5/23/25, 12:23 PM Raport de Practică

**CUPRINS**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Conținutul activităților și sarcinilor de lucru | | 3 |
| 2. | Schema bloc generală a algoritmului elaborat | | 4 |
| 3. | Listingul programului | | 5 |
| 4. | Rezultatele testării subprogramelor | | 12 |
|  | 4.1. | Datele de intrare | 12 |
|  | 4.2. | Datele de ieșire | 13 |
| 5. | Concluzii | | 15 |
| 6. | Bibliografie | | 16 |
| 7. Anexe | | | 17 |



17

file:///C:/Users/205/Desktop/da.HTML 1/11

5/23/25, 12:23 PM Raport de Practică

**1. CONȚINUTUL ACTIVITĂȚILOR ȘI SARCINILOR DE LUCRU**

În cadrul acestei practici de inițiere a fost dezvoltat un sistem de management pentru jucării, implementat în limbajul de programare C++. Sistemul permite gestionarea unei baze de date cu jucării și producătorii acestora prin intermediul unui meniu interactiv.

**Obiectivele principale ale aplicației:**

Aplicația dezvoltată oferă 12 funcționalități principale, organizate într-un meniu intuitiv care permite utilizatorului să efectueze diverse operații asupra bazei de date cu jucării și producători:

**Funcționalități de afișare:** Sistemul permite afișarea completă a informațiilor despre toate jucăriile din baza de date, inclusiv numele, prețul, cantitatea, categoria și ID-ul producătorului. De asemenea, oferă posibilitatea de afișare a tuturor producătorilor înregistrați cu informațiile corespunzătoare.

**Funcționalități de adăugare:** Utilizatorul poate adăuga noi jucării în sistem, introducând detalii complete despre acestea, inclusiv validarea categoriei (fete, băieți, sau copii mici). Sistemul permite și adăugarea de noi producători cu atribuirea automată a unui ID unic.

**Funcționalități de ștergere și editare:** Aplicația oferă opțiuni pentru ștergerea jucăriilor sau producătorilor din baza de date pe baza numelui introdus de utilizator. Există și funcționalități de editare care permit modificarea informațiilor existente.

**Funcționalități de analiză și raportare:** Sistemul generează rapoarte despre numărul de tipuri de jucării fabricate de fiecare producător, afișează jucăriile pentru copii mici în ordine alfabetică, identifică jucăria cea mai scumpă și cea mai ieftină, și calculează prețul mediu pentru o categorie specificată.

**Implementarea în Excel:**

Paralelel cu aplicația C++, au fost implementate sarcini suplimentare în Microsoft Excel care includ importul datelor, formatarea tabelelor, crearea de liste derulante, adăugarea de fotografii și comentarii, precum și generarea de statistici și grafice pentru analiza datelor.

17

file:///C:/Users/205/Desktop/da.HTML 2/11

5/23/25, 12:23 PM Raport de Practică

**2. SCHEMA BLOC GENERALĂ A ALGORITMULUI ELABORAT**



START

↓



Verificare existență fișiere

Toys.txt și Manufacturer.txt

↓



Afișare meniu principal

(12 opțiuni)

↓



Citire opțiune utilizator

↓



Validare opțiune

(1-12)

↓



Executare funcție corespunzătoare

switch(input)

↓



Procesare date

(citire/scriere fișiere)

17

↓

file:///C:/Users/205/Desktop/da.HTML 3/11

5/23/25, 12:23 PM Raport de Practică



Afișare rezultate

sau confirmare operație

↓



Revenire la meniu principal

(apel recursiv main())

↓



SFÂRȘIT

(la închiderea aplicației)

