

PREZENTAREA GENERALĂ A MEDIULUI DE DEZVOLTARE

1. Mediul de dezvoltare Eclipse

La sfârșitul anilor '90, *IBM* a început dezvoltarea a ceea ce este cunoscut azi ca **Eclipse**. În noiembrie 2001, s-a decis să se adopte licențierea *open source* și modelul acestei tehnologii pentru a accelera adoptarea Eclipse ca platformă globală de dezvoltare. Principiul de bază al consorțiului a fost următorul: comunitățile *open-source* vor dirija programarea, iar consorțiul comercial va promova produsul respectiv. Aceasta era o nouă și interesantă aplicație a modelului *open source*, bazată pe o platformă gratuită și deschisă. Pentru mai multe informații accesați:

- www.eclipse.org
- <http://www.ibm.com/developerworks/rational/library/nov05/cernosek/>

Eclipse este mediul de dezvoltare integrat (**IDE – Integrated Development Environment**) pe care îl vom folosi pentru scrierea programelor C. El permite dezvoltarea de aplicații în mai multe limbaje de programare: Java, C, C++, C#. În cadrul laboratorului vom scrie programe C de tip consolă.

Pentru instalarea versiunii de lucru trebuie accesate următoarele link-uri:

a) Jave SE Runtime Environment (JRE)

b) Mediul de dezvoltare **Eclipse**

<http://eclipse.org/downloads>

Se selectează IDE (Integrated Development Environment) pentru **C**
Eclipse IDE for C/C++ Developers

c) Pentru cazul în care se lucrează pe sistemul de operare Windows, trebuie instalat compilatorul de **C (MinGW –** care este varianta pentru Windows a **gcc –** compilator și link editor pentru **C/C++**) și eventual debuggerul (programul de depanare) **gdb**. În cazul în care se lucrează pe Linux sau MacOS atât **gcc**, cât și **gdb** vin instalate odată cu sistemul de operare.

Pentru informații suplimentare referitoare la instalarea Eclipse se accesează pagina **Welcome** a mediului (Help->Welcome) se alege secțiunea **Tutorials**, apoi **C/C++ Development** și din panoul din stânga **C/C++ development User Guide -> Before you begin**.

2. Pornirea aplicației

În primul rând se face precizarea că explicațiile care urmează (în cazul în care nu se specifică altfel sunt valabile pentru toate cele trei sisteme de operare amintite mai sus (Windows, Linux, MacOS)

Pentru a lansa mediul de dezvoltare **Eclipse** se face dublu clic pe fișierul **eclipse** existent în directorul în care a fost instalată aplicația sau pe icoana corespunzătoare de pe desktop. Apare fereastra **Workspace Launcher** (prezentată în Figura 1). Această fereastră permite alegerea directorului în care vor fi stocate proiectele realizate. În exemplul din Figura 1 se stabilește ca director de lucru directorul c:\Temp. Faceți clic pe OK pentru a confirma alegerea directorului de lucru.

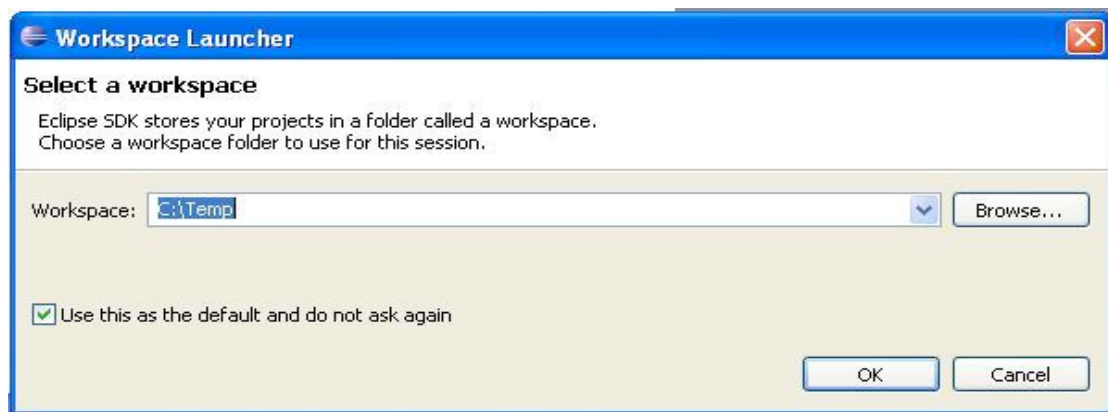


Figura 1. Fereastra Workspace Launcher.

După închiderea ferestrei **Workspace Launcher** apare fereastra Welcome (Figura 2) dacă mediul Eclipse este lansat pentru prima dată, sau fereastra mediului de dezvoltare (prezentată în Figura 3). Fereastra Welcome poate fi reafișată din Help->Welcome.

