**Proiect la PCLP II**

**Profesor Coordonator :**

Giorgian Neculoiu

Gabriela Olteanu

**Studenti :**

Plăeaşu Teodor Andrei

Moroşanu Florin Daniel

Cuprins

1. Introducere.............................................................................. 3 1.1 Motivarea alegerii temei ………………………………………………. 3

1.2 Despre sistemul de gestionare a sarcinilor ………………………………………………....... 3

2. Concepte și tehnologii folosite în elaborarea proiectului …...5

2.1 Visual Studio Code …………………………………………………… 5

2.2 C++ ……………………………………………………………………6

3. Scenariu de functionare ……………………………………..7

3.1 Scurta prezentare ……………………………………………………….7

3.2 Prezentare detaliata …………………………………………………….8

4. Concluzie …………………………………………………..15

5. Bibliografie …………………………………………………16

1. Introducere

1.1 Motivarea alegerii temei

Am ales să creăm un sistem de management al sarcinilor deoarece combină eficient aspectele organizaționale, interfața cu utilizatorul și logica aplicată în viața reală. Într-o lume în care eficiența și managementul timpului sunt din ce în ce mai importante, un astfel de sistem poate aduce un plus de productivitate atât în ​​mediul profesional, cât și în cel personal. Tema este practică și actuală, abordând o nevoie reală a utilizatorilor: de a-și planifica și urmări sarcinile într-un mod clar și structurat.

1.2 Despre Sistemul de gestionare a sarcinilor

Sistemele de gestionare a sarcinilor sunt esențiale pentru planificarea activității atât personale, cât și profesionale. Conceptul de organizare, monitorizare și prioritizare a muncii nu este nou, în ciuda a ceea ce ar putea părea a fi evoluții contemporane; a apărut la mijlocul secolului al XX-lea, când industria japoneză a îmbrățișat tehnici de planificare precum diagramele Gantt și abordarea Kanban.

Odată cu lansarea primelor programe software de management de proiect și digitalizarea proceselor industriale în anii 1990, modernizarea acestor sisteme a început să prindă contur. Utilizate inițial în cea mai mare parte de întreprinderile mici, aceste aplicații au devenit ușor accesibile publicului larg pe măsură ce dispozitivele mobile și internetul se dezvoltau. Scopul a fost întotdeauna același: creșterea productivității prin oferirea unui cadru clar pentru organizarea activităților.

1. Concepte și tehnologii folosite în elaborarea proiectului
   1. Visual Studio Code

A logo with blue letters

AI-generated content may be incorrect.

Visual Studio Code (prescurtat VS Code) este un editor de cod sursă dezvoltat de Microsoft, lansat oficial în anul 2015. Este gratuit, open-source și cross-platform, ceea ce înseamnă că poate fi utilizat pe Windows, macOS și Linux. VS Code s-a impus rapid ca unul dintre cele mai populare editoare de cod din lume datorită simplității sale, performanței excelente și ecosistemului extins de extensii.

Editorul oferă suport încorporat pentru un număr mare de limbaje de programare, cum ar fi JavaScript, Python, C++, Java, HTML/CSS și altele. În plus, prin extensii din marketplace, utilizatorii pot adăuga suport pentru tehnologii și limbaje suplimentare, pot integra debuggere, terminale, controlul versiunilor (Git) sau chiar medii de dezvoltare virtuale.

Visual Studio Code este folosit frecvent în mediul universitar, mai ales în cadrul facultăților cu profil informatic. Studenții îl utilizează pentru scrierea și testarea codului în diverse limbaje de programare, fiind preferat pentru interfața intuitivă, instalarea rapidă și compatibilitatea cu platforme educaționale. De asemenea, multe cursuri și laboratoare recomandă sau solicită folosirea acestui editor, datorită integrării sale eficiente cu GitHub, terminalul integrat și suportul pentru debugging, facilitând astfel procesul de învățare și dezvoltare.

2.2 C++



C++ este un limbaj de programare de nivel înalt, dezvoltat inițial de Bjarne Stroustrup la începutul anilor 1980, ca o extensie a limbajului C. A fost creat cu scopul de a oferi programatorilor un limbaj care să combine eficiența și controlul specific C-ului cu facilități moderne de programare orientată pe obiecte. De-a lungul timpului, C++ a evoluat semnificativ, ajungând să fie un limbaj puternic, flexibil și utilizat într-o gamă foarte largă de domenii, de la sisteme de operare și software embedded până la jocuri video, aplicații financiare și simulări științifice.

C++ oferă suport atât pentru programarea procedurală, cât și pentru cea orientată pe obiect, fiind astfel potrivit pentru proiecte de orice dimensiune. Limbajul permite un control detaliat asupra resurselor sistemului (memorie, procesor), ceea ce îl face foarte eficient din punct de vedere al performanței, dar presupune și o atenție sporită din partea programatorului. Standardele moderne ale limbajului, precum C++11, C++14, C++17 și C++20, au adus numeroase îmbunătățiri, inclusiv în ceea ce privește expresivitatea și siguranța codului.

