>Virtualizare vs Paravirtualizare</p> <p>Tehnici de virtualizare in IaaS</p> <p>Furnizori de servicii Cloud (IaaS si/sau PaaS si/sau SaaS)</p> <p>Piramida norului</p> <p>Storage as a Service</p> <p>Database as a Service (DaaS)</p> <p>Information as a Service</p> <p>Communication as a Service (CaaS)</p> <p>Identity as a Service</p> <p>Monitoring as a Service (MaaS)</p> <p>Process as a Service Integration as a Service</p> <p>MaaS si TaaS</p> <p>Backup as a Service</p> <p>Orice ca serviciu (XaaS)</p> <p>Serviciile Amazon</p> <p>Windows Azure</p> <p>Cursul 10 2h</p> <p>Shared Information systems</p> <p>Definitii si clasificari</p> <p>Model Ierarhic</p> <p>Modelul Retea</p> <p>Modelul Relational</p> <p>Hybrid Object/Relational Model prin compunerea in stiva a celor clasice</p> <p>PURE OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEMS - OODBMS</p> <p>Hybrid Object/Relational Database Management Systems (HORDBMS)</p> <p>Sisteme de baze de date distribuite</p> <p>Arhitectura DRDA a IBM</p> <p>Sistemele de baze de date multiple</p> <ul style="list-style-type: none"> - Abordarea cu schema globala - Abordarea cu schema multipla <p>Sisteme de baze de date distribuite federative</p> <p>Sistemele (de productie) expert</p> <p>Strategii de refacere locale sau distribuite</p> <p>Tipuri de jurnalizare utilizate</p> <p>Proprietati DDB</p> <p>Baze de date NoSQL</p> <p>Baze de date "cloud"</p> <p>Xeround</p>		
---	--	--

<p>MongoDB RiaK Hypertable Cassandra Curs 11 2h Peer to Peer P2P Definitii si clasificari Utilizarea descentralizata a resurselor Autoorganizarea distribuita Aspecte privind dezvoltarea P2P Sisteme P2P nestructurate Sisteme P2P Structurate dht Arii de aplicare a sistemelor P2P</p> <ul style="list-style-type: none"> - Presence Information - Gestiunea documentelor (document management systems – DMS) - Colaborare <p>Resursele gestionate de o aplicatie P2P Gnutella Napster – RIP FreeNet Largimea de banda Stocarea: Utilizarea ciclurilor processor Furt cicluri idle Proiectarea unei aplicatii P2P</p> <ul style="list-style-type: none"> - Probleme specific transferului de date - Securitatea - Arhitectura - Comunicare - Tabela peer - Tabela share <p>Clasificare documentelor Problema clasificarii documentelor Graful de dependinte a datelor Partitionare si comunicare Aglomerarea si maparea Algoritm de tip Manager/worker Manager/Worker versus SPMD Tabele hash Coduri Hash Implementarea</p> <p>Cursul 12 2h Sisteme distribuite tolerante la erori Definitii si context Servere, servicii si relatia lor de dependenta</p> <ul style="list-style-type: none"> - Relatia de dependenta - Clasificarea caderilor - Propagarea ierarhica a erorilor si mascarea acestora <p>Mascarea erorilor prin utilizarea unor grupuri de servere Abordari hardware Arhitectura cu procesoare in tandem Arhitectura VAX cu cluster de procesoare IBM XRF/1 Arhitectura Stratus Arhitectura Sequoia /1</p>		
--	--	--