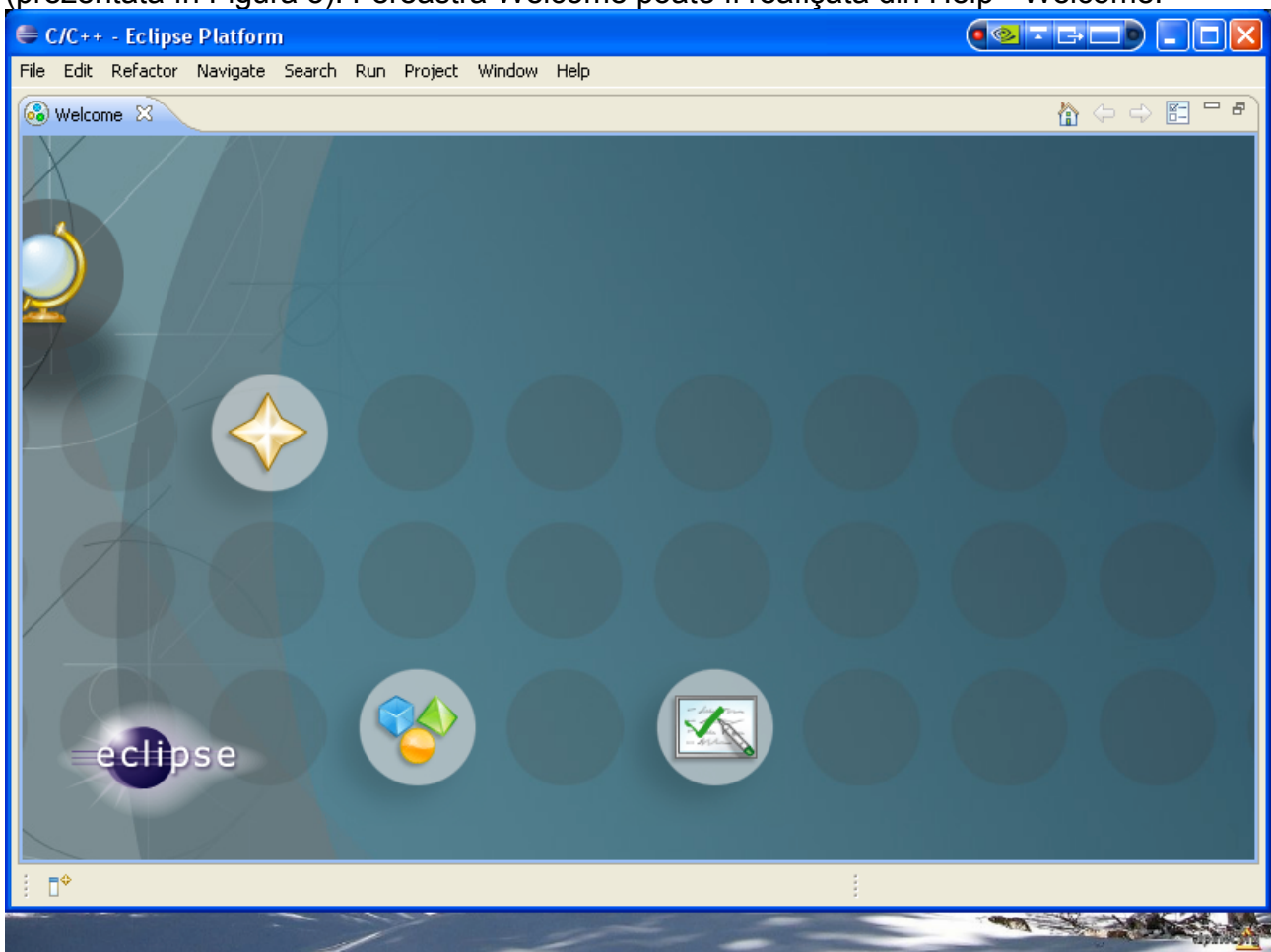


Figura 2. Fereastra Welcome a mediului Eclipse

Închiderea ferestrei Welcome are ca efect apariția ferestrei mediului de dezvoltare (Figura 3).

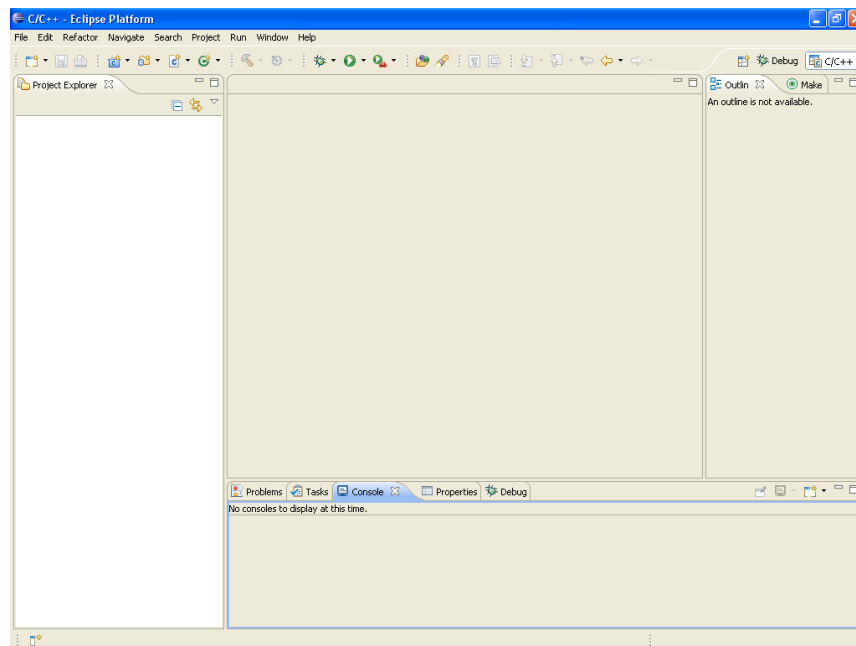


Figura 3. Fereastra mediului de dezvoltare

3. Crearea unui proiect

Pentru fiecare problemă pe care trebuie să o rezolvăm trebuie să creăm un proiect. Pentru a crea un proiect nou alegem File -> New -> C Project (Figura 4)

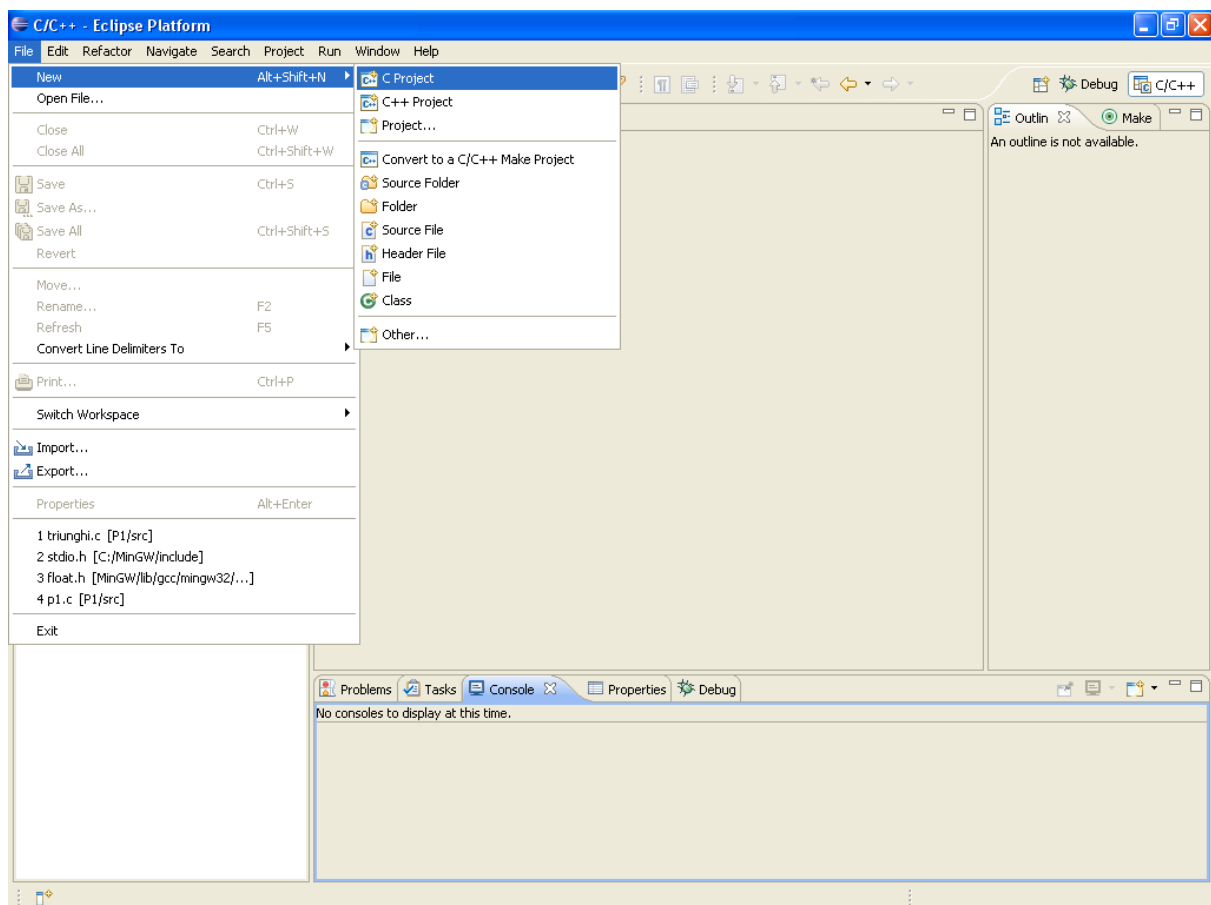


Figura 4. Deschiderea unui proiect C nou

Selectarea opțiunii **C Project** are ca efect afișarea ferestrei C New Project wizard, după cum este prezentat în Figura 5.

