****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Calancea Catalin**

**MI-222**

**Raport**

**pentru lucrarea de laborator Nr.2**

***la cursul de “Programarea Orientată pe Obiect”***

Verificat:

lector universitar

Buldumac Oleg

**Chișinău – 20****23**

**CUPRINS**

[1.INTRODUCERE 3](#_Toc149568596)

[2.SARCINA 4](#_Toc149568597)

[3.CONCLUZIA 5](#_Toc149568598)

[4.WEBOGRAFIE 6](#_Toc149568599)

[5.ANEXĂ 7](#_Toc149568600)

# 1.INTRODUCERE

# În programarea orientată pe obiect, constructorul și destructorul reprezintă două concepte esențiale în gestionarea obiectelor. Constructorul este o funcție specială responsabilă de inițializarea unui obiect în momentul creării sale, în timp ce destructorul este utilizat pentru eliberarea resurselor și curățarea obiectului în momentul în care acesta nu mai este necesar. Aceste două elemente sunt cruciale pentru asigurarea corectitudinii și eficienței programelor.

# Această lucrare de laborator se concentrează pe studiul și înțelegerea principiilor de definire și utilizare a constructorilor și destructorilor în programarea orientată pe obiect. Vom explora modul în care aceste funcții speciale sunt utilizate pentru a crea și distruge obiecte în limbajele de programare, cu un accent deosebit pe C++ , dar principiile prezentate se aplică în mare măsură și altor limbaje.

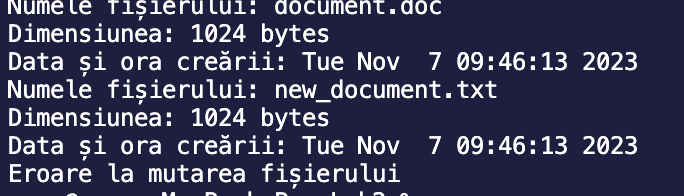
# Una dintre problemele majore în dezvoltarea software constă în gestionarea corectă a obiectelor și a resurselor alocate acestora. Erori de programare precum utilizarea obiectelor fără inițializare anterioară pot avea consecințe grave, cum ar fi scurgeri de memorie sau comportament nedefinit. Constructorii și destructorii reprezintă soluții esențiale pentru prevenirea acestor probleme și pentru asigurarea unei dezvoltări software fiabile și eficiente.

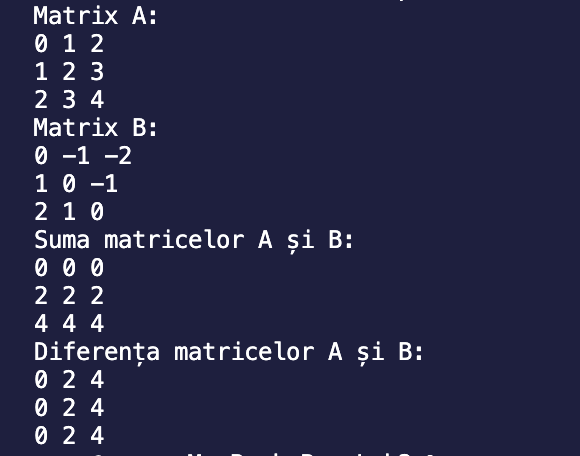
# Această lucrare reprezintă o introducere în conceptele esențiale legate de constructori și destructori, cu scopul de a oferi o bază solidă pentru dezvoltatorii de software în înțelegerea și utilizarea acestor funcții importante. În cele ce urmează, vom explora în detaliu fiecare aspect și vom furniza exemple practice pentru a ilustra conceptele prezentate.