<p>Sincronizarea membrilor unui grup tolerant la erori Sincronizare relaxat Puncte de salvare a executiei si refacerea Punct local de salvare (checkpoint) Stari stabile Interactiuni cu lumea exterioara Mesajele Refacerea dupa eroare Checkpointing necoordonat Tehnica urmaririi dependentei directe Checkpointing coordonat (blocant si nebloant) Checkpointurile induse de comunicatie Log-based Rollback Recovery Conditia de consistenta pentru a evita orfanii Inregistrare pesimista - Pessimistic Logging Inregistrare optimista - Optimistic Logging Causal Logging Algoritmul Koo-Toueg pentru salvare coordonata Algoritmul Juang-Venkatesan pentru salvari si restaurari asincrone</p> <p>Curs 13 2h</p> <p>Caclul mobil Limitarile dispozitivelor mobile Proprietatile sistemului distribuit mobil Contextul Rețele sensor Diferente între R. Sensor și R. Clasice Modele de mobilitate Protocoale de rutare ad-hoc Protocoale proactive <ul style="list-style-type: none"> - Destination-Sequenced (DSDV) - Clusterhead Gateway Switch Routing (CGSR) Protocoale reactive <ul style="list-style-type: none"> - Protocolul Dynamic Source Routing (DSR) - Ad Hoc On-Demand Distance Vector(AODV) - Algoritmul Temporally Ordered Routing (TORA) - Associativity-Based Routing (ABR) - Signal Stability-Based Adaptive Routing (SSR) Protocoale hibride Protocoale bazate pe pozitie Protocoale care țin cont de consumul de energie Rutarea în rețele deconectate Rutarea bazată pe agenți în rețele ad-hoc Broadcasting și multicasting deficitar Probability Based Methods Area Based Methods Neighbor Knowledge Methods Problema “Consensului” Probleme de autostabilizare Problema partitionarii</p> <p>Curs 14 2h</p> <p>Inteligența Artificială Distribuția – DAI Cotext și definiții Caracteristici agenți Clasificări agenți Agenți mobili <ul style="list-style-type: none"> - Mediul de lucru - Ciclu de viață </p>		
--	--	--

<ul style="list-style-type: none"> - Model bazat pe actiuni - Model computational - Model securitate <p>Comunicare in SMA Comunicare indirecta Comunicare directa Nivelele comunicarii Limbaje specific pentru comunicare (ACL, KQML, FIPA) Limbaje pentru continut (KIF, Prolog, Clips, SQL, FIPA-SL, FIPA-KIF) Protocoloale de interactiune</p> <ul style="list-style-type: none"> - Automate finite - Conversatii in KQML - Retele Petri <p>Modele arhitecturale de agenti</p> <ul style="list-style-type: none"> - Structura conceptuala a agentilor - Arhitecturi de agenti cognitive <ul style="list-style-type: none"> o Agenti inteligenti(BDI full) o Agenti emotionali - Arhitecturi de agenti reactivi <ul style="list-style-type: none"> o De subsumare 		
<p>Bibliografie curs:</p> <ol style="list-style-type: none"> 1. Muhl, G., Fiege L, Pietzuch P., „Distributed event based systems”, Springer-Verlag Berlin Heidelberg, Germania, 2006 2. Păstravanu O. “Sisteme cu evenimente discrete – Tehnici calitative bazate pe formalismul rețelelor Petri”, Ed. Matrix Rom, București, 1997. 3. Pfister G., “In search of clusters”, Prentice Hall, 1995. 4. Raynal, M. , “Distributed Algorithms and Protocols”, John Wiley & Sons Ltd, 1988. 5. Raynal, M., Helary, J.M., “Synchronisation and Control of Distributed Systems and Programs ”, John Wiley & Sons Ltd, 1988. 6. Sloam, M., Kramer, J. "Distributed Systems and Computer Networks ", Prentice Hall 1987. 7. Tanenbaum A. S. “Rețele de calculatoare”, Computer Press Agora, 1997. 8. Tanenbaum A. S., „Modern Operating Systems”, Prentice Hall, USA, 1992. 9. Tanenbaum A. S., "Distributed Systems", Prentice Hall 1993. 10. Zaharia M. H., “Fluxuri de producție în medii distribuite”, Ed. F.C. “Renașterea Română”, Iași, 2000. 11. M. H. Zaharia, Sisteme Paralele și distribuite (“Parallel and distributed systems”), Ed. “Gh. Asachi”, 2003, ISBN 973-621-056-1, 187 pg format academic 12. Mihai Horia Zaharia, Cătălin Horghidan, Claudiu Mihăilă, Sisteme Distribuite: îndrumar de laborator, Politehniun, 2007, ISBN 978-973-621-175-1, 225 format academic 13. www.cs.iit.edu/~cs550/lectures/10_distributed_shared_memory.ppt 14. Distributed Shared Memory 15. http://courses.washington.edu/css434/slides/DSM.ppt 16. ftp://ftp.cs.wisc.edu/pub/techreports/1989/TR825.pdf 17. http://web.cecs.pdx.edu/~howe/cs410/ 18. http://www.cs.utsa.edu/~korkmaz/teaching/cs7123/ 19. http://www.nordugrid.org/ 20. https://www.nsf.gov/div/index.jsp?div=ACI 21. http://www.cloudbus.org/course/ 22. http://toolkit.globus.org/toolkit/ 23. http://www.slac.stanford.edu/econf/C0303241/ 24. http://grid.itim-cj.ro/vo.html 25. http://profs.info.uaic.ro/~alaiba/mw/index.php?title=Sistemul_de_calcul_pe_grid_Globus_Toolkit 26. https://www.globus.org/alliance/publications/papers/anatomy.pdf 27. Binildas C. A. (2008) Service Oriented Java Business Integration Enterprise Service Bus integration solutions for Java developers 28. Packt Publishing Ltd, Birmingham, UK 29. http://thor.info.uaic.ro/~adria/teach/courses/pcd/ 		

