



Iată ce vă așteaptă!

enjoy

Evoluția unui PROGRAMATOR

Vă prezentăm în cadrul acestui articol modul în care vor arăta programele voastre simple pe măsura trecerii timpului.

Elev de gimnaziu

```
10 PRINT "HELLO WORLD"
20 END
```

Elev de liceu

```
Program Hello;
Begin
    Writeln('Hello World')
End.
```

Student

```
(defun hello
  (print
    (cons 'Hello (list 'World))))
```

Programator începător

```
#include <stdio.h>
void main(void) {
    char *message[] = {"Hello ", "World"};
    int i;
    for(i = 0; i < 2; ++i)
        printf("%s", message[i]);
    printf("\n");
}
```

Programator ocazional

```
#include <iostream.h>
#include <string.h>

class string{
private:
    int size;
    char *ptr;
public:
    string() : size(0), ptr(new char('\0'))
    {}
```

```
string(const string &s) : size(s.size) {
    ptr = new char[size + 1];
    strcpy(ptr, s.ptr);
}
```

```
~string() {
    delete[] ptr;
}
```

```
friend ostream &operator <<(ostream &,
                             const string &);
string &operator =(const char *);
};
```

```
ostream &operator <<(ostream &stream,
                    const string &s) {
    return (stream << s.ptr);
}
```

```
string &string::operator=(const char *chrs) {
    if (this != &chrs) {
        delete[] ptr;
        size = strlen(chrs);
        ptr = new char[size + 1];
        strcpy(ptr, chrs);
    }
    return (*this);
}
```

```
int main() {
    string str;

    str = "Hello World";
    cout << str << endl;
    return (0);
}
```

Programator expert

```
[uuid(2573F8F4-CFEE-101A-9A9F-00AA00342820)]
```

```
library LHello{
    importlib("actimp.tlb");
    importlib("actexp.tlb");
```

```
#include "pshlo.idl"
```

```
[uuid(2573F8F5-CFEE-101A-9A9F-00AA00342820)]
```

```
cotype THello{
    interface IHello;
    interface IPersistFile;
};
```

```
[exe, uuid(2573F890-CFEE-101A-9A9F-00AA00342820)]
```

```
module CHelloLib{
    importheader(<windows.h>);
    importheader(<ole2.h>);
    importheader(<except.hxx>);
    importheader("pshlo.h");
    importheader("shlo.hxx");
    importheader("mycls.hxx");
```

```
importlib("actimp.tlb");
importlib("actexp.tlb");
importlib("thlo.tlb");
```

```
[uuid(2573F891-CFEE-101A-9A9F-00AA00342820), aggregatable]
```

```
coclass CHello{
    cotype THello;
};
```

```
#include "ipfix.hxx"
```

```
extern HANDLE hEvent;
```

```
class CHello: public CHelloBase{
public:
    IPFIX(CLSID_CHello);
    CHello(IUnknown *pUnk);
    ~CHello();
    HRESULT __stdcall PrintSz(LPWSTR
                             pwszString);
```

```
private:
    static int cObjRef;
```

```
};
```

```
#include <windows.h>
#include <ole2.h>
#include <stdio.h>
#include <stdlib.h>
#include "thlo.h"
#include "pshlo.h"
#include "shlo.hxx"
#include "mycls.hxx"
```

```
int CHello::cObjRef = 0;
```

```
CHello::CHello(IUnknown *pUnk) :
    CHelloBase(pUnk) {
    cObjRef++;
    return;
}
```

```
HRESULT __stdcall CHello::PrintSz(LPWSTR
                                   pwszString) {
    printf("%ws\n", pwszString);
    return (ResultFromCode(S_OK));
}
```

```
CHello::~~CHello(void) {
    cObjRef--;
    if (cObjRef == 0)
        PulseEvent(hEvent);
    return;
}
```

```
#include <windows.h>
#include <ole2.h>
#include "pshlo.h"
#include "shlo.hxx"
#include "mycls.hxx"
```

```
HANDLE hEvent;
```

```
int _cdecl main(int argc, char *argv[]) {
    ULONG ulRef;
    DWORD dwRegistration;
    CHelloCF *pCF = new CHelloCF();

    hEvent = CreateEvent(NULL, FALSE, FALSE,
                        NULL);

    CoInitializeEx(NULL,
                   COINIT_MULTITHREADED);
    CoRegisterClassObject(CLSID_CHello,
                          pCF, CLSCTX_LOCAL_SERVER, REGCLS_MULTIPLEUSE,
                          &dwRegistration);

    WaitForSingleObject(hEvent, INFINITE);
    CoRevokeClassObject(dwRegistration);
    ulRef = pCF->Release();
    CoUninitialize();
    return(0);
}
```

```
extern CLSID CLSID_CHello;
extern UUID LIBID_CHelloLib;
CLSID CLSID_CHello = {
    0x2573F891,
    0xCFEE,
    0x101A,
    {0x9A, 0x9F, 0x00, 0xAA, 0x00, 0x34, 0x28,
     0x20}
};
```

```
UUID LIBID_CHelloLib = {
    0x2573F890,
    0xCFEE,
    0x101A,
    {0x9A, 0x9F, 0x00, 0xAA, 0x00, 0x34, 0x28,
     0x20}
};
```