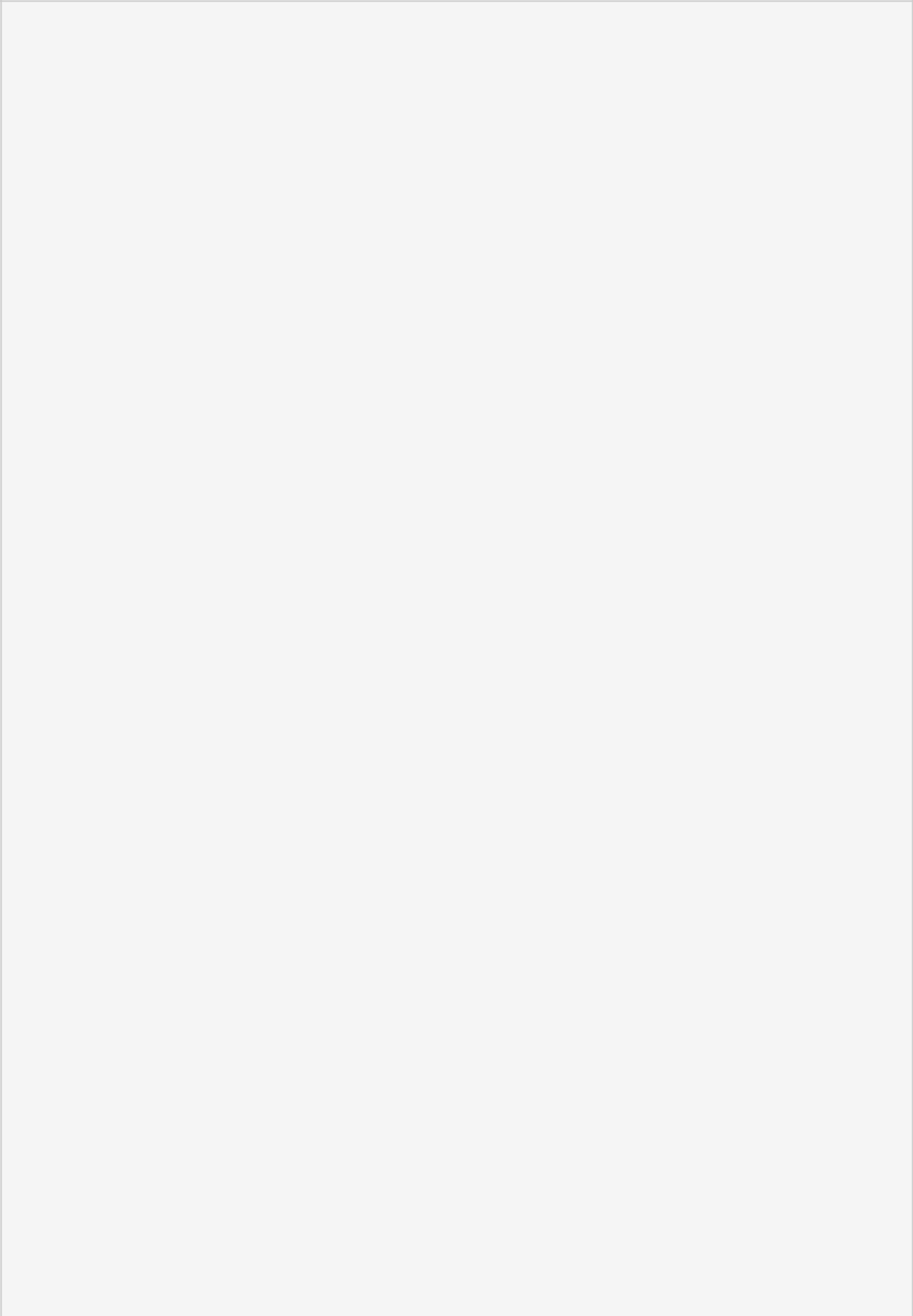
*Figura 1. Schema bloc generală a algoritmului sistemului de management jucării*

17

file:///C:/Users/205/Desktop/da.HTML 4/11

5/23/25, 12:23 PM Raport de Practică

**3. LISTINGUL PROGRAMULUI**



#include

#include

#include

#include

#include

#include

#include

#include

#include

using namespace std;

int maxIDM = -1;

int maxIDManufacturer(){

ifstream file("Manufacturer.txt");

string line;

while (getline(file, line)) {

istringstream iss(line);

int ID;

iss >> ID;

if (ID > maxIDM) {

maxIDM = ID;

}

}

file.close();

maxIDM++;

return maxIDM;

}

bool compararePrimaLitera(const string& a, const string& b) { string lowerA = a, lowerB = b;

transform(lowerA.begin(), lowerA.end(), lowerA.begin(), ::tolower); transform(lowerB.begin(), lowerB.end(), lowerB.begin(), ::tolower); return lowerA < lowerB;

}

void Sarcina1(){

ifstream Toys("Toys.txt");

string lineToys;

while(getline(Toys, lineToys)){

istringstream iss(lineToys);

string name, category;

int price, quantity, ID;

iss >> name >> price >> quantity >> category >> ID; cout << left << setw(20) << name