30. http://cks.univnt.ro/uploads/cks_2013_articles/index.php?dir=4_IT_in_Social_Sciences%2F&download=cks_2013_it_003.pdf
31. <http://banking.incloud.ro/wp-content/uploads/2013/02/GARTNER-Felix-enescu-v02.ppsx>
32. <http://download.microsoft.com/download/B/5/F/B5F597E5-CBE7-420B-B8D8-5BC9EDC0C575/Introducing%20Windows%20Azure.ppt>
33. <http://web.info.uvt.ro/~petcu/distrib.htm>
34. Ozsu, M. Tamer, and P. Valduriez. *Principles of Distributed Database Systems*, Englewood Cliffs, NJ: Prentice Hall, 1991.
35. Angelo R. Bobak, *Distributed and Multi-Database Systems*, Boston, US, Artech House, 1995
36. <http://ctrl-d.ro/development/resurse-development/introducere-in-tematica-bazelor-de-date-in-cloud/>
37. http://www.todaysoftmag.ro/article/ro/10/Bazele_de_date_NoSQL_-_o_analiza_comparativa_304
38. Quang Hieu Vu, Mihai Lupu, Beng Chin Ooi, *Peer-to-Peer Computing Principles and Applications*, Springer Heidelberg, 2010
39. Ralf Steinmetz Klaus Wehrle, *Peer-to-Peer Systems and Applications*, Springer Heidelberg, 2005
40. Dmitry Korzun, Andrei Gurtov, *Structured Peer-to-Peer Systems - Fundamentals of Hierarchical Organization, Routing, Scaling, and Security*, Springer Heidelberg, 2013
41. Wolfgang Effelsberg, Ralf Steinmetz, Thorsten Strufe, *Benchmarking Peer-to-Peer Systems Understanding Quality of Service in Large-Scale Distributed Systems*, Springer Heidelberg, 2013
42. Vikas Gupta, Avnish Dass, Harpreet Singh Matharu, Ankur Verma, Yashraj Chauhan, *Peer-to-Peer Application Development - Cracking the Code*, Hungry Minds, Inc., New York, 2002
43. **A.D. Kshemkalyani, M. Singhal, [Distributed Computing: Principles, Algorithms, and Systems](#), ISBN: 9780521189842, paperback edition, Cambridge University Press, March 2011**

8.2a Seminar	Metode de predare ²⁰	Observații
-	-	-
8.2b Laborator	Metode de predare ²¹	Observații
Laborator 1 Task 1. Obisnuirea cu mediul de dezvoltare (5 min) Task 2. Analiza mediului pentru descriere rápida a GUI (tip Borland) 10 min Task 3. Crearea unui buton si tratarea evenimentului generat de acesta (5 min) codul din spate este automat dezvoltat de mediu Task 4 Crearea unei cutii de dialog (citire scriere in ea) tratarea evenimentelor Task 5 Combinarea cutiei cu doua butoane asociate cu numerele 1 si 2 si un buton de calcul. Se vor trata (timp 20 minute) - Evenimentele de la fereastra principala - Evenimentele de la elementele incluse in ea - Se va face calcul 1+3 si se va afisa rezultat Task 6. Se va realiza calculatorul cu patru operatii (60 minute) Tema pe acasa: Calulatorul cu patru operatii dar va suporta expresii complexe (polonesa) implementata clasic cu arbori sau cu colectii) Laborator 2 Programarea soket-urilor- Presentare detaleata Crearea unui socket Send / receive BeginReceive Onconnect Callback Creare server Creare client Se testeaza (eventual completeaza) exemplele prezentate Tema: Modificati exemplul in care se realizeaza comunicarea asincrona astfel incat: - serverul sa accepte conexiuni de la mai multi clienti - serverul sa transmita mesaje catre toti clientii	Demonstratie practica execitiu si experiment	none

