

**Mirel COȘULSCHI**

**Octavian MUSTAFA**

---

**Programare în C++. Concepte moderne și aplicații**



**Mirel COȘULSCHI**

**Octavian MUSTAFA**

# **Programare în C++.**

## **Concepte moderne și aplicații**



**Editura UNIVERSITARIA**  
**Craiova, 2015**



**Editura PROUNIVERSITARIA**  
**București, 2015**

## **Referenți științifici:**

Prof. univ. dr. Ion IANCU

Lect. univ. dr. Petre BAZAVAN

Copyright 2015 Editura Universitaria

Toate drepturile sunt rezervate Editurii Universitaria

---

## **Descrierea CIP a Bibliotecii Naționale a României**

**COȘULSCHI, MIREL**

**Programare în C++ : concepte moderne și aplicații /**

Mirel Coșulschi, Octavian Mustafa. - Craiova : Universitaria ;

București : Pro Universitaria, 2015

Bibliogr.

ISBN 978-606-14-0928-0

ISBN 978-606-26-0333-5

004.43 C++

# Prefață

În materialul care urmează prezentăm conceptele și tehnicile principale ale *programării orientate spre obiecte* folosind programe scrise în limbajul C++ modern. Codurile tuturor programelor sunt complete, ceea ce îi va permite cititorului să constate că acest tip de programare reușește să scoată în relief, cu multă ușurință, ideile centrale din algoritmiile programelor respective. Este, evident, mai simplu să tratăm o fereastră de pe ecran ca pe un *tot unitar* — adică, un *obiect* — în loc să descriem, ori de câte ori fereastra trebuie deplasată, mișcarea fiecăruia dintre componentele sale (bară orizontală, butoane, etc).

Unele dintre programe sunt construite în variante de complexitate crescândă. În acest fel, descoperim necesitatea unor caracteristici ale obiectelor manipulate de programe care pot părea — la prima vedere — inutile. Există și programe care tratează, profitând de expresivitatea limbajului C++, diverse probleme de matematică ori algoritmică. Ele sunt de interes și pentru elevii, studenții ori cercetătorii cu preocupări în matematica computațională.

Fiecare dintre capitolele lucrării se încheie cu o listă de probleme, având grade variate de dificultate dar și anumite sugestii de rezolvare. Parcurgerea lor îi va permite cititorului aprofundarea câtorva dintre tehnicile importante ale programării orientate spre obiecte, componentă indispensabilă a educației oricărui programator.

Craiova, mai, 2015

Mirel Coșulschi  
Octavian Mustafa



# Capitolul 1

## Început

Limbajul C++ a fost inventat de profesorul Bjarne Stroustrup de la Universitatea A&M Texas și este descris în cartea [39]. Limbajul este o extensie (un superset) a limbajului C [19], fiind denumit la început și *C cu clase*, vezi [38]. De aceea, instrucțiunile dintr-un program C “portabil” (vezi POSIX, ANSI C [34]) vor funcționa excelent în cadrul unui program C++. O *versiune de lucru*<sup>1</sup> a standardului pentru limbajul C++ se găsește la adresa [37].

