

Programare Orientată Obiect

= Tema 2 =

Scop general:

- înțelegerea conceptelor simple de programare orientată obiect;
- utilizarea mediului de dezvoltare Visual Studio 6;

Scop specific:

- înțelegerea conceptelor de constructor, constructor de copiere, destructor, operator de atribuire, alți operatori, funcții friend, operatori de lucru cu streamuri;
- implementare în limbaj C++ a unei clase, definirea și implementarea funcționalităților necesare; scrierea unor programe de test pentru a pune în evidență implementarea realizată;
- deprinderea cu particularitățile lucrului în mediul de dezvoltare Visual C++ 6: lucrul cu proiecte, utilizare editor/compiler, depanarea aplicațiilor prin execuție pas cu pas, inspectarea variabilelor;
- deprinderea cu utilizarea MSDN pentru regăsirea informațiilor utile;

Documentație specifică:

- carte C/C++; recomandarea noastră H. Schildt, The complete reference C++, 3rd edition, ed. McGraw Hill, 1999, ISBN 0-07-882476-1;
- Microsoft MSDN 6, 2CD sau web site;
- lucrări de laborator la disciplina PCLP 1 & 2 și POO, Fac. ETTI;

Cerințe:

- tema conține o singură problemă. Enunțul problemei permite o rezolvare cu o complexitate ce crește gradual (precizând o soluție minimală); enunțul și modalitatea de rezolvare a fost discutat pe parcursul orelor de aplicații;
- va fi completată o fișă de verificare dovedind cunoașterea soluției;
- notarea este proporțională cu complexitatea și corectitudinea soluției furnizate pe baza informației din fișa de verificare și pe baza răspunsurilor la întrebări legate de soluția propusă;
- tema se predă semnată, scris de mână sau tipărit față-verso, capsată și introdusă într-o folie. **Tema conține tot codul sursă indicând fișierele și este însoțită de fișa de verificare. Fișa de verificare se va completa pe loc la predarea temei.** Tema se poate verifica prin întrebări la predare sau până la definitivarea notei la disciplina POO, inclusiv la examen;
- temele și proiectul se păstrează într-un dosar;
- **termen de predare: în săptămâna a 11-a.**

Tema II - Problema

Cerințe minimale:

Să se implementeze o clasă Matrice și o clasă Vector precum și principalele funcționalități necesare pentru lucrul cu matrici / vectori.

Să se scrie un program de test în care să se exemplifice crearea de matrici și vectori, citirea și afișarea datelor de la consolă, realizarea de operații matematice cu scalari, vectori și matrici, scrierea și citirea unor vectori și matrici într-un fișier, încărcarea datelor din fișier.

Observație 0 – Se discută o implementare cu alocare dinamică de tip C++;

Observație 1 - În cadrul laboratorului s-a urmărit funcționarea unui program pentru Vectori;

Observație 2 – În Tema I s-a realizat o aplicație C pentru lucru cu vectori și matrici cu cerințele cunoscute;

Cerințe în completare (1):

Se precizează funcționalitățile sugerate a fi implementate:

- constructor implicit;
- constructor explicit (mai multe variante ... permite specificarea dimensiunii, permite specificarea dimensiunii și a unei valori inițiale pentru elementele matricii, permite specificarea dimensiunii și a valorilor pentru elementele matricii într-un tablou uni/bi-dimensional);
- constructor de copiere și operator de atribuire;
- metode pentru aflarea dimensiunilor matricii / dimensiunii vectorului;
- metode pentru inițializarea valorilor matricii (permite specificarea unei valori inițiale pentru elementele matricii, permite specificarea valorilor pentru elemente într-un tablou uni / bi-dimensional);
- metode pentru citirea / afișarea la consolă;
- operatori << și >> de lucru cu streamuri;
- operatori binari +, -, * (implementează $A + B$);
- operatori binari +, -, * de lucru cu matrici și scalari (implementează $5+C$ și $C+5$);
- operator unar - (implementează $-D$);

Se va scrie un program de test pentru exemplificarea funcționalităților.

Cerințe în completare (2):

Pentru exemplificarea lucrului cu fișiere se va realiza:

- un program care creează un număr de vectori, matrici și le scrie într-un fișier;
- un al doilea program care citește fișierul și afișează conținutul lui;

Cerințe în completare (3):

Să se realizeze un mecanism prin care să se afișeze numărul de vectori și matrici existenți în memorie.

Cerințe în completare (4):

Să se realizeze un mecanism prin care să se afișeze la cerere conținutul tuturor vectorilor și matricilor aflate în memorie. Să se utilizeze funcții de bibliotecă pentru testarea integrității zonei de memorie alocate dinamic (heap).