Subiectele pentru proba practică din cadrul examenului de Sisteme Distribuite Sesiunea iunie 2014

Considerații generale cu privire la proba practică:

La examen studentul va avea acces la „MSDN Library for Visual Studio 2008 SP1” care poate fi descărcată de la adresa: http://www.microsoft.com/en-us/download/details.aspx?id=20955.

Biletul va fi compus din două subiecte din lista de mai jos. Combinația de două subiecte va fi generată în mod aleatoriu. Timpul de rezolvare pentru ambele subiecte va fi de 2 ore.

Pentru a promova proba de laborator trebuie îndeplinite următoarele:

- Măcar una din problemele de pe bilet trebuie rezolvată complet în decursul celor două ore.
- Studentul trebuie să fie capabil să răspundă la întrebări cu privire la rezolvarea problemei.

În cazul în care studentul primește o notă mai mică ca 5 la proba de laborator nu va mai susține proba teoretică, examenul considerându-se picat.

Pentru studenții care au rezolvat o problemă complet se mai poate extinde cu maxim jumătate de ora examinarea practică în cazul în care mai au nevoie de timp ca să termine ce-a de-a doua problemă de pe bilet.

Lista de subiecte

1. Realizați o aplicație în stilul calculatorului din Windows. Aplicația va trebui să permită operațiile de adunare, scădere, înmulțire, împărțire, ridicare la putere și logaritm pentru numere pozitive și negative cu virgulă.
2. Realizați o aplicație în stilul calculatorului din Windows. Aplicația va trebui să permită operațiile de adunare, scădere, înmulțire și împărțire. Programul va lucra cu numere întregi și va permite expresii de forma: 1+3*7. Pentru implementare se poate folosi forma poloneză.
3. Realizați o aplicație client-server care să suporte mai mulți clienți. Serverul trebuie să fie de tip multicast-echo: orice mesaj trimis de client ajunge la toți clienții (inclusiv la cel care a trimis mesajul). Pentru implementare se vor folosi socket-uri Tcp și metodele blocante Send() și Receive().
4. Realizați o aplicație client-server care să suporte mai mulți clienți. Serverul trebuie să fie de tip echo: orice mesaj trimis de client ajunge înapoi la client. Pentru implementare se vor folosi socket-uri Tcp, metoda blocantă Send() și metoda BeginReceive() pentru recepționarea asincronă a datelor atât la server cât și la client.
5. Realizați două aplicații care să utilizeze serviciul de mesaje din Windows (MSMQ). Prima aplicație va scrie în MSMQ obiecte de tipul Student (care va conține câmpurile: nume, prenume și grupa) la fiecare 3 secunde, iar a doua va citi datele în mod asincron și le va afișa la consolă. Se vor folosi formatorul BinaryMessageFormatter/ XmlMessageFormatter.
6. Realizați un serviciu Web (care să ruleze pe portul 12345) și o aplicație client care să permită notarea studenților la diverse materii. Serviciile care vor fi disponibile sunt:
   - adaugaMaterie(nume) => idMaterie (generat de server),
   - adaugaStudent(nume, grupa) => idStudent (generat de server),
   - adaugaNota(nota, idStudent, idMaterie),
   - returneazaNote(idMaterie) => o listă de perechi numeStudent-nota (se poate folosi o structura de tip Dictionary),
   - returneazaStudenti(grupa) => o lista cu numele studenților din grupa respectivă.
7. Realizați un server care expune un obiect wellknown (SAO) în modul Singleton/ SingleCall și un client care utilizeze obiectul respectiv pentru a permite notarea studenților la diverse materii. Metodele din obiectul remote sunt:
   - adaugaMaterie(nume) => idMaterie (generat de server),

- adaugaStudent(nume, grupa) => idStudent (generat de server),
- adaugaNota(nota, idStudent, idMaterie),
- returneazaNote(idMaterie) => o listă de perechi numeStudent-nota (se poate folosi o structura de tip Dictionary),
- returneazaStudenti(grupa) => o lista cu numele studenților din grupa respectivă.

8. Realizați un server care expune un obiect activat de client (CAO) și un client care utilizeze obiectul respectiv pentru a permite notarea studenților la diverse materii. Metodele din obiectul remote sunt:
    - adaugaMaterie(nume) => idMaterie (generat de server),
    - adaugaStudent(nume, grupa) => idStudent (generat de server),
    - adaugaNota(nota, idStudent, idMaterie),
    - returneazaNote(idMaterie) => o listă de perechi numeStudent-nota (se poate folosi o structura de tip Dictionary),
    - returneazaStudenti(grupa) => o lista cu numele studenților din grupa respectivă.
9. Realizați un serviciu Web și un client care sa vehiculeze clase de tip Employee. Testați în aplicația client toate metodele din serviciul web creat. Numele angajaților și a managerilor sunt citite de la tastatură. Interfața serviciului Web trebuie să fie următoarea:
    - void AddManager(Employee e);
    - void AddEmployee(Employee m, Employee e);
    - Employee GetManagerOf(Employee e);
    - Employee GetEmpoyeesOf(Employee manager)
10. Realizați o aplicație care să calculeze suma unui vector de numere. Fiecare două numere consecutive din vector vor fi adunate de către un fir. Rezultatul fiecărei adunări va fi scris într-un fișier.
11. Sa se implementeze un calcul $\sum_0^n i$ cu 4 thread-uri simultane care fiecare calculeaza pe un subinterval folosind modelul master/slave
12. Să se realizeze o procesare după model pipeline al unui tablou de întregi. Primul thread din pipe înmulțește toate elementele vectorului V cu o constantă alpha, următorul thread din pipe va ordona vectorul, iar la final ultimul thread îl va afișa în coordinate x și y.
13. Realizați o aplicație care implementează un algoritm de sortare recursiv, iar la fiecare apel să se genereze un nou fir de execuție.
14. Realizați o aplicație care să permită citirea a *n* fișiere text și scrierea într-un alt fișier. Citirea și scrierea se vor face linie cu line. Fișierele vor fi citite în paralel. Pentru citire se vor folosi fire de execuție create cu ajutorul clasei Thread.
15. Realizați o aplicație care să permită citirea a *n* fișiere text și scrierea într-un alt fișier. Citirea și scrierea se vor face linie cu line. Fișierele vor fi citite în paralel. Citirea trebuie să se facă asincron prin utilizarea unui delegat asincron.
16. Realizați o aplicație Windows Forms care să utilizeze funcția Win32 API GetSystemTime (definită în kernel32.lib) pentru a afișa din secundă în secundă timpul curent al sistemului.
17. Realizați o aplicație care să citească și să salveze nodurile dintr-un control de tip TreeView într-un fișier XML. Nodurile din TreeView trebuie să aibă și checkbox.
18. Realizați o aplicație care să permită citirea și scrierea în XML a întregii structuri de directoare și fișiere ale unui director selectat.
19. Realizați o aplicație care să ofere suport de limbă pentru 3 limbi care să se schimbe automat din 2 în 2 secunde.
20. Realizați o aplicație Windows Forms care va conține două butoane. La apăsarea butonului Calculeaza, se calculează suma unui șir de numere specificate într-un fișier de configurare, iar la apăsarea butonului Salveaza, rezultatul sumei va fi scris într-un fișier XML.