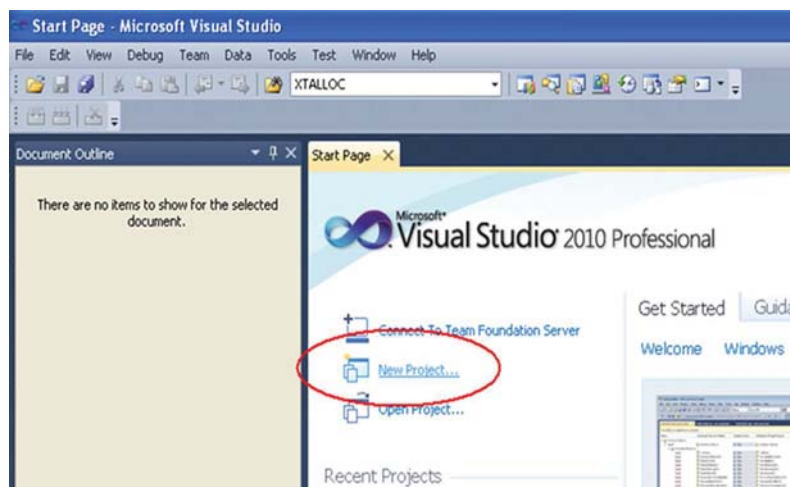


Figura 1.1 Visual Studio 2010, pagina de start

Începem discuția pregătind mediul de dezvoltare a programelor noastre, aici **Visual Studio 2010 Professional**<sup>2</sup> (VS). Altă opțiune este oferită de **g++**, compilatorul gratuit de C++, care poate fi folosit via **Cywin**, urmând instrucțiunile de la adresa [4].

Construim un director (folder) destinat viitoarelor programe C++, și anume **C:\Proiecte leMele\_Cpp**, după care apelăm Visual Studio-ul, a se vedea Figura 1.1. Acesta va fi testat cu un program C extrem de simplu.

Alegem **New Project**, după care, a se vedea Figura 1.2, în lista **Installed Templates**, optăm pentru **Visual C++**. În tabelul central, selectăm **Empty Project**. Numele proiectului — în primul din câmpurile albe din partea de jos a ferestrei (pop-up) **New Project** — va fi

<sup>1</sup>În limba engleză, *draft*.

<sup>2</sup>Varianta gratuită, **Express**, a VS-ului poate fi descărcată de la adresa [46].

`primul_program` iar pentru a stabili locația acestuia în directorul construit anterior folosim butonul **Browse**.

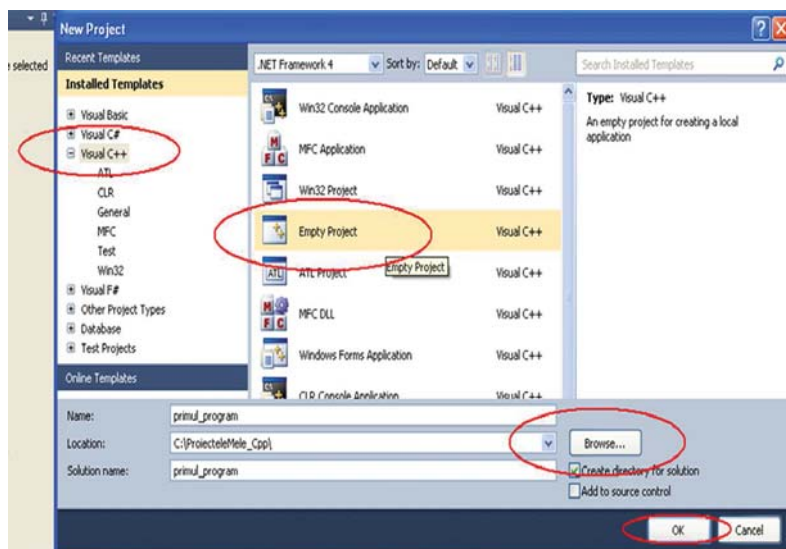


Figura 1.2 Crearea unui proiect nou, *primul\_program*

Ultimul câmp al ferestrei **New Project**, și anume **Solution name** (soluția...), va deveni, automat, `primul_program`. Evident, *soluția* — fișierul `primul_program.sln` din directorul `C:\ProiecteleMele_Cpp\primul_program` — va permite în urma unei duble apăsări pe butonul stâng al mouse-ului încărcarea proiectului nostru în mediul Visual Studio, indiferent dacă acesta este în execuție sau nu. Însă aceasta numai după ce vom fi construit primul nostru proiect!

