

Programare Orientată pe Obiecte Lucrarea de laborator Nr. 1

Mediul de dezvoltare Developer Studio. Elementele de bază ale unui proiect în C++.

Programarea orientată pe obiecte este o metodă de programare în care programele sunt organizate ca și colecții de obiecte cooperante, fiecare dintre ele reprezentând o instanță a unei clase, iar clasele sunt membre ale unei ierarhii de clase, corelate între ele prin relații de moștenire.

Limbajul C++ este unul dintre cele mai utilizate limbaje de programare orientate pe obiecte; compilatoare, biblioteci și instrumente de dezvoltare a programelor C++ sunt disponibile atât pentru calculatoarele personale cât și pentru cele mai dezvoltate sisteme și stații de lucru.

În cadrul laboratorului de POO se va folosi Microsoft Visual C++ 6.0 rulând sub sistemul de operare Windows XP.

În această lucrare de laborator, veți începe să învățați câteva dintre conceptele de bază ale limbajului C++ și veți crea un program simplu în mod consolă.

1. Logarea pe stații

Cele 18 stații de lucru din laborator sunt legate într-o rețea locală. Logarea pe fiecare stație se realizează astfel:

- Se apasă Ctrl+Alt+Del;
- În caseta de dialog care apare se selectează pentru domeniu (Domain) EAII (nu C1, C2 ... C18 care reprezintă numele stațiilor locale), pentru numele utilizatorului (User name) se introduce user_bdi (*i* reprezentând numărul stației de lucru ce aparține intervalului 1-18) iar parola (Password) este identică cu numele utilizatorului.

Lucrările de laborator sunt accesibile în format electronic pe stația C1 (accesare: My Computer, My Network Places, etc. sau prin intermediul instrumentului Windows Commander cu Shortcut pe Desktop reprezentat de icon-ul ce simbolizează o dischetă în culorile albastru, roșu, alb), în directorul Laborator_POO, sub formă de fișiere cu extensia .pdf (de exemplu, LaboratorPOO1.pdf) și pot fi accesate ca atare sau copiate de către fiecare utilizator pe fiecare stație de lucru în subdirectorul Studenți (C:\POO\Studenți). Vizualizarea se face cu ajutorul instrumentului Adobe Acrobat (Shortcut-ul este disponibil pe Desktop). În subdirectorul Studenți puteți crea propriile subdirectoare.

Programele scrise folosind Visual C++ sunt *compile* ceea ce înseamnă că un fișier text denumit *fișier sursă* este convertit într-un program de sine stătător care poate fi rulat în afara mediului Visual C++. Aproape toate sistemele de dezvoltare C++ compilează programe executabile. Unele limbaje de programare, cum ar fi anumite versiuni de BASIC, crează programe care pot rula numai în mediul de dezvoltare. Alte limbaje, cum ar fi Smalltalk, rulează în corelație cu o bibliotecă de rulare, care asistă programul aflat în execuție. Dimpotrivă, programele scrise folosind Visual C++, pot fi scrise și compilate într-un program executabil. Acel program poate fi apoi executat pe orice calculator care dispune de un sistem de operare Windows. În majoritatea sistemelor de dezvoltare C++, un program C++ parcurge mai multe etape până la transformarea sa într-un program executabil:

- Se crează unul sau mai multe fișiere sursă, care conțin textul unui program C++ (fișiere cu extensia .cpp).

- Se compilează programul folosind un compilator de C++. Această etapă poate consta, de fapt, din mai multe comenzi de compilare a fișierelor sursă în fișiere intermediare, cunoscute și sub numele de fișiere în cod obiect – object code (fișiere cu extensia .obj).
- Fișierele în cod obiect sunt legate la un loc folosind un instrument denumit program de editare a legăturilor – linker.
- Acum se poate lansa programul executabil (fișier cu extensia .exe).