- cand serverul primeste un mesaj de la un client sa il retransmita catre ceilalti

Labrator 3

MSMQ

3. Cum folosim coada de mesaje în .NET

3.1. Crearea aplicației care transmite mesajul

3.2. Trimiterea unui mesaj simplu

3.3. Trimiterea mesajelor complexe

3.4. Referinta la cozi cu cai directe

3.5. Construirea aplicatiei de receptionare

3.6. Folosirea XmlMessageFormatter

3.7. Verificarea continua (polling) a cozii

3.8. Citirea Asincrona a Mesajelor

3.9. Trimiterea instantelor claselor definite de utilizator in mesaje

3.10. Folosirea XmlMessageFormatter

3.11. Utilizarea BinaryMessageFormatter

3.12. Folosirea ActiveXMessageFormatter

Tema lab:

Construiti doua aplicatii una care sa transmita si una care sa primeasca mesaje. Aplicatia care receptionaza mesajele sa receptioneze mesajele fie prin intermediul unui callback delegate fie prin intermediul evenimentului ReceiveCompleted. Cele doua aplicatii sa vehiculeze date impachetate printr-o clasa Customer care sa contina diferite campuri. Încercați mai întâi sa folosiți XmlMessageFormatter si apoi BinaryMessageFormatter.

Tema acasa

Folosind MSMQ realizati un editor text colaborativ) ce vede-scrie-sterge unul vad toti si vice versa)

Laborator 4

.Net Remoting 1

4. Implementarea claselor Well Known (Server Activated

4.1. Crearea serverului

4.2. Crearea aplicatiei client

4.3. Modul Singleton vs. modul SingleCall

4.4. Implementarea obiectelor Client-Activated

4.5. Crearea serverului

4.6. Crearea clientului

4.7. Configurarea remoting

4.8. Configurarea Serverului

4.9. Configurarea clientului

Tema lab: Realizati MathServer si MathClient prezentata in laborator.

Configurarea sa se realizeze cu ajutorul fisierelor de configurare.

Modificati fisierele de configurare astfel incat clasa SimpleMath sa fie expusa in modul Client Activated.

Tema acasa: Modificati tema de laborator astfel incat serverul sa ofere facilitati de sortare, cautare si stergere pentru orice vector trimis la intrare si sa livreze rezultatul in cati vectori/variabile sunt necesare.