enjoy

```
#include <windows.h>
#include <ole2.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "pshlo.h"
#include "shlo.hxx"
#include "clsid.h"

int _cdecl main(int argc, char *argv[]){
    HRESULT hRslt;
    IHello *pHello;
    ULONG ulCnt;
    IMoniker *pmk;
    WCHAR wcsT[_MAX_PATH];
    WCHAR wcsPath[2*_MAX_PATH];

    wcsPath[0] = '\\0';
    wcsT[0] = '\\0';
    if(argc > 1){
        mbstowcs(wcsPath, argv[1],
                 strlen(argv[1]) + 1);
        wcsupr(wcsPath);
    }
    else{
        fprintf(stderr, "Object path must be
                    specified\n");
        return(1);
    }

    if(argc > 2)
        mbstowcs(wcsT, argv[2],
                 strlen(argv[2]) + 1);
    else
        wcscpy(wcsT, "Hello World");

    printf("Linking to object %ws\n",
           wcsPath);
    printf("Text String %ws\n", wcsT);
    hRslt = CoInitializeEx(NULL,
                          COINIT_MULTITHREADED);
    if(SUCCEEDED(hRslt)){
        hRslt = CreateFileMoniker(wcsPath,
                                  &pmk);

        if(SUCCEEDED(hRslt))
            hRslt = BindMoniker(pmk, 0,
                                IID_IHello, (void **)&pHello);

        if(SUCCEEDED(hRslt)){
            pHello->PrintSz(wcsT);
            Sleep(2000);
            ulCnt = pHello->Release();
        }
        else
            printf("Failure to connect, status:
                    %lx", hRslt);
        CoUninitialize();
    }
    return (0);
}
```

Hacker începător

```
#!/usr/local/bin/perl
$msg="Hello, world.\n";
if ($#ARGV >= 0){
    while(defined($arg=shift(@ARGV))){
        $outfilename = $arg;
        open(FILE, ">" . $outfilename) ||
            die "Can't write $arg: $!\n";
        print (FILE $msg);
        close(FILE) || die "Can't close $arg:
            $!\n";
    }
}
else{
    print ($msg);
}
```

Hacker expert

```
#include <stdio.h>
#define S "Hello, World\n"
main(){exit(printf(S) == strlen(S) ? 0 : 1);}
```

Hacker ocazional

```
% cc -o a.out ~/src/misc/hw/hw.c
% a.out
```

Hacker Guru

```
% cat
Hello, world.
^D
```

Manager începător

```
10 PRINT "HELLO WORLD"
20 END
```

Manager expert

```
mail -s "Hello, world." bob@b12
Bob, dacă te rog frumos, ai putea să-mi faci
un program care tipărește "Hello, world."?
Am nevoie de el mâine.
^D
```

Manager principal

```
% zmail jim
Am nevoie de un program "Hello, world." în
după-amiaza asta.
```

Director

```
% letter
letter: Command not found.
% mail
To: ^X ^F ^C
% help mail
help: Command not found.
% la naiba!
!: Event unrecognized
% logout
```