Cu ceva timp în urmă, majoritatea sistemelor de dezvoltare pretindeau ca fiecare din aceste etape să fie executată fie cu produse separate, din linia de comandă, fie cu ajutorul unui instrument care determină fișierele care necesită actualizarea. Folosind instrumente moderne, precum Visual C++, alcătuirea unui program este cu mult mai simplă. Mediul Visual C++ permite editarea, compilarea, executarea și depanarea unui program în interiorul aceluiași mediu. Toate etapele prezentate mai sus sunt supervizate de mediul de dezvoltare Visual C++ (întâlnit și sub acronimul IDE – Integrated Development Environment, sau Developer Studio sau Visual Studio), ceea ce permite utilizatorului să se concentreze asupra scrierii programului, în loc de a da atenție unei mulțimi de detalii.

2. Microsoft Visual C++ 6.0

Lansarea mediului Visual C++ se poate face accesând Shortcut-ul de pe Desktop care este însoțit de icon-ul ce simbolizează semnul infinit în culorile roșu, galben, verde și albastru (MSDEV – Microsoft Developer Studio). La lansarea mediului Visual C++ este afișată fereastra Visual Studio. Visual Studio este numele dat interfeței cu utilizatorul a lui Visual C++ și va reprezenta suprafața de lucru.

Când se folosește Visual C++ pentru dezvoltarea unui program, majoritatea activității este privită drept un proiect. Fiecare proiect stochează informația necesară creării unui program în C++. În funcție de tipul și de complexitatea programului, proiectul se poate reduce la câteva fișiere sau poate include sute de fișiere.

Creați un proiect astfel:

- Selectați comanda **New** a meniului **File** pentru a crea un nou proiect; caseta de dialog New va fi afișată.
- Specificați tipul proiectului pe care îl creați din pagina **Projects** a casetei de dialog New (din lista proiectelor posibile alegeți **Win32 Console Application**).
- În caseta **Project Name** din caseta de dialog New specificați numele proiectului (indicație: personalizați proiectele pe care le creați).
- Folosiți caseta **Location** pentru a preciza directorul în care vor fi plasate fișierele proiectului (C:\POO\Studenti\....). Calea afișată inițial în caseta Location depinde de opțiunile exprimate la instalarea mediului Visual C++. Pentru a modifica această locație, fie editați calea explicit, fie apăsând butonul aflat în partea dreaptă a casetei Location urmăriți ierarhia de directoare până în subdirectorul dorit. Locația implicită se bazează pe numele proiectului și este C:\Program Files\Microsoft Visual Studio\My Projects\numele proiectului.
- După stabilirea opțiunilor din cadrul casetei de dialog New, apăsarea butonului **OK** inițiază generarea proiectului. În cazul unei aplicații de tip Win32 Console Application, procesul de generare a proiectului conține un singur pas în care trebuie să alegeți ce tip de aplicație în mod consolă veți crea. Alegeți **An empty project**. La apăsarea butonului **Finish** este afișată o casetă de dialog care vă informează despre crearea unui „schelet” de proiect vid, adică nu vor fi create sau/și adăugate fișiere suplimentare la proiect. Este precizată de asemenea calea către directorul proiectului. În acest moment, puteți vizualiza urmând această cale fișierele create.

Închideți mediul de dezvoltare.

3. Mediul Developer Studio

Mediul de dezvoltare Developer Studio utilizat de Visual C++ pare destul de complex la prima vedere (peste 100 de opțiuni de meniu și aproximativ tot atâtea butoane de pe bara de instrumente care pot fi selectate). Multe dintre acestea conduc la casete de dialog complexe și la pagini de proprietăți care conțin numeroase opțiuni. Funcționalitatea atât de bogată oferită de Developer Studio este justificată de faptul că acest mediu este utilizat pe scară largă pentru a produce aplicații complexe, profesionale. În cadrul acestui laborator, veți învăța o parte din întreaga funcționalitate.

Lansați din nou Visual C++. Deschideți proiectul creat anterior astfel:

- Din meniul **File**, selectați proiectul din cadrul listei **Recent Workspaces**.
- Pentru a deschide un proiect care nu figurează în lista Recent Workspaces, selectați **Open Workspace** din meniul File. Pe ecran este afișată fereastra Open Workspace în care accesați directorul folosit pentru proiectul creat. Selectați din listă fișierul care poartă numele proiectului și extensia **.dsw** și alegeți **Open**.