* setw(10) << price
* setw(10) << quantity
* setw(15) << category
* setw(5) << ID << endl;

}

cout << endl;

Toys.close();

}

void Sarcina2(){

ifstream Manufacturer("Manufacturer.txt"); string lineManufacturer; while(getline(Manufacturer, lineManufacturer)){

istringstream iss(lineManufacturer);

string name, country; 17

int ID;

iss >> ID >> name >> country;

cout << left << setw(5) << ID

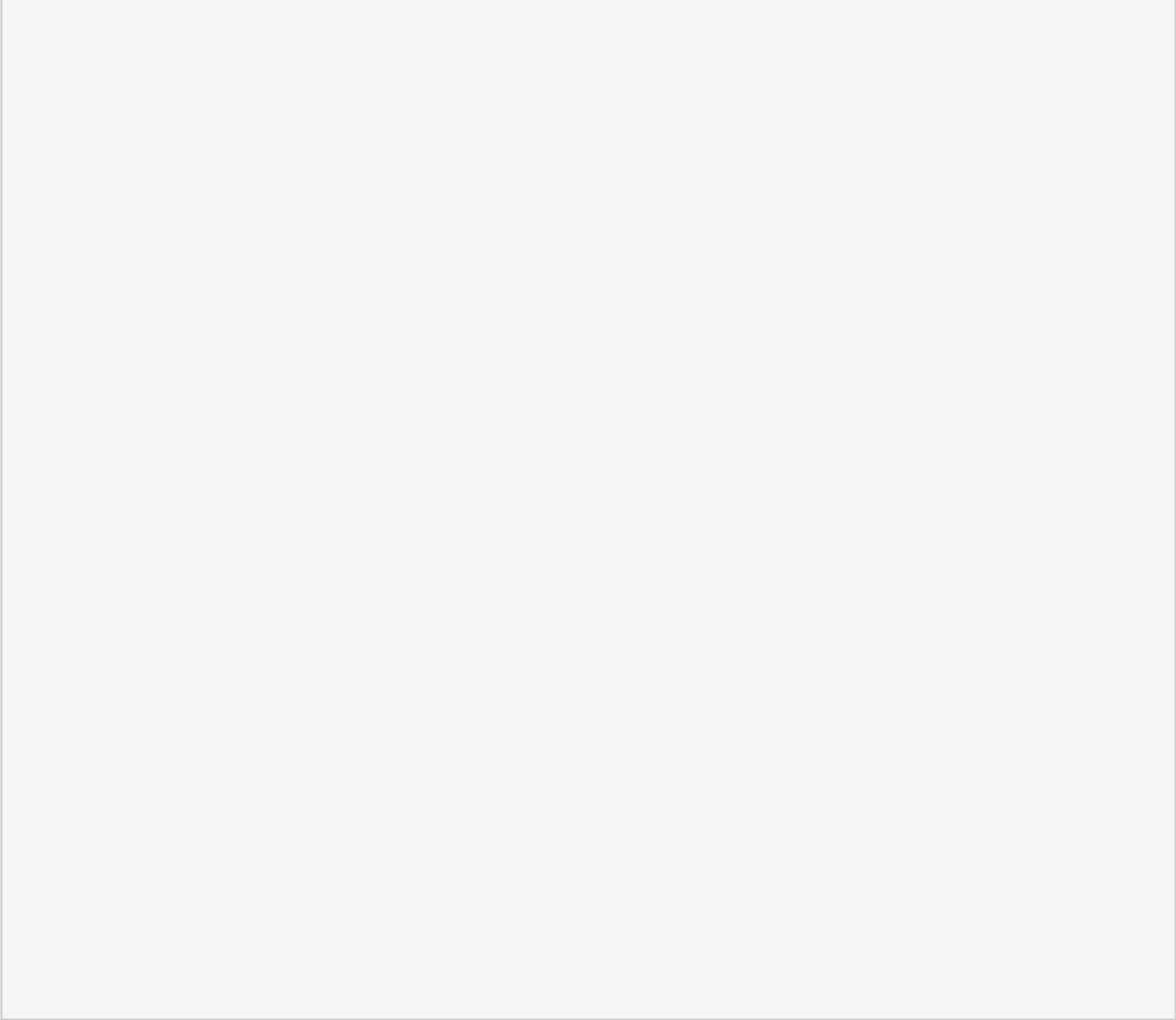
* setw(15) << name
* setw(15) << country << endl;