Evident clientul trebuie modificat pentru testare

Laboratorul 5

.NET Remoting 2

5. Obiectele Marshal-by-reference

5.1. Modificarea proprietatilor leasului

5.2. Initializarea Lease-ului

5.3. Reinoirea Lease-urilor

5.4. Sponsorii

5.5. Folosirea Sponsorului pentru a renoi un lease

5.6. Sink-uri si lanturi de Sink-uri

<p>5.7. Procesarea mesajelor in lantul de sink-uri de canal 5.8. Crearea lanturilor de sink-uri de canale 5.9. Sink-urile de formatare 5.10. Sink-urile de canal particularizate 5.11. Sink-urile de transport Tema lab: Identificati de ce nu functioneaza corect exemplul din laborator. Gasiti doua moduri de a rezolva cauza erorii fara a modifica configurarea de lifetime a domeniului de aplicatie al serverului (elementul <lifetime>). Realizati o clasa care implementeaza ISponsor pentru a face exemplul sa functioneze. Tema acasa: Sa se implementeze in retea un joc de ping pong cu sau patru jucatori folosind . Interfata grafica ca mai jos (poate fi facuta inclusiv in mod text). Se vor folosi cunostiinte din acest laborator.</p> <p>Laboratorul 6 Construirea si utilizarea serviciilor Web in .NET</p> <p>6. IIS</p> <p>6.1. Folosirea Code-Behind 6.2. Crearea serviciilor Web cu ajutorul Visual Studio .NET 6.3. Utilizarea serviciului Web 6.4. Stabilirea unei referinte Web 6.5. Folosirea Proxy-ului generat 6.6. Apelarea asincrona a serviciilor Web 6.7. Intoarcerea tipurilor definite de utilizator de la serviciul Web 6.8. Returnarea tipurilor generice 6.9. Folosirea obiectului de sesiune ASP.NET</p> <p>Tema lab: Realizati un serviciu Web si un client care sa vehiculeze clase de tip Employee. Serviciul Web sa permita realizarea unei ierarhii. Interfata serviciului Web sa fie urmatoarea: Void AddManager(Employee e); Void AddEmployee(Employee m, Employee e); Employee GetManagerOf(Employee e); Employee GetEmployeesOf(Employee manager);</p> <p>Tema acasa. Creati un chat bazat pe combinare de servicii apelate din „alta parte”(un serviciu comunica intr-o directie unul intr-alta, am serviciu care face multicast (cu specificarea numarului de conexiuni) sau broadcast</p> <p>Laborator 7 Fire de executie</p> <p>7. Cum functioneaza multi-threadingul in .NET?</p> <p>7.1. Transmiterea parametrilor catre firele de executie 7.2. Probleme de acces simultan la date 7.3. Accesul exclusiv folosind instructiunea lock si a metodele Moitor.Enter/Exit 7.4. WaitHandles - Auto/ManualResetEvent si Mutex 7.5. Auto/ManualResetEvent 7.6. Mutex 7.7. Thread pool si Metodele asincrone</p> <p>Tema lab: Realizati o aplicatie care sa calculeze suma unui vector de numere. Fiecare doua numere consecutive din vector vor fi adunate de catre un fir. Rezultatul fiecarei adunari va fi scris intr-un fisier. pseudocod:...</p> <p>Tema acasa: Sa se realizeze o aplicatie care implementeaza un algoritm de sortare recursive dar la fiecare apel se genereaza un nou fir</p> <p>Laborator 8 Interoperabilitatea COM</p> <p>8. Interoperabilitate dinspre cod managed catre unmanaged 8.1. Sa intelegem Runtime Callable Wrapper</p>		
--	--	--

<p>8.2. Construirea unui assembly interop</p> <p>8.3. Construirea unui assembly de interoperabilitate cu ajutorul Visual Studio .NET</p> <p>8.4. Crearea unui assembly de inetroperabilitate cu ajutorul importatorului de librarie de tipuri.</p> <p>8.5. Interoperabilitate dinspre cod unmanaged catre managed</p> <p>8.6. Implementarea explicita a interfetelor</p> <p>8.7. Generarea automata a interfetei clasei</p> <p>8.8. P/Invoke Folosirea bibliotecilor Win32</p> <p>Tema lab: creati un COM server care sa dea ora exacta sub forma de sir de caracter</p> <p>Tema acasa: realizati un ceas grafic clasic care sa foloseasca com serverul anterior</p> <p>Laborator 9</p> <p>ADO.NET</p> <p>9. Arhitectura ADO.NET</p> <p>9.1. Content components</p> <p>9.2. DataSet</p> <p>9.3. Relatii si constrangeri</p> <p>9.4. DataView</p> <p>9.5. DataRelation</p> <p>9.6. Managed Providers</p> <p>9.7. Arhitectura managed provider-ilor oferiti de .NET Framework</p> <p>Tema lab:</p> <p>Realizati o aplicatie care sa citeasca datele dintr-o baza de date (doua tabele cu o relatie intre ele), sa afiseze datele intr-un DataGridView, sa permita editarea si salvarea modificarilor.</p> <p>Tema acasa: Sa se creeze un mini excell dar care sa suporte numai una maxim doua operatii in celulr. SE va salva in format propriu</p> <p>Laborator 10</p> <p>Lucrul cu document XML in .NET Framework</p> <p>10. XML</p> <p>10.1. Namespace-urile si clasele XML</p> <p>10.2. Citirea dcumentelor XML</p> <p>10.3. Citirea proprietatilor unui document XML</p> <p>10.4. Obtinerea informatiilor legate de tipurile de noduri:</p> <p>10.5. Scrierea documentelor XML</p> <p>10.6. Folosirea XmlDocument</p> <p>Tema lab:</p> <p>Realizati o aplicatie care sa citeasca si sa salveze nodurile dintr-un control de tip TreeView intr-un fisier XML. Nodurile din TreeView sa aiba si checkBox.</p> <p>Tema acasa: Sa se citeasca si sa se salveze in XML toata structura de directoare si fisiere ale uni drive selectat</p> <p>Laborator 11</p> <p>Globalizarea si localizarea aplicatiilor .NET</p> <p>11. Globalizarea</p> <p>11.1. Folosirea culturii invariabile</p> <p>11.2. Crearea fisierelor de resurse</p> <p>11.3. Construirea assembly-urilor satelit</p> <p>11.4. Compilarea Assembly-urilor satellite</p> <p>11.5. Instalarea in Global Assembly Cache a unui assembly satelit</p> <p>11.6. Structura de directoare pentru assembly-urile satelit care nu sunt instalate in global assembly cache</p> <p>11.7. Obtinerea resurselor din assembly-uri satelit</p> <p>11.8. Localizarea resurselor Windows Forms</p> <p>Tema lab: Realizati o aplicatie care sa ofere suport de limba pentru</p>		
---	--	--