Indiferent de metoda utilizată, proiectul ar trebui să fie acum deschis, cu titlul său afișat în bara de titlu a ferestrei Developer Studio. Mediul de lucru a reținut unde ați rămas și afișarea este identică cu cea de la închiderea mediului de dezvoltare.

Afișarea cuprinde trei ferestre: cea a spațiului de lucru al proiectului, cea pentru editare și cea de ieșire. Fereastra spațiului de lucru al proiectului – în partea stângă, este o fereastră andocabilă. Există mai multe astfel de ferestre (fereastra Output – în partea de jos este una dintre ele), unele apărând în modul depanare. Pentru a comuta între modul liber și modul andocabil pentru o fereastră, efectuați un dublu click pe bara de titlu a ferestrei respective. Fereastra spațiului de lucru al proiectului poate fi închisă prin efectuarea unui click pe butonul de închidere, aflat în colțul dreapta sus. Redeschiderea ferestrei se realizează prin selectarea comenzii **Workspace** din meniul **View**. Toate ferestrele andocabile au comportamente similare. Fereastra spațiului de lucru al proiectului permite vizualizarea acestuia din mai multe perspective. După ce ați deschis proiectul, în partea inferioară a spațiului de lucru, sunt disponibile două pagini: **ClassView** și **FileView**. Paginile din partea inferioară a ferestrei împart proiectul în componente logice. Efectuați click pe o pagină pentru a vizualiza reprezentarea corespunzătoare. Fiecare reprezentare afișează o structură arborescentă de elemente care reprezintă componente ale proiectului. Puteți să expandați sau să condensați elementele arborelui efectuând click pe simbolul plus (+) sau minus (-) alăturat în partea stângă a acestora. Pentru că ați creat o aplicație în mod consolă ca un proiect gol nici un fișier nu există în acest moment în proiect și nu poate fi vizualizat.

Exercițiul 1. Creați alte tipuri de proiecte pentru aplicații în mod consolă și vizualizați fișierele create de către mediul de dezvoltare.

Despre pagina ClassView vom discuta la momentul oportun. Pentru tipurile de proiecte create analizați conținutul paginii FileView. Este afișată o listă cu o parte din fișierele incluse în proiect. Fișierele sunt organizate pe cataloage în funcție de tipul acestora (cataloagele nu reprezintă locația fișierelor pe hard disk). Efectuarea unui dublu click pe numele unui fișier deschide fișierul în fereastra din dreapta, fereastra de editare. Faceți o comparație între această listă și conținutul directorului proiectului în care veți vedea alte câteva tipuri de fișiere. Unele dintre acestea sunt fișiere temporare folosite de către Developer Studio; altele depind de tipul proiectului și de conținutul său. În pagina File View se pot adăuga fișiere proiectului sau se pot înlătura fișiere din proiect prin selectarea numelui fișierului și apăsarea tastei **Delete**. Aceasta duce doar la înlăturarea fișierului din proiectul Visual Studio fără a șterge efectiv fișierul de pe disc.