În câmpul **Project Name** se introduce numele proiectului (în Figura 5 este MyFirstProgram). Pe măsură ce se tastează numele programului, câmpul Directory devine c: \ Temp \ MyFirstProgram. Se alege tipul proiectului (din meniul **Project types**) ca fiind **Empty Project**. Se poate alege tipul de compilator pe care îl folosim din meniul **Toolchain**. În acest caz singurul compilator instalat este **MinGW GCC** după cum este prezentat în Figura 5 (care este compilatorul implicit în această instalare). Pentru Linux trebuie selectat **Linux GCC**, iar pentru **MacOSX GCC**.

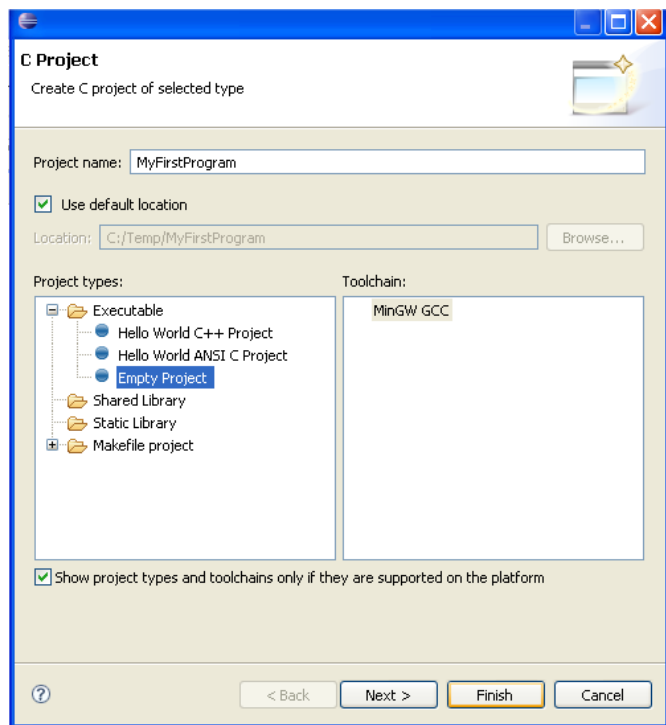


Figura 5: Deschiderea unui nou proiect cu numele **MyFirstProgram**

Se selectează **Finish** pentru a finaliza crearea proiectului C.

Se deschide proiectul în perspectivă **C**. O perspectivă definește setarea inițială a modului de vizualizare în fereastră. Perspectiva controlează ceea ce apare în anumite meniuri și bare de instrumente.

Proiectul este creat și deschis, ar trebui să vedem fereastra **Project explorer** – în partea stângă. De aici se pot crea sau deschide fișierele proiectului. Inițial proiectul conține (în mod implicit) fișierele header de bibliotecă disponibile (Figura 6).

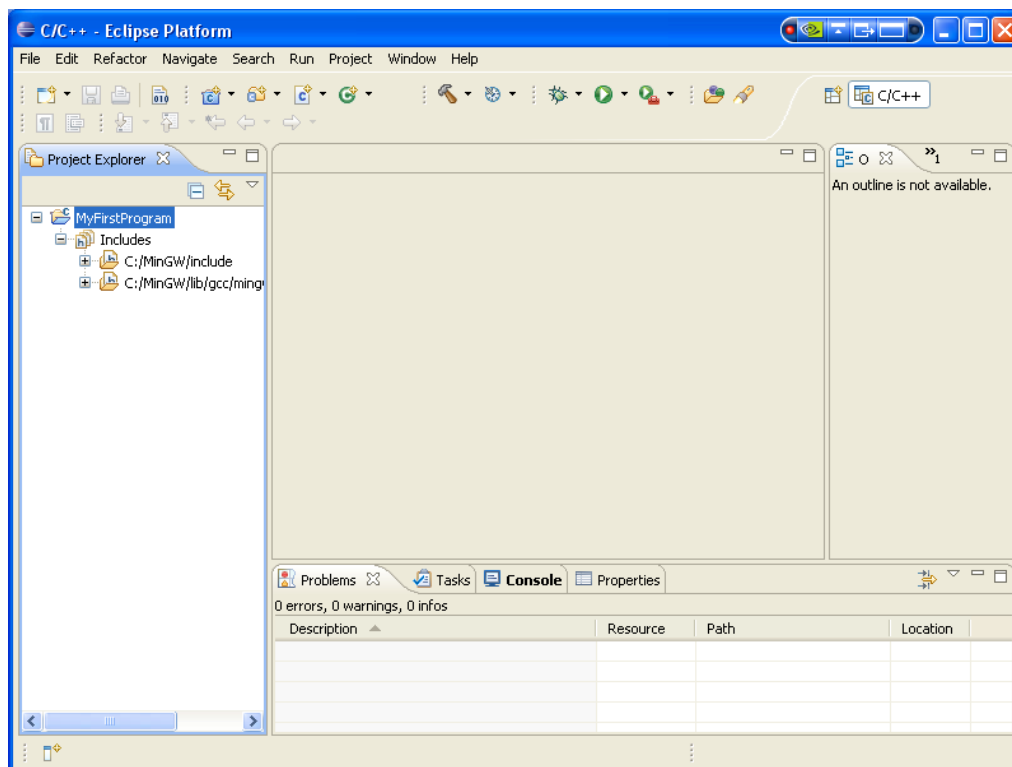


Figura 6. Fereastra inițială a proiectului

Putem stabili un stil de scriere pentru programele sursa din Window->Preferences

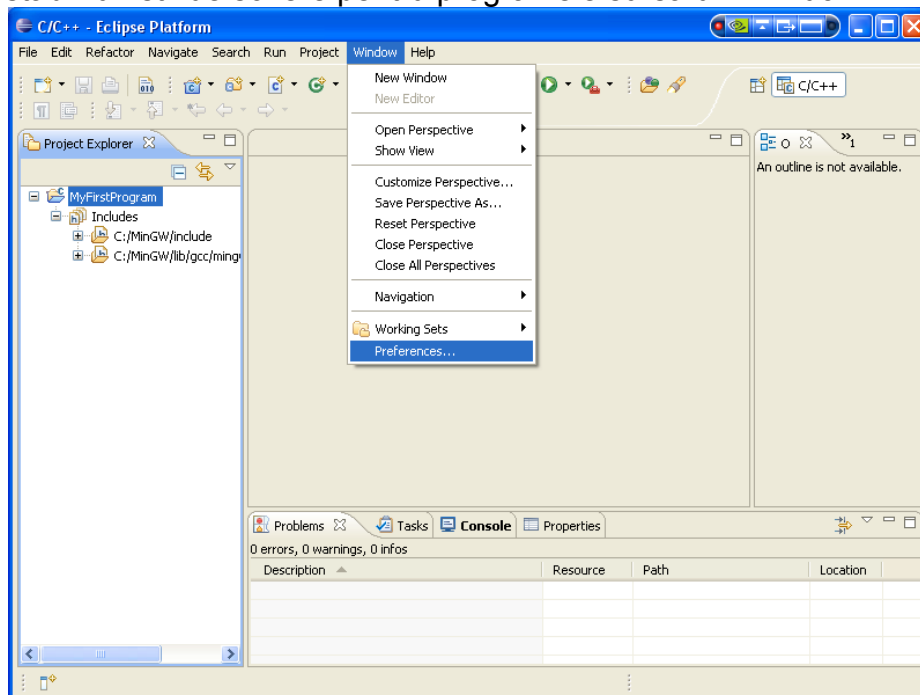


Figura 6a. Alegerea opțiunilor mediului

Pentru lucrările de laborator din acest semestru vom utiliza stilul BSD/Allman (built-in) (Figura 6b).