Dacă totul este în ordine, apăsăm butonul **OK** (a se vedea Figura 1.2). Mai departe, în funcție de setările mediului Visual Studio, pe ecran vor apărea diferite ferestre în care scriem cod, respectiv sunt afișate diverse informații, a se vedea Figura 1.3. Folosind butonul **View** din meniul principal, putem vizualiza — dacă nu au fost setate deja — ferestrele **Solution Explorer**, **Properties**, **Output**, **Document Outline**, **Error List** ș.a.m.d. Atenție, deși în Figura 1.3 fereastra centrală este de culoare albă, ceea ce vom vedea inițial va fi un spațiu de culoare albastru metalizat, ca în centrul Figurii 1.4.

În acest moment, putem introduce fișierele proiectului nostru, și anume `centru.c` și `headerulnostru.h`. Pentru aceasta, urmărind Figura 1.4, ne poziționăm cu cursorul pe elementul **Source Files** al listei din fereastra **Solution Explorer**. Apăsând butonul *drept* al mouse-ului, remarcăm apariția unei noi ferestre, din care vom selecta elementul **Add**. Acesta, odată selectat, va face vizibilă o altă fereastră, în cadrul căreia vom opta pentru elementul **New Item**.

Având cursorul poziționat pe elementul **New Item**, apăsăm butonul stâng al mouse-ului, ceea ce produce apariția ferestrei **Add New Item** `primul_program`, a se vedea Figura 1.5. Aici, selectăm din tabelul central elementul **C++ File( .cpp)**. Urmărind Figura 1.5, ne deplasăm în partea de jos a ferestrei active până la câmpul **Name**. Acesta va fi completat cu numele *întreg* al fișierului nostru, adică `program.c`. Atenție, dacă nu scriem și particula `.c`, fișierul va fi salvat ca `program.cpp`, adică un fișier cu cod C++. La urma urmei, compilatorul `cl.exe` care stă ascuns sub capota mediului de dezvoltare Visual Studio este un compilator



C/C++! Apoi apăsăm butonul Add.

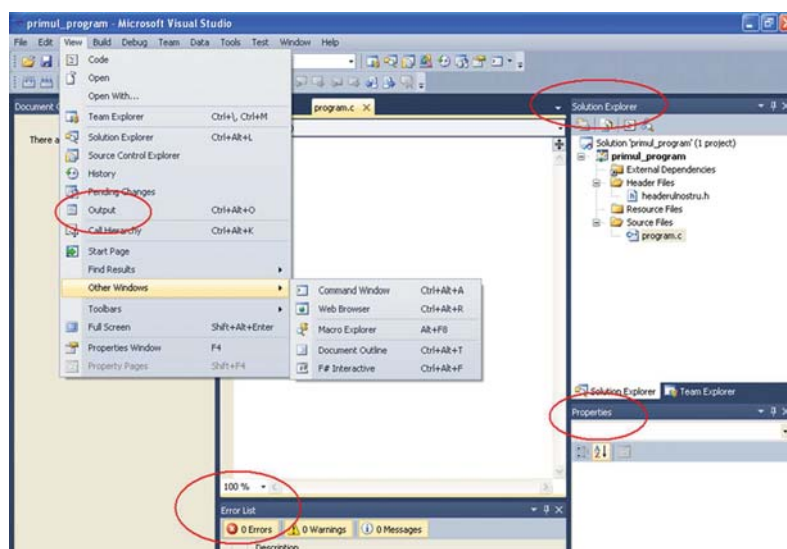


Figura 1.3 Ferestre de lucru în Visual Studio

Finalizarea cu succes a operației ne conduce la apariția elementului **program.c** în lista **Source Files** din fereastra **Solution Explorer**, a se vedea Figura 1.3. Reluăm procedeul, a se vedea Figura 1.4, plecând de la elementul **Header Files** din fereastra **Solution Explorer**, și alegem, urmărind Figura 1.5, elementul **Header File(.h)** din tabelul central. Odată încheiat acest pas, putem practic trece la scrierea codului... din primul nostru proiect Visual Studio.