# 2.SARCINA

а) Să se creeze clasa *File* – fişier, care conţine informaţia despre numele complet al fişierului şi anexa de asociere (doc – Word, psd – Photoshop, etc.), utilizînd memoria dinamică, mărimea, data şi timpul creării. Să se definească toti constructorii. Constructorul de schimbare a tipului are parametrul – numele de fişier. Să se definească funcţiile de redenumire a fişierului, de transfer în alt directoriu şi de modificare a anexei de asociere.

b) Să se creeze clasa *Matrix –* matrice. Clasa conţine pointer spre *long*, numărul de rînduri şi de coloane şi o variabilă – codul erorii. Să se definească constructorul fără parametri (constructorul implicit), constructorul cu un parametru – matrice pătrată şi constructorul cu doi parametri – matrice dreptunghiulară ş. a. Să se definească funcţiile membru de acces: returnarea şi setarea valorii elementului (i,j). Să se definească funcţiile de adunare şi scădere a două matrice; înmulţirea unei matrice cu alta; înmulţirea unei matrice cu un număr. Să se testeze funcţionarea clasei. În caz de insuficienţă de memorie, necorespondenţă a dimensiunilor matricelor, depăşire a limitei memoriei utilizate să se stabilească codul erorii.

a) 

b) 

# 3.CONCLUZIA

În cadrul acestei lucrări de laborator am studiat principiile definirii și utilizării constructorilor și destructorilor în programarea orientată pe obiect. Am înțeles importanța acestor concepte în dezvoltarea software-ului, în special în ceea ce privește gestionarea obiectelor și a resurselor alocate acestora.

Pe de altă parte, destructorii au un rol crucial în eliberarea resurselor și curățarea obiectelor în momentul în care acestea nu mai sunt necesare. Am discutat despre modul în care aceștia sunt declarați și folosiți, și am subliniat importanța eliberării resurselor pentru evitarea scurgerilor de memorie și a problemelor legate de performanță.

În concluzie, constructorii și destructorii reprezintă instrumente esențiale în dezvoltarea software și sunt fundamentale pentru asigurarea corectitudinii și eficienței programelor. Prin învățarea și aplicarea principiilor legate de aceste concepte, dezvoltatorii pot crea programe mai robuste și mai ușor de gestionat, contribuind la dezvoltarea unui software de calitate superioară.

# 4.BIBLIOGRAFIE

1. Stroustrup, B. (2013). "The C++ Programming Language," 4th Edition. Addison-Wesley Professional. - Această carte oferă o acoperire exhaustivă a limbajului de programare C++, inclusiv constructori și destructori.

2. Lippman, S. B., Lajoie, J., & Moo, B. E. (2012). "C++ Primer," 5th Edition. Addison-Wesley. - Cartea "C++ Primer" este o resursă excelentă pentru a înțelege conceptele fundamentale ale C++, inclusiv constructorii și destructorii.

3. Liberty, J., & Cadenhead, R. (2015). "Sams Teach Yourself C++ in One Hour a Day," 7th Edition. Sams Publishing. - Această carte oferă instrucțiuni pas cu pas pentru învățarea C++ și acoperă constructorii și destructorii într-un stil accesibil.

4. Horstmann, C. S. (2016). "Big C++," 2nd Edition. Wiley. - Acest manual acoperă programarea C++ în contextul dezvoltării de software și oferă informații despre constructori și destructori.

5. Deitel, P., Deitel, H., & Deitel, A. M. (2017). "C++ How to Program," 10th Edition. Pearson. - Această carte este un ghid detaliat pentru programarea C++ și include capitole dedicate constructorilor și destructorilor.

6. Strachan, J. (2013). "Beginning C# Object-Oriented Programming." Apress. - Dacă doriți să învățați despre constructori și destructori în limbajul C#, această carte este o sursă valoroasă.

7. Troelsen, A. (2012). "Pro C# 5.0 and the .NET 4.5 Framework." Apress. - Această carte oferă o prezentare detaliată a programării C# și acoperă utilizarea constructorilor și destructorilor în acest limbaj.

# 5.ANEXĂ

Codul sursa :

Sarcina a ) :

<https://github.com/KetSchnaider/Anul2/blob/main/POO/Lab2/Lab2a.cpp>

Sarcina b) :

<https://github.com/KetSchnaider/Anul2/blob/main/POO/Lab2/Lab2b.cpp>