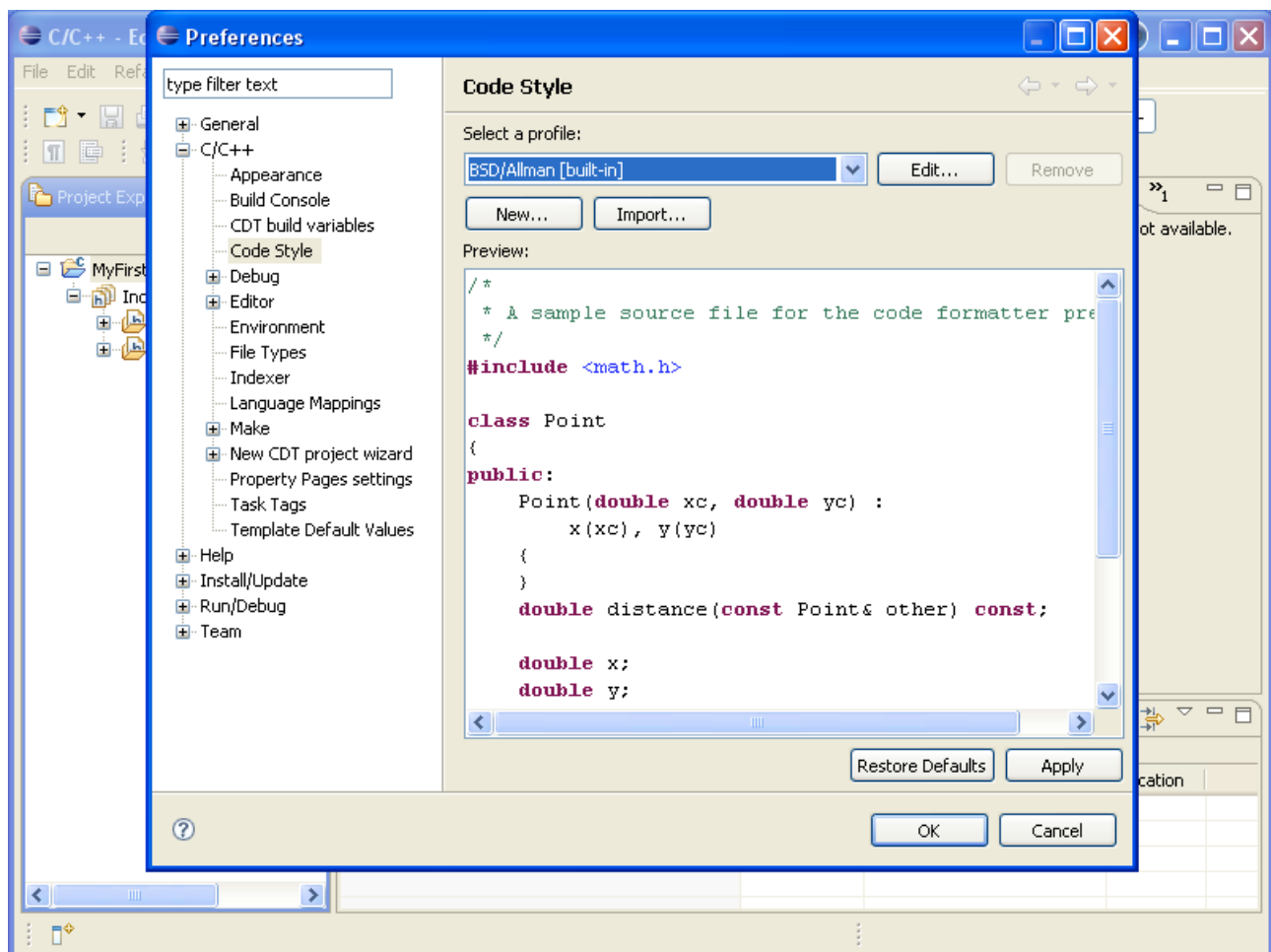


Figura 6b. Alegerea stilului de scriere a codului.

3.1. Adăugarea de fișiere la proiect

Pentru a putea scrie cel mai simplu program C în Eclipse, trebuie să adăugăm fișiere sursă / header. Pentru început vom adăuga directorul **src**, iar în acest director vom adăuga fișierele proiectului. Pentru crearea directorului **src** vom executa următorii pași

- Selectare folder **MyFirstProgram** în **Project Explorer**
- File / New / Folder
- Selectare **Folder name:** src

Acum vom adăuga următoarele fișiere în directorul src:

- FirstProgramMain.c fișierul care va conține funcția **main**
- MyFunction.h fișierul care va conține definiția funcțiilor
- MyFunction.c . fișierul care va conține implementarea funcțiilor

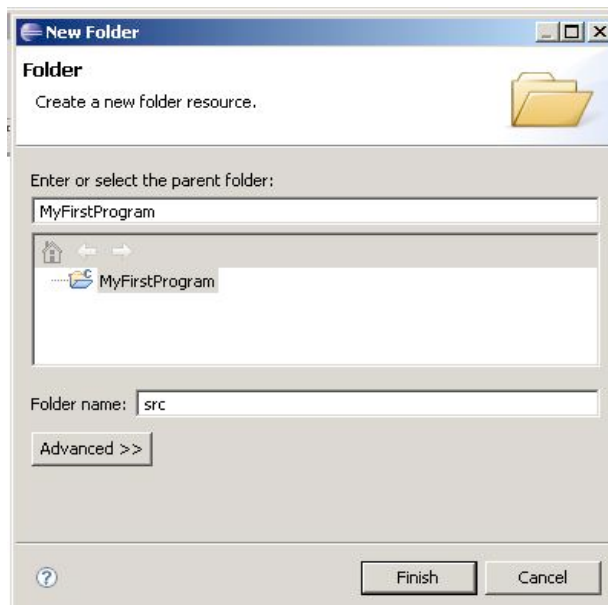


Figura 7: Creare director src.

Pentru adăugarea celor trei fișiere vom executa următorii pași:

1. Adăugare fișier header (Figura 8):

- Selectare folder **src** în **Project Explorer**
- File / New / HeaderFile
- Scrieți numele fișierului header **Header File:** MyFunction.h - figura 8.

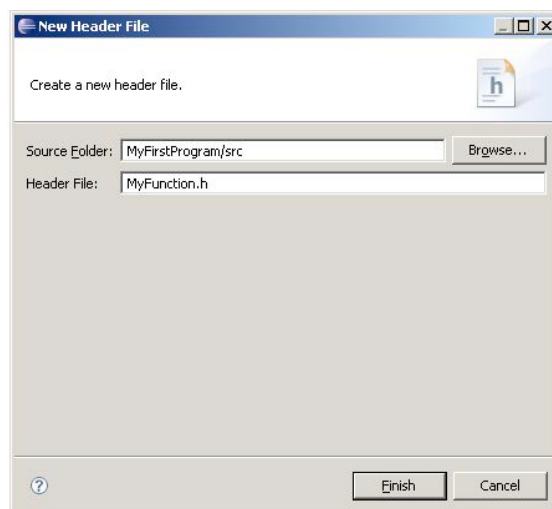


Figura 8: Adaugare fișier header.