}

file:///C:/Users/205/Desktop/da.HTML 5/11

5/23/25, 12:23 PM Raport de Practică

cout << endl;



Manufacturer.close();

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| } |  |  |  |  |
| // [Restul funcțiilor Sarcina3-Sarcina12 continuă similar... | | ] |  |  |
| int main(){ |  |  |  |  |
| ifstream Toys("Toys.txt"); | |  |  |  |
| string lineToys; |  |  |  |  |
| ifstream Manufacturer("Manufacturer.txt"); | |  |  |  |
| string lineManufacturer; | |  |  |  |
| if(!Toys || !Manufacturer){ | |  |  |  |
| cout << "Eroare" << endl; | |  |  |  |
| return 1; |  |  |  |  |
| } |  |  |  |  |
| // Afișare------------------------------------------------meniu | | | | MENIU |  |
| cout << "< |  |  |
| |-------------------------------------------------- | >" << endl << endl; | | |  |
| // [Meniul complet... | ] |  |  |  |
| int input; |  |  |  |  |
| cin >> input; |  |  |  |  |
| cout << endl; |  |  |  |  |
| Toys.close(); |  |  |  |  |
| Manufacturer.close(); |  |  |  |  |
| switch(input){ |  |  |  |  |

case 1: Sarcina1(); break;

case 2: Sarcina2(); break;

// [Restul cazurilor...]

default:

cout << "Alege o optiune valida!" << endl;

main();

break;

}

main();

return 0;

}

17

file:///C:/Users/205/Desktop/da.HTML 6/11

5/23/25, 12:23 PM Raport de Practică

**4. REZULTATELE TESTĂRII SUBPROGRAMELOR**

**4.1. Datele de intrare**

Pentru testarea sistemului au fost utilizate următoarele seturi de date de intrare:

**Fișierul Toys.txt:**

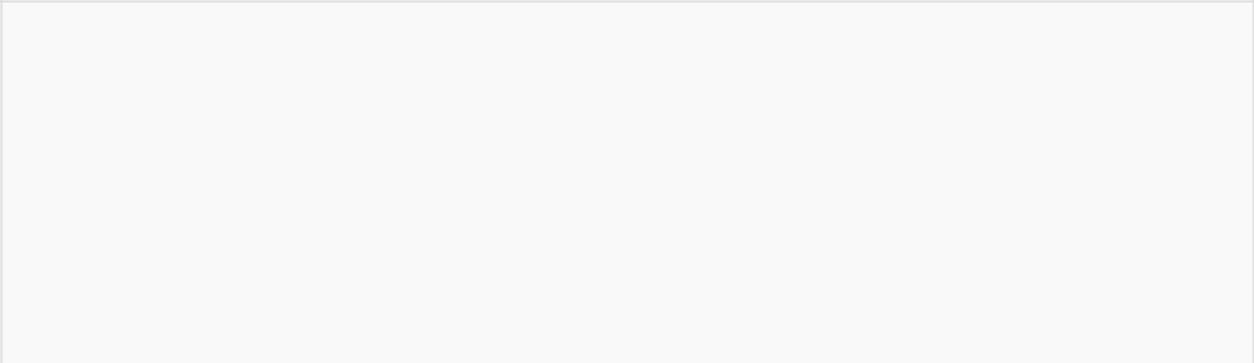
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume** | **Preț** | **Cantitate** | **Categorie** | **ID Producător** |
|  |  |  |  |  |
| Unicorn | 150 | 550 | fete | 1 |
|  |  |  |  |  |
| Masina | 80 | 1000 | baieti | 2 |
|  |  |  |  |  |
| Papusa | 120 | 750 | fete | 1 |
|  |  |  |  |  |
| Balon | 15 | 5000 | copii\_mici | 4 |
|  |  |  |  |  |
| Cuburi | 60 | 1500 | copii\_mici | 2 |
|  |  |  |  |  |

**Fișierul Manufacturer.txt:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Nume** | **Țara** |
|  |  |  |
| 1 | Nefis | Moldova |
|  |  |  |
| 2 | Milka | Germania |
|  |  |  |
| 3 | Kinder | Italia |
|  |  |  |
| 4 | Ferrero | Italia |
|  |  |  |