<p>limba franceza (fr) si engleza (en). Trebuie localizate atat formurile cat si anumite mesaje folosite la afisarea unor MessageBox-uri. Localizarea sa se faca folosind assembly-uri satelit. Tema acasa: Realizati o aplicatie care sa ofere suport de limba pentru 6 limbi care sa se schimbe automat din doua in doua ore</p> <p>Laborator 12 Realizarea install-erelor 12. Planuri de deploy 12.1. VS.NET Setup Projects Tema lab : Realizati un installer care sa faca deploy la o aplicatie. Acesta sa adauge o cheie in registri si sa modifice un fisier de configurare a aplicatiei folosind datele obtinute din controalele definite cu editorul de interfete pentru utilizator. Adaugati o conditie de install la alegere. Tema acasa: sa se faca deploy la orice proiect anterior dar sa se scrie un serial in registri (codat) care sa fie verificat la pornire (hash etc) daca este ok</p> <p>Laborator 13 -14 Extinderea CLR Configuration System CLR O trecere in revista a sistemului de configurare Structura fisierului de configuratie Implementarea interfetei Setari lipsa si valori implicite Configuration Parenting Tema lab: Realizati o aplicatie care sa initializeze o clasa cu variabile de configurare folosind un section handler. Clasa sa aiba urmatoarea structura:... Tema acasa: Sa se realizeze un fisier de configurare la oricare din proiectele anterior predate precum si installer-ul asociat</p>		
8.2c Proiect	Metode de predare ²²	Observații
-	-	-
<p>Bibliografie aplicații (seminar / laborator / proiect):</p> <ol style="list-style-type: none"> 1. Biblioteca suport on off line a MSDN 2. http://www.developerfusion.co.uk 3. http://www.codeproject.com/csharp/socketsincs.asp 4. M. H. Zaharia, Sisteme Paralele si distribuite ("Parallel and distributed systems"), Ed. "Gh. Asachi", 2003, ISBN 973-621-056-1, 187 pg format academic 5. Mihai Horia Zaharia, Cătălin Horghidan, Claudiu Mihăilă, Sisteme Distribuite: indrumar de laborator, Politehnum, 2007, ISBN 978-973-621-175-1, 225 format academic 		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului²³

<ul style="list-style-type: none"> • Materia pune bazele pentru urmatoarele cursuri de nivel superior: algoritmi paraleli si distribuiti, inteligenta artificiala, baze de date distribuite, tehnologii web, regasirea informatiilor pe web, extragerea infromatiilor, retele senzor • Materia ofera ajutor pentru materiile de sisteme multiagent, comert electronic, baze de date orientat obiect • Materia creaza legaturi directe/imediate cu umartoarele discipline : retele de calculatoare, modelare si simulare, arhitectura calculatoarelor paralele, baze de date, paradigma de programare • Cu privire la utilitatea pentru piata muncii: deoarece prezinta cam tote tipurile de calcul distribuit si de mare peformata impreuna cu problemele lor aceasta materi acopera 90% din tipurile de abrdari si tehnologii majore utilizate in dezvoltarea apliactiilor pentru o societate informationala