2. Adăugare fișiere sursă (Figura 9)

- Selectare folder **src** în **Project Explorer**
- File / New / SourceFile
- Scrieți numele fișierului sursă **Source File:** MyFunction.c - figura 7
- File / New / SourceFile
- Scrieți numele fișierului sursă **Source File:** FirstProgramMain.c

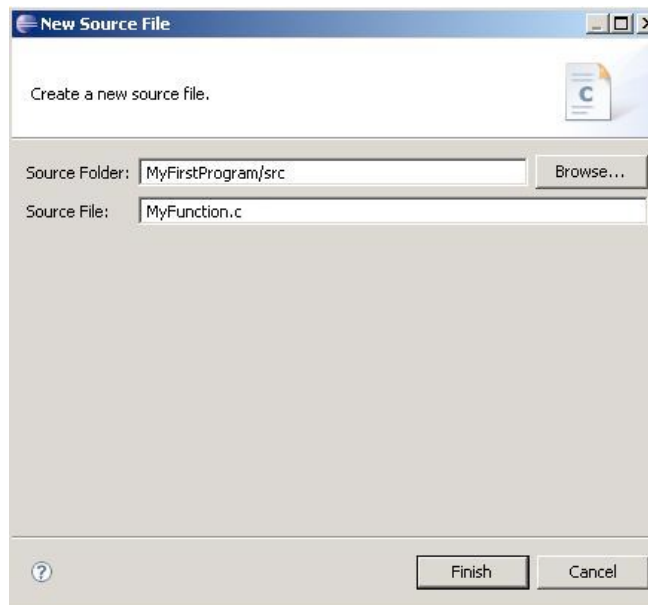


Figura 9. Adăugare fișier sursă.

Atenție: când adăugați fișiere header sau fișiere sursă este obligatoriu să precizați și extensia acestora.

3.2 Scrierea primului program

Să se scrie un program care citește de la tastatură trei numere întregi și afișează maximul lor.

Selectați din **Project Explorer** fișierul care vreți să-l editați (clic pe el), apoi scrieți codul C dorit:

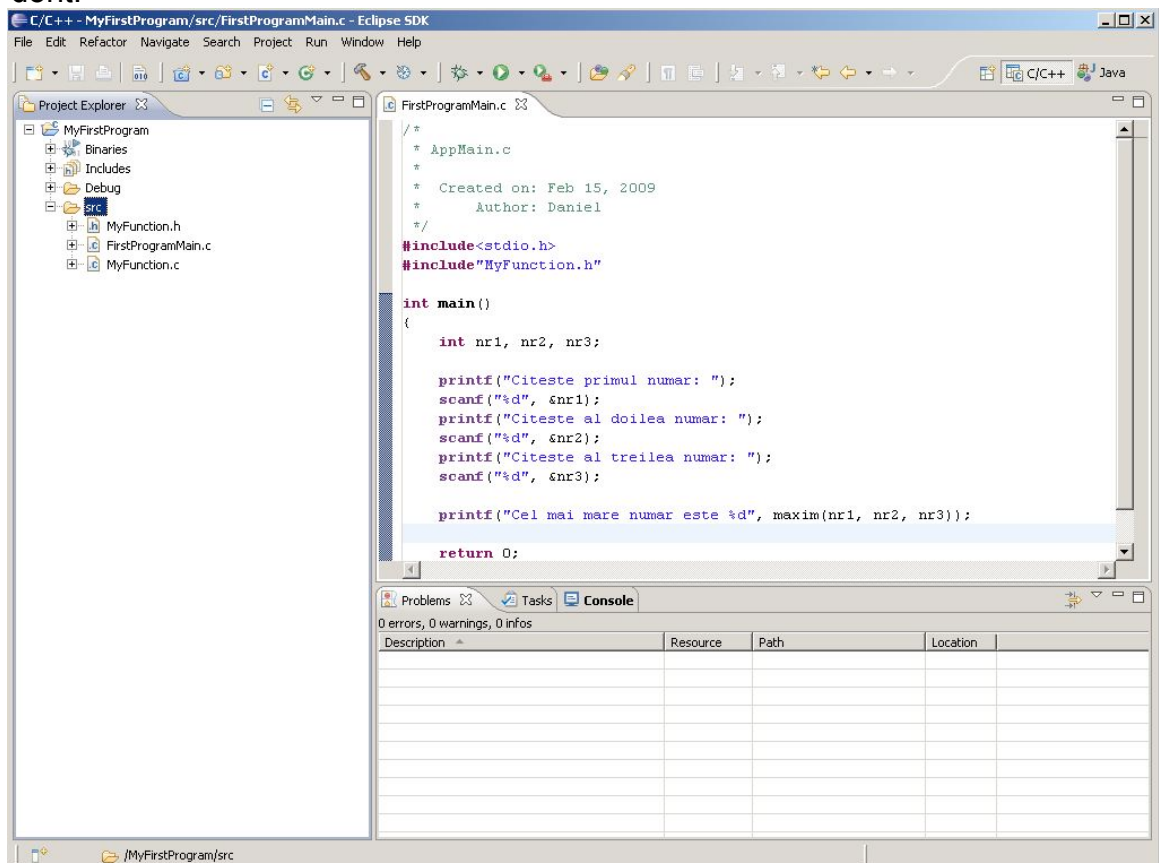


Figura 10. Eclipse IDE după adăugarea fișierelor proiectului

Mai jos este listat codul sursă din cele trei fișiere ale aplicației.

Observație: Fiecare student va completa cu numele său linia de comentariu care se referă la autorul programului.

```
/*
 * MyFunction.h - Proiect MyFirstProgram
 *
 * Creat pe 11.02.2009
 * Autor <nume autor program>
 * Copyright 2009
 *
 * Descriere Fisierul cu prototipurile functiilor
 * pentru primul proiect
 */

#ifndef MYFUNCTION_H_
#define MYFUNCTION_H_

int maxim(int x, int y, int z);

#endif /* MYFUNCTION_H_ */

/*
 * MyFunction.c - Proiect MyFirstProgram
 *
 * Creat pe 11.02.2009
 * Autor <nume autor program>
 * Copyright 2009
 *
 * Descriere: Functiile pentru primul proiect
 */
#include "MyFunction.h"

/*
 * maxim – Functie pentru determinarea maximului dintre numere
 * intregi
 * Parametri de intrare: trei numere intregi
 * Parametri de iesire: maximul dintre cele trei numere
 */

int maxim(int x, int y, int z)
{
    int nrMaxim;

    if (x > y)
    {
        nrMaxim = x;
    }
    else
    {
        nrMaxim = y;
    }
    if(nrMaxim < z)
    {
        nrMaxim = z;
    }
}
```

```
    }
    return nrMaxim;
}

/*
 * FirstProgramMain.c - Proiect MyFirstProgram
 *
 * Creat pe 11.02.2009
 * Autor <nume autor program>
 * Copyright 2009
 *
 * Descriere: Fisierul cu functia main pentru primul proiect
 *            de calculare a maximului dintre 3 numere
 */

#include <stdio.h>
#include "MyFunction.h"

int main(void)
{
    int nr1, nr2, nr3;
    int max;

    printf("Citeste primul numar: ");
    scanf("%d", &nr1);
    printf("Citeste al doilea numar: ");
    scanf("%d", &nr2);
    printf("Citeste al treilea numar: ");
    scanf("%d", &nr3);

    max = maxim(nr1, nr2, nr3);
    printf("Cel mai mare numar este %d\n", max);

    return 0;
}
```

3.3 Compilarea și rularea proiectului. Detectarea erorilor.

Pentru a compila proiectul selectăm din meniul principal **Project** → **Build Project**.