Poziționând cursorul pe elementul **headerulnostru.h** din fereastra **Solution Explorer**, apăsăm butonul stâng al mouse-ului, ceea ce va produce apariția unei zone de culoare albă în centrul ecranului, a se vedea Figura 1.3. Înainte de a scrie primul nostru program — desigur, dacă acest detaliu nu a fost încă setat —, dorim să numerotăm liniile de cod. Ne va fi mai ușor astfel să depanăm (debug) programele C/C++. Pentru a activa numerotarea în cauză, apăsăm butonul **Tools** din meniul principal al Visual Studio-ului, ca în Figura 1.6. Din fereastra nouă apărută, selectăm elementul **Options**.

Apăsând butonul **Options**, producem apariția ferestrei **Options**, a se vedea Figura 1.7. Aici, din lista situată în partea stângă, în zona de culoare albă, alegem elementul **Text Editor**, ceea ce va permite vizualizarea unei noi liste de opțiuni. Dintre acestea, alegem elementul **C/C++**. În acest moment, urmărind Figura 1.7, putem selecta în zona **Display** din partea dreaptă a ferestrei active opțiunea **Line numbers**. Pentru a salva această alegere, apăsăm butonul **OK**.

Acum, revenind la câmpul central de culoare albă cu titlul **headerulnostru.h**, introducem următoarele linii de cod.

Un header “utilitar” în C: **headerulnostru.h**

```
1 #if !defined( HEADERULNOSTRU )
2 #define HEADERULNOSTRU
3 /*-----*/
4
```

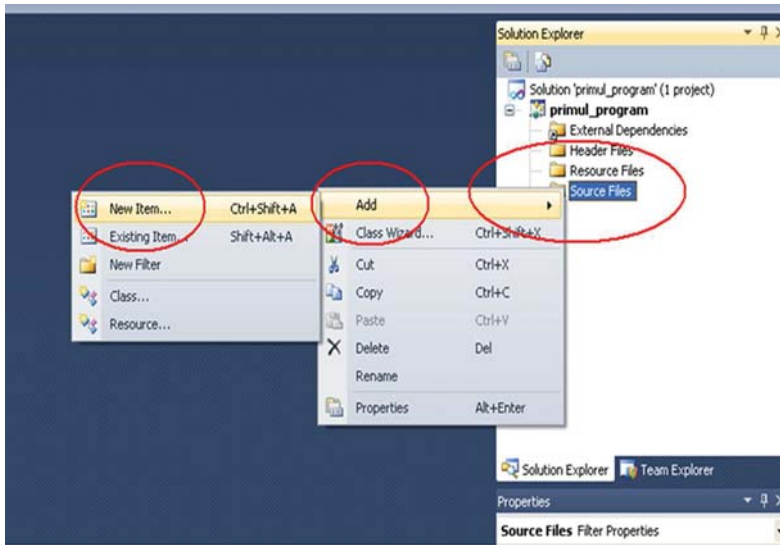


Figura 1.4 Adăugarea unei resurse noi la fișierele sursă ale proiectului

```

5 /*diverse headers:*/
6 #include <stdio.h>
7 #include <conio.h>
8 #include <stdlib.h>
9 #include <malloc.h>
10 #include <Windows.h>
11 #include <WinNT.h>
12
13 /* rutina de incheiere */
14 #undef LA_REVEDERE
15 #define LA_REVEDERE \
16     MessageBox(NULL, \
17         TEXT("\u00CEncheiem minunatul program?\t"), \
18         TEXT("Cutie de dialog"), \
19         MB.OK);
20
21 /*-----*/
22 #endif

```

În mod identic, adăugăm în câmpul `program.c` codul sursă al programului.

Primul nostru program în VS C: `program.c`

```

1 #include "headerulnostru.h"
2
3 main(){
4     printf("\nIncepem...\n");
5     LA_REVEDERE
6 }

```

Pentru instrucțiunile de *preprocesor*, marcate cu `#` în codul sursă, am urmat recomandările din [19, pg. 89, 90, 229]. Astfel, pentru a defini un *macro* pe mai multe linii se folosește