În mediul universitar, C++ este adesea unul dintre primele limbaje predate în cadrul programelor de informatică sau inginerie. Este considerat ideal pentru învățarea conceptelor fundamentale ale programării, precum structuri de control, tipuri de date, pointeri, clase și moștenire. De asemenea, oferă studenților o înțelegere profundă asupra modului în care funcționează computerele la un nivel mai apropiat de hardware, în comparație cu alte limbaje de nivel mai înalt. Prin proiecte, laboratoare și examene practice, studenții deprind atât noțiuni teoretice, cât și abilități concrete de programare în C++.

3. Scenariu de functionare

3.1 Scurtă prezentare

Acest program reprezintă un sistem de gestionare a sarcinilor, care este atât simplu, cât și eficient, dezvoltat în C++. Aplicația dispune de o interfață bazată pe consolă ce le permite utilizatorilor să creeze, să vizualizeze și să marcheze sarcinile ca fiind completate. Fiecare sarcină cuprinde un titlu și o descriere. Informațiile sunt stocate într-un fișier text (sarcini.txt), permițând astfel păstrarea datelor între rulările programului.

Interacțiunea cu utilizatorul se face prin intermediul unui meniu simplu, în care acesta poate selecta una dintre acțiunile principale: adăugarea unei sarcini noi, afișarea listei de sarcini, marcarea unei sarcini ca finalizată sau ieșirea din program. Structura Sarcina este utilizată pentru a organiza datele aferente fiecărei intrări, iar sarcinile sunt gestionate prin intermediul unui vector dinamic.

Programul folosește concepte fundamentale ale programării, precum structuri, vectori, manipularea fișierelor, validarea inputului și gestionarea meniurilor. Deși simplu, acest sistem demonstrează principiile de bază ale unui manager de sarcini și poate fi extins cu ușurință pentru a include funcționalități mai complexe, precum ștergerea sarcinilor, filtrarea sau sortarea acestora după dată sau statut.

3.2 Prezentare detaliată

* Inițializarea programului și definirea structurii de dateA screen shot of a computer code

  AI-generated content may be incorrect.
* Salvarea datelor in fişier

A computer code on a black background

AI-generated content may be incorrect.

* Încărcarea datelor din fișier

A screen shot of a computer program

AI-generated content may be incorrect.

* Adăugarea unei noi sarcini

A screen shot of a computer code

AI-generated content may be incorrect.

* Afișarea sarcinilor existente

A computer screen with colorful text

AI-generated content may be incorrect.

* Marcarea unei sarcini ca finalizată

A computer screen shot of code

AI-generated content may be incorrect.

* Meniul principal și controlul programului

A screen shot of a computer program

AI-generated content may be incorrect.

4. Concluzie

Proiectul dezvoltat (implementarea unei aplicații de gestionare a sarcinilor în limbajul C++) a reprezentat un exercițiu practic complet de aplicare a conceptelor fundamentale de programare. Scopul principal a fost crearea unei aplicații simple și funcționale, operabilă din consolă, care să permită utilizatorului să adauge, vizualizeze și marcheze sarcini într-un mod intuitiv, eficient și organizat.

Pe parcursul dezvoltării, au fost utilizate structuri de date relevante (precum vectori de structuri) pentru stocarea informațiilor despre sarcini, manipularea textului și lucrul cu fișiere. De asemenea, au fost integrate tehnici fundamentale precum:

* citirea și validarea inputului din partea utilizatorului;
* salvarea persistentă a datelor într-un fișier;
* încărcarea și interpretarea structurată a acestora la pornirea programului;
* controlul fluxului aplicației printr-un meniu interactiv clar și accesibil.

Funcționalitățile cheie ale aplicației includ:

* Adăugarea de sarcini cu titlu, descriere și dată de finalizare;
* Afișarea organizată a tuturor sarcinilor, evidențiind cele completate și cele în lucru;
* Marcarea unei sarcini ca fiind finalizată;
* Persistența tuturor modificărilor prin salvare automată în fișier;
* Posibilitatea reluării lucrului cu aplicația fără pierderea datelor între rulări.

În concluzie, aplicația de gestionare a sarcinilor demonstrează nu doar aplicarea corectă a principiilor fundamentale ale limbajului C++, ci și capacitatea de a crea o soluție practică și funcțională pentru o problemă reală. Aceasta poate fi extinsă în viitor prin adăugarea de funcționalități suplimentare, precum editarea sau ștergerea sarcinilor, filtrarea după dată sau stare, sau integrarea unei interfețe grafice, constituind astfel o bază solidă pentru dezvoltarea unor proiecte mai complexe.

5. Bibliografie

<https://en.cppreference.com/w/>

<https://cplusplus.com/doc/tutorial/>

<https://cplusplus.com/reference/vector/vector/>

<https://www.geeksforgeeks.org/introduction-to-recursion-2/>

<https://cplusplus.com/reference/cstdlib/rand/>