Atenție: ceea ce se compilează este codul de la ultima salvare. De aceea salvați întotdeauna înainte de compilare: **CTRL + S**.

Dacă sunt detectate erori de compilare, în partea de jos a ecranului va apărea fereastra **Problems**. Să introducem intenționat o eroare, de exemplu, putem înlocui linia "scanf("%d", &nr);" cu "scanf("%d", &nr)" scoțînd ";". La compilare, compilatorul își dă seama că lipsește ";" și afișează în fereastra **Problems** următorul mesaj:

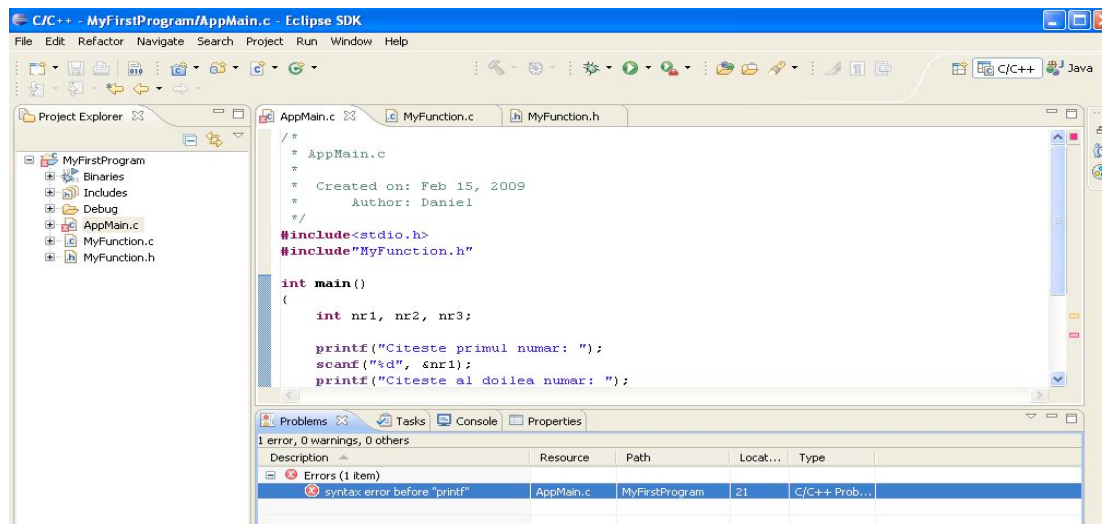


Figura 11. Compilare cu erori

Atenție: se observă "X" - ul roșu cu care este marcat atât fișierul, cât și linia unde a apărut eroarea.

Dacă dăm dublu-clic pe mesajul de eroare, cursorul se va deplasa la linia de cod conținând eroarea. În general s-a observat următoarea regulă – eroarea trebuie căutată ori pe linia afișată în error list, ori cu o linie mai sus. În exemplul nostru, mesajul de eroare ne duce cu o linie mai jos.

După compilare, link-editare și eliminarea tuturor erorilor lansarea în execuție se poate face (pentru Windows) în două moduri:

A. Folosind Command prompt din Windows (cmd.exe)

- pornim cmd.exe: Start /Run

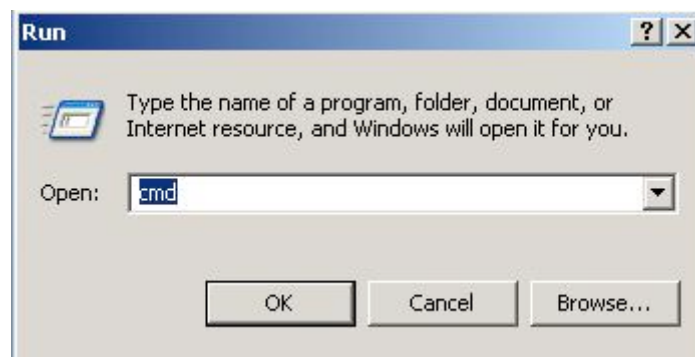


Figura 12. Rulare cmd.exe

- din linia de comandă a aplicației **cmd**, folosind comanda MS-DOS **cd** (change directory), setăm directorul în care se găsește programul executabil (fișierul de tip **exe**).

Alte comenzi MS-DOS utile:

type <nume_fisier> - listează pe monitor conținutul fișierului <nume_fisier>
dir – listează toate fișierele din directorul curent

- lansăm în execuție fișierul **exe** rezultat în urma link-editării tastând numele dat proiectului: MyFirstProgram

Observație: pentru fișierele de tip exe (executabile) nu este necesară indicarea extensiei atunci când sunt lansate în execuție.

B. Folosind Total Commander

- lansăm în execuție Total Commander și ne poziționăm pe subdirectorul Debug al directorului în care se găsește programul sursă (Figura 13).

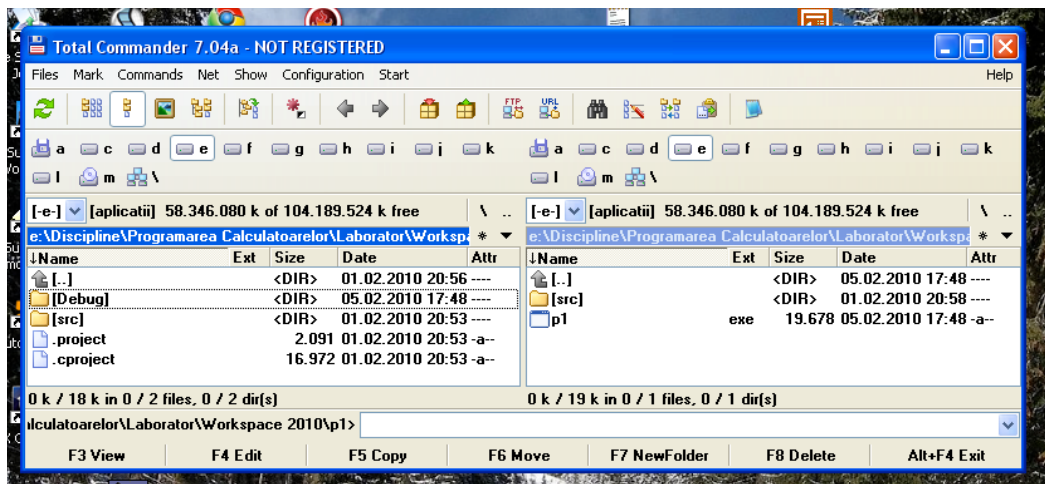


Figura 13. Fereastră Total Commander

- Selectăm din meniul Command → Run Dos (Figura 14).