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<ul style="list-style-type: none"> • Cunoștințe teoretice însușite 	Teste pe parcurs ²⁴ : 0 Teme de casă: 13	0% 25%

	(cantitatea, corectitudinea, acuratețea)	Evaluare finală:	50% (minim 5)
10.5a Seminar	• Frecvența/relevanța intervențiilor sau răspunsurilor	• Evidența intervențiilor, portofoliu de lucrări (referate, sinteze științifice)	0%
10.5b Laborator	• Cunoașterea aparatului, a modului de utilizare a instrumentelor specifice; evaluarea unor instrumente sau realizări, prelucrarea și interpretarea unor rezultate	• Răspuns oral • Demonstrație practică	25% (minim 5)
10.5c Proiect	• Calitatea proiectului realizat, corectitudinea documentației proiectului, justificarea soluțiilor alese	• Autoevaluarea, prezentarea și/sau susținerea proiectului • Evaluarea critică a unui proiect	0% (minim 5)
10.5d Alte activități ²⁵	•	•	0% (minim 5)
10.6 Standard minim de performanță ²⁶ cf CT1			
• Dobândirea abilităților necesare înțelegerii problemei, proiectării, validării și implementării unor aplicații distribuite de mari dimensiuni.			

Data completării,

22.09.2014

Semnătura titularului de curs,
Mihai Horia Zaharia

.....

Semnătura titularului de aplicații,
Mihai Horia Zaharia

.....

Alexandrescu Adrian

.....

Gavrila Ionut

.....

Data avizării în departament,

25.09.2014

Director departament,
Petru Cascaval

.....

¹ Licență / Master

² 1-4 pentru licență, 1-2 pentru master

³ 1-8 pentru licență, 1-3 pentru master

⁴ Examen, colocviu sau VP A/R – din planul de învățământ

⁵ DF - disciplină fundamentală, DID - disciplină în domeniu, DS – disciplină de specialitate sau DC - disciplină complementară - din planul de învățământ

⁶ Este egal cu 14 săptămâni x numărul de ore de la punctul 3.1 (similar pentru 3.5, 3.6abc)

⁷ Liniile de mai jos se referă la studiul individual; totalul se completează la punctul 3.7.

⁸ Între 7 și 14 ore

⁹ Între 2 și 6 ore

¹⁰ Suma valorilor de pe liniile anterioare, care se referă la studiul individual.

¹¹ Suma dintre numărul de ore de activitate didactică directă (3.4) și numărul de ore de studiu individual (3.7); trebuie să fie egală cu numărul de credite alocate disciplinei (punctul 3.9) x 24 de ore pe credit.

¹² Se menționează disciplinele obligatorii a fi promovate anterior sau echivalente

¹³ Tablă, videoproiector, flipchart, materiale didactice specifice etc.

¹⁴ Tehnică de calcul, pachete software, standuri experimentale, etc.

¹⁵ Competențele din Grilele G1 și G1bis ale programului de studii, adaptate la specificul disciplinei, pentru care se repartizează credite (www.rncis.ro sau site-ul facultății)

¹⁶ Din planul de învățământ

¹⁷ Creditele alocate disciplinei se distribuie pe competențe profesionale și transversale în funcție de specificul disciplinei

¹⁸ Titluri de capitole și paragrafe

¹⁹ Expunere, prelegere, prezentare la tablă a problematicei studiate, utilizare videoproiector, discuții cu studenții (pentru fiecare capitol, dacă este cazul)

²⁰ Discuții, dezbateri, prezentare și/sau analiză de lucrări, rezolvare de exerciții și probleme

²¹ Demonstrație practică, exercițiu, experiment

²² Studiu de caz, demonstrație, exercițiu, analiza erorilor etc.

²³ Legătura cu alte discipline, utilitatea disciplinei pe piața muncii

²⁴ Se va preciza numărul de teste și săptămânile în care vor fi susținute.

²⁵ Cercuri științifice, concursuri profesionale etc.

²⁶ *Se particularizează la specificul disciplinei standardul minim de performanță din grila de competențe a programului de studii.*