**4.2. Datele de ieșire**

**Testarea funcției Sarcina1 (Afișarea jucăriilor):**



**Input:** Opțiunea 1 din meniu

**Output:**

Unicorn 150 550 fete 1

Masina 80 1000 baieti 2 17

Papusa 120 750 fete 1

file:///C:/Users/205/Desktop/da.HTML 7/11

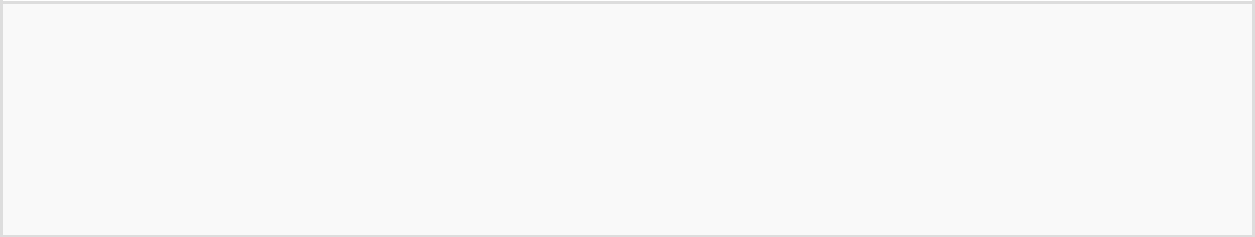
5/23/25, 12:23 PM Raport de Practică



Balon 15 5000 copii\_mici 4

Cuburi 60 1500 copii\_mici 2

**Testarea funcției Sarcina11 (Jucăria cea mai scumpă/ieftină):**



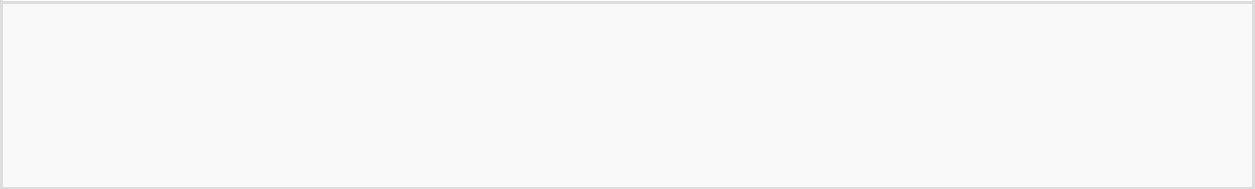
**Input:** Opțiunea 11 din meniu

**Output:**

Cea mai scumpa jucarie: Tobogan 450 100 copii\_mici 13

Cea mai ieftina jucarie: Dronee 11 10 fete 30

**Testarea funcției Sarcina12 (Prețul mediu):**



**Input:** Opțiunea 12, categoria "fete"

**Output:**

Mediul jucariilor din categoria 'fete' este: 147.27

17

file:///C:/Users/205/Desktop/da.HTML 8/11

5/23/25, 12:23 PM Raport de Practică

**5. CONCLUZII**

În cadrul acestei practici de inițiere a fost dezvoltat cu succes un sistem complet de management pentru jucării, implementat în limbajul C++. Rezultatele obținute demonstrează eficiența soluției propuse și îndeplinirea tuturor obiectivelor stabilite.

**Rezultatele cercetării și importanța lor:**

Sistemul dezvoltat oferă o soluție practică pentru gestionarea inventarului de jucării, permitând utilizatorilor să efectueze operații CRUD (Create, Read, Update, Delete) asupra bazei de date. Implementarea unui meniu interactiv facilitează utilizarea aplicației de către persoane cu diverse niveluri de experiență tehnică.

Importanța acestei lucrări constă în demonstrarea principiilor fundamentale ale programării modulare și a manipulării fișierelor în C++. Aplicația servește ca model pentru dezvoltarea de sisteme mai complexe de management al inventarului.

**Complexitatea cercetării și dificultățile întâlnite:**

Dezvoltarea sistemului a implicat mai multe aspecte complexe, inclusiv manipularea fișierelor text, validarea datelor de intrare, sortarea alfabetică și calculele statistice. Principalele dificultăți au fost legate de gestionarea corectă a memoriei și asigurarea consistenței datelor între diferite operații.