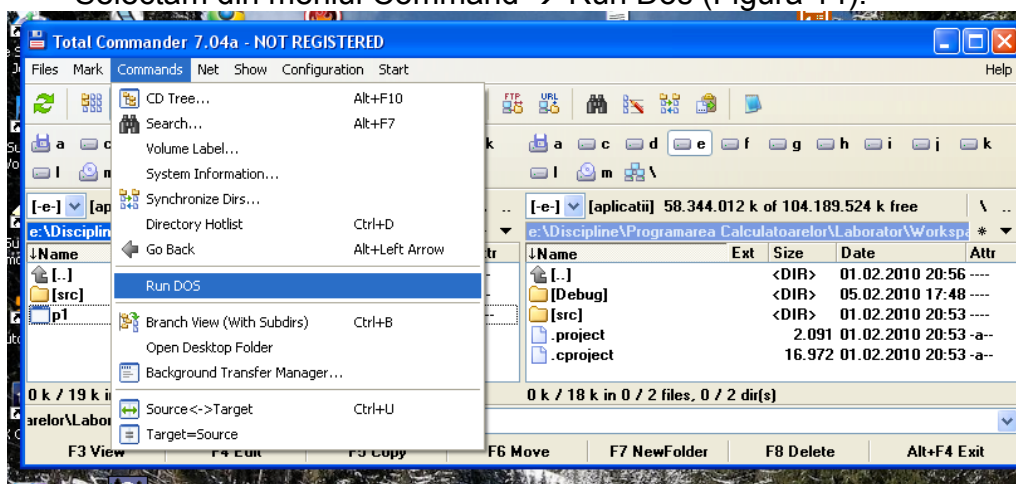


Figura 14. Lansarea în execuție a MS-DOS

- Apare fereastra MS-DOS din Figura 15.

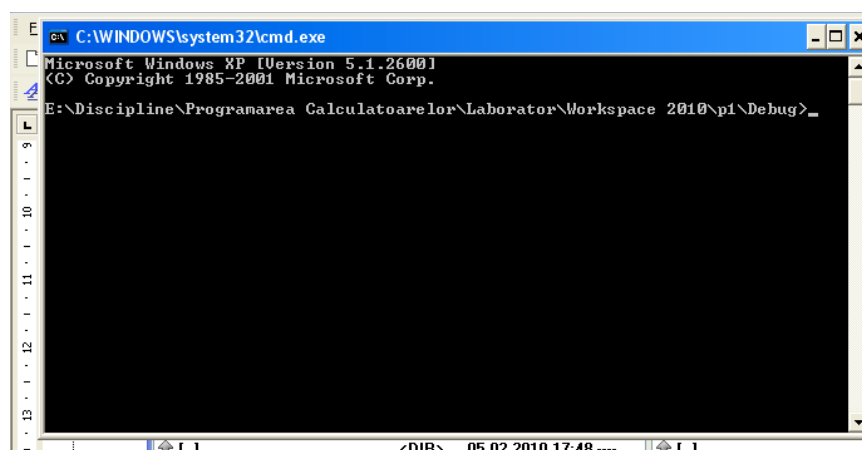


Figura 15. Fereastra MS-DOS

- Se lansează în execuție prin tastarea numelui programului (aici p1). Numele programului coincide întotdeauna cu numele proiectului. În Figura 16 este fereastra MS-DOS în care s-a executat programul **p1** (care cere numele utilizatorului și afișează mesajul **Primul program in C**)

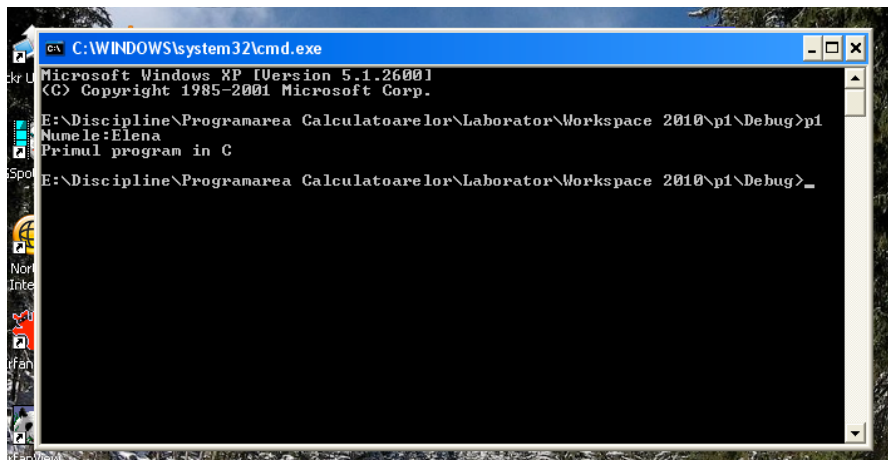


Figura 16. Lansarea în execuție și rularea programului **p1**

4. Deschiderea unui proiect existent

Pentru a încărca un proiect existent se vor executa următorii pași:

Selecția File / Import

Selecția General / Existing Projects ca în figura de mai jos. Apăsarea butonului Next.

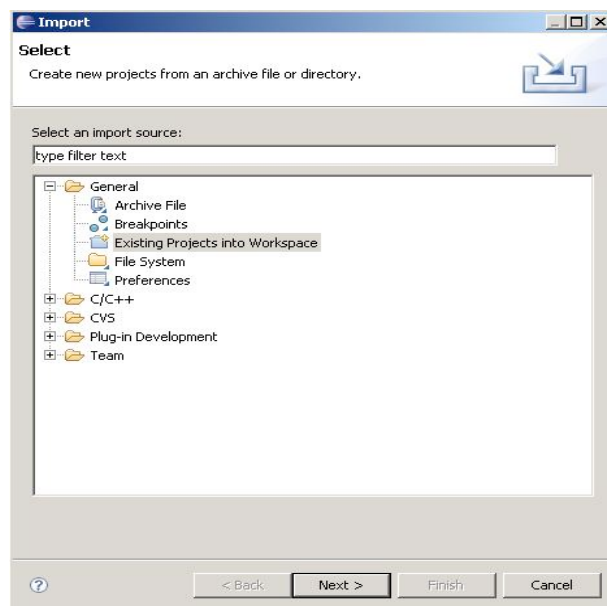


Figura 17. Deschiderea unui proiect existent

- Apăsarea butonului **Browse** și selecția directorului în care se află aplicația: C:\Temp\MyFirstProgram
- Bifarea proiectelor dorite din lista **Projects**. În cazul nostru avem doar un proiect în directorul specificat – Figura 18.

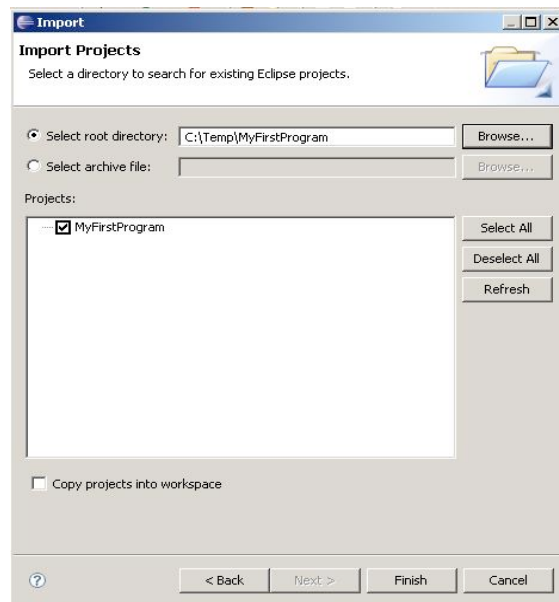


Figura 18. Selectare proiecte.

- Faceți clic pe **Finish** pentru a încărca proiectul.