4. Crearea unui fișier sursă în C++

Cea mai importantă componentă a oricărui program în C++ o reprezintă fișierele sursă. Pentru simplitate, vați lucra la început cu programe care conțin un singur fișier sursă. Deși programul model prezentat în listingul următor este foarte scurt, conține o bună parte dintre elementele existente în toate programele C++:

```
# include <iostream.h>
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

Reveniți la primul proiect creat. O modalitate de a deschide un fișier sursă nou în vederea editării este următoarea: selectați **File**, apoi **New** din meniul principal. Din pagina **Files** a casetei de dialog **New** care este afișată alegeți opțiunea **C++ Source File**, dați un nume noului fișier (poate fi același ca al proiectului). Observați că, implicit este setată opțiunea de adăugare a noului fișier proiectului curent. În fereastra de editare se deschide noul fișier. Adăugați codul de mai sus. Compilați fișierul sursă (**Build – Compile nume_fișier.cpp**). Legați proiectul (**Build – Build nume_proiect.exe**). Proiectul poate fi alcătuit și prin selectarea direct a comenzii **Build nume_proiect.exe**, compilarea realizându-se implicit. În fereastra de jos vor apare informații privind compilarea fișierelor și legarea proiectului. Dacă ați introdus corect conținutul listingului proiectul va fi creat fără erori și fără avertizări.

Exercițiul 2. Intenționat stergeți din codul programului ghilimele sau punct și virgulă sau paranteze. Compilați de fiecare dată și urmăriți erorile și mesajele de avertizare. Un dublu click pe prima linie a unui mesaj de eroare poziționează cursorul în fereastra de editare în dreptul liniei unde compilatorul consideră că se află eroarea. Pot apare erori ca urmare a altor erori așa încât se corectează prima eroare și apoi se recompilează s.a.m.d.

După compilarea și legarea fără erori a proiectului, puteți rula programul selectând **Build – Execute nume_proiect.exe** din meniu.

5. Elementele de bază ale unui proiect în C++

Programul creat are multe în comun cu programe C++ de dimensiuni mult mai mari:

- Prima linie din program este un mesaj către compilator de a include și un alt fișier la compilarea fișierului creat. Instrucțiunea `#include` cere compilatorului să caute fișierul `iostream.h` și să-l introducă în fișierul sursă.
- Următoarea linie reprezintă începutul funcției principale (`main`). Toate programele în C++ constau dintr-una sau din mai multe funcții. Funcția denumită `main` este locul în care începe execuția programului, motiv pentru care toate programele în C++ trebuie să aibă o funcție denumită `main`. Funcția returnează (`return`) o valoare întreagă (`int`) sistemului de operare la încheierea executării. Funcția nu acceptă parametrii. Sistemul de operare folosește valoarea returnată de funcția `main` pentru a afla dacă programul a fost executat cu succes. Este posibilă returnarea către sistemul de operare a unor valori care indică reușită sau eșec.

Exercițiul 3. Pentru celelalte tipuri de proiecte pe care le-ați creat identificați funcția `main` deja creată de către mediul de dezvoltare.

- Următoarele 4 linii de după `int main()` reprezintă corpul funcției `main`, cunoscut și sub numele de bloc instrucțiune, încadrat întotdeauna între o pereche de acolade. În interiorul funcției `main` se află o singură instrucțiune (care se încheie cu `;` ca toate instrucțiunile din C++) care afișează la consolă o linie de caractere folosind obiectul `cout` din `iostream`. Folosind biblioteca `iostream`, se simplifică operațiile de intrare/ieșire pentru programul în mod consolă. Biblioteca `iostream` folosește simbolul `<<` pentru ieșiri și `>>` pentru intrări din, respectiv în fluxul de date (secvență de octeți, fișier pe disc, datele de ieșire pentru imprimantă sau pentru ecrane mod caracter, etc). O regulă simplă este aceea că atunci când întâlnești simbolul `<<`, valoarea de la dreapta simbolului va fi ieșirea pentru obiectul IO din stânga iar când întâlnești simbolul `>>`, datele din obiectul IO din stânga sunt stocate într-o variabilă din dreapta. `cout` înseamnă *console output* („ieșire la consolă”) iar `<<` poate fi tradus prin „pune la”.

Ca o ultimă observație, când utilizați C++, nu uitați că deosebirea dintre literele mari și mici este importantă, dar spațiile libere, cum ar fi numărul de spații dinaintea cuvântului `cout`, nu sunt importante din punctul de vedere al compilatorului. Editorul ordonează automat codul sursă și oferă facilități de highlight pentru o deosebire ușoară a cuvintelor cheie ale limbajului de restul codului.

`endl` (*end line*) indică sfârșit de linie, astfel că, după afișarea mesajului, cursorul trece la linie nouă.

Exercițiul 4. Afișați la consolă mai multe mesaje pe aceeași linie, apoi pe linii diferite.