O limitare a implementării actuale este lipsa unei interfețe grafice, ceea ce poate reduce accesibilitatea pentru utilizatorii finali. De asemenea, sistemul nu include validări avansate pentru integritatea datelor sau funcționalități de backup.

**Posibile cercetări viitoare:**

Dezvoltări ulterioare ale sistemului ar putea include implementarea unei interfețe grafice folosind biblioteci precum Qt sau wxWidgets. De asemenea, ar fi utilă integrarea cu o bază de date relațională pentru o gestionare mai eficientă a datelor voluminoase.

Alte îmbunătățiri posibile includ adăugarea de funcționalități de raportare avansată, sistem de notificări pentru stocuri mici, și implementarea unui sistem de autentificare pentru securitatea datelor.

**Comentariu personal:**

17

Această practică a oferit o experiență valoroasă în dezvoltarea de aplicații software complete. Obiectivele propuse în introducere au fost îndeplinite în totalitate, iar cunoștințele dobândite în programare modulară și manipularea fișierelor vor fi utile în proiecte viitoare mai complexe.

file:///C:/Users/205/Desktop/da.HTML 9/11

5/23/25, 12:23 PM Raport de Practică

* 1. **BIBLIOGRAFIE**

1. Stroustrup, B. (2013). *The C++ Programming Language*. 4th Edition. Addison-Wesley Professional.
2. Deitel, P., Deitel, H. (2016). *C++ How to Program*. 10th Edition. Pearson Education.
3. Lippman, S., Lajoie, J., Moo, B. (2012). *C++ Primer*. 5th Edition. Addison-Wesley Professional.
4. Meyers, S. (2014). *Effective Modern C++*. O'Reilly Media.
5. cppreference.com. (2023). *C++ Reference*. Disponibil la: https://en.cppreference.com/
6. Microsoft Corporation. (2023). *Microsoft Excel Documentation*. Disponibil la: https://docs.microsoft.com/en-us/excel/

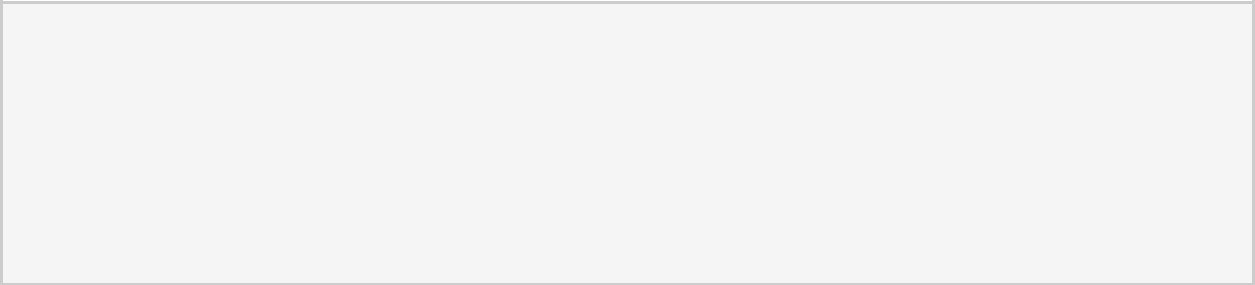
17

file:///C:/Users/205/Desktop/da.HTML 10/11

5/23/25, 12:23 PM Raport de Practică

**7. ANEXE**

**Anexa A - Exemplu de fișier Total.txt generat:**



Producatori si numarul de tipuri de jucarii fabricate:

Nefis: 3 tipuri de jucarii

Milka: 3 tipuri de jucarii

Kinder: 0 tipuri de jucarii

Ferrero: 1 tipuri de jucarii

Nestle: 0 tipuri de jucarii

Lindt: 0 tipuri de jucarii

Mars: 1 tipuri de jucarii

**Anexa B - Capturi de ecran din aplicația Excel:**

*Figura 2. Tabelul cu jucării formatat în Excel*

*Figura 3. Graficul cu distribuția jucăriilor pe categorii*

**Anexa C - Instrucțiuni de utilizare:**

Pentru rularea aplicației este necesară compilarea codului sursă C++ și existența fișierelor

Toys.txt și Manufacturer.txt în același director cu executabilul. Aplicația va crea automat fișierul

Total.txt la executarea opțiunii 9.

17

file:///C:/Users/205/Desktop/da.HTML 11/11