5. Utilizarea debugger-ului

Debugger-ul din acest mediu integrat este puternic și ușor de folosit. Pentru a starta aplicația în modul debug:

- Clic Run / Debug (F11).
- În acest moment veți fi întrebați dacă doriți să deschideți perspectiva debug. O perspectivă definește setarea inițială a modului de vizualizare în fereastră. **Perspectiva Debug** conține moduri de vizualizare ce sunt folosite pentru depanare de programe (clic pe yes). În acest moment aplicația se va deschide în modul debug după cum este prezentat în Figura 19.:

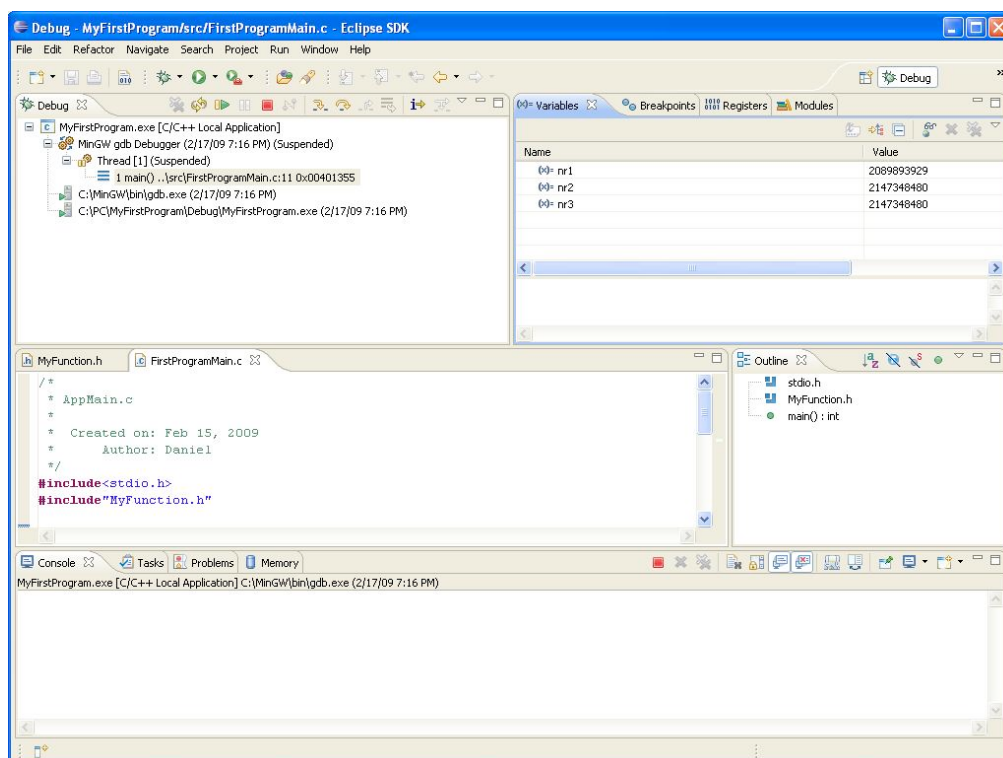


Figura 19. Perspectiva Debug.

- Acum putem trasa aplicația pas cu pas. Există câteva butoane pe care le vom utiliza pentru a depana proiectul nostru (meniul Run):
 - Butonul "Resume" - **F8** este folosit pentru a relua executarea până când se termină sau întâlnește un alt breakpoint.
 - Butonul "Stop" **CTRL-F2** va opri procesul de depanare.
 - Butonul "Step into" – **F5** va executa codul linie cu linie și va intra în metodele care vor fi apelate. Altfel, se va comporta ca butonul "Step over".
 - Butonul "Step over" – **F6** va executa codul linie cu linie fără a intra în metode.
- Apăsăți **F5** și observați linia la care a ajuns programul, linia selectată este linia care se execută când apăsați **F5**. Când ați ajuns pe instrucțiunea **scanf** (citește un număr de la tastatură) apăsați încă o dată **F5**, apoi introduceți valoarea primului număr în consolă ca în figura de mai jos:

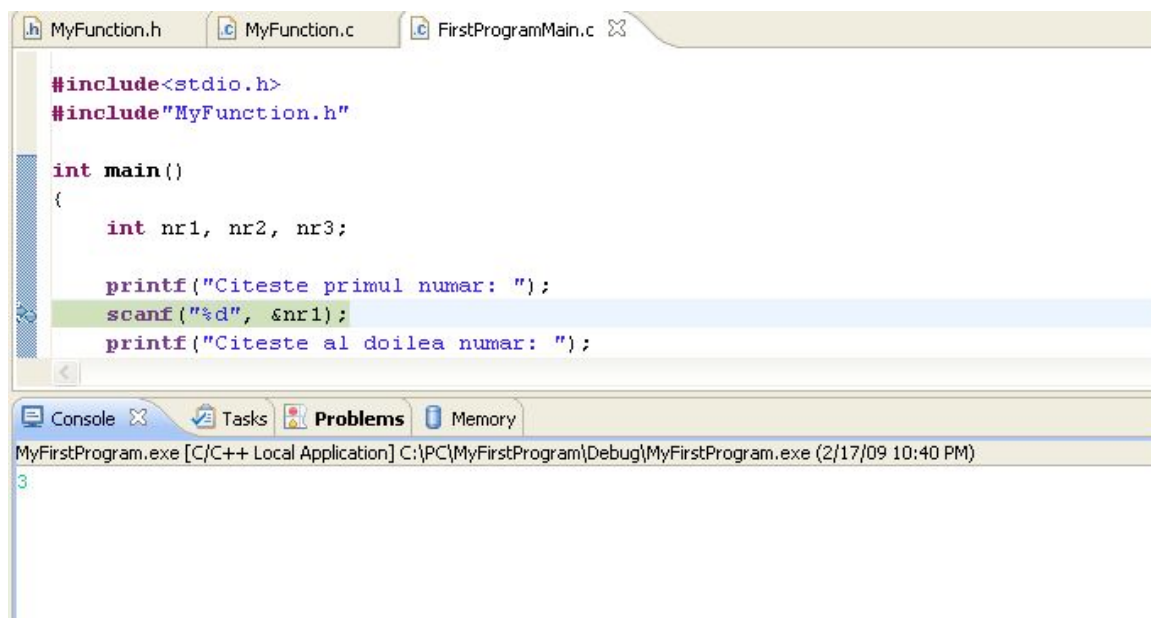


Figura 20. Debug pas cu pas.

- **Variable:** uitați-vă la cadrul dreapta sus la fereastra **Variables** aici vor apărea valorile variabilelor. Parcurgeți liniile de cod pas cu pas apăsând butonul **Step over** (F5 ca o comandă rapidă), uitați-vă în fereastra **Variables**. O dată ce o nouă variabilă este inițializată va apărea acolo și atunci când ea își schimbă valoarea, se va actualiza automat.
- Se vor introduce valorile nr1= 3, nr2 = 76, nr3 = 43. Observați că, apăsând F5 când linia selectată este: **printf("Cel mai mare numar este %d", maxim(nr1, nr2, nr3));** se va trasa codul funcției **maxim**, urmăriți ramurile de execuție ale algoritmului în cadrul acestei funcții.
- Puteți pune un breakpoint pe o anumită linie:
 - selectând linia respectivă - de exemplu: **printf("Citeste al doilea numar: ");**
 - apăsând **CTRL + SHIFT + B** sau meniul Run – Toggle Breakpoint acum aplicația se va opri la fiecare execuție în mod debug la linia ce conține breakpointul.

Fiecare problemă se va rezolva folosind un proiect care va conține:

- un fișier header cu prototipurile funcțiilor folosite
- un fișier cu funcția main
- un fișier cu funcțiile din program

(După modelul de la curs).

De asemenea, pentru fiecare problemă se va face o analiză pentru identificarea datelor de intrare și de ieșire din program și din fiecare funcție, a algoritmilor folosiți, cât și a seturilor de date de test).