Proiect: Aplica■ii C++ pentru Proba Practic■

Acest proiect con■ine mai multe aplica■ii C++ separate, fiecare realizând una dintre cerin■ele specificate în proba practic■. Fiecare aplica■ie este explicat■ pas cu pas pentru a facilita în■elegerea func■ionalit■■ii ■i implement■rii.

1. Declararea ■i utilizarea variabilelor de orice tip

```
#include <iostream>
using namespace std;

int main() {
    // Declararea variabilelor
    int varInt = 0;       // Variabil■ întreag■ ini■ializat■ cu 0
    float varFloat = 0.0; // Variabil■ float ini■ializat■ cu 0.0
    char varChar = 'A';   // Variabil■ caracter ini■ializat■ cu 'A'

    // Afi■area valorilor variabilelor
    cout << "Valoarea lui varInt: " << varInt << endl;
    cout << "Valoarea lui varFloat: " << varFloat << endl;
    cout << "Valoarea lui varChar: " << varChar << endl;

    return 0;
}
```

Explica■ii:

- Se declar■ trei variabile de tip int, float ■i char.

- Fiecare variabil■ este ini■ializat■ cu o valoare implicit■ ■i afi■at■ în consol■.

2. Preluarea ■i afi■area datelor de la tastatur■

```
#include <iostream>
using namespace std;

int main() {
    int varInt;
    cout << "Introduce■i un num■r întreg: ";
    cin >> varInt; // Cite■te un num■r întreg de la tastatur■
    cout << "A■i introdus: " << varInt << endl; // Afi■eaz■ num■rul introdus

    return 0;
}
```

Explica■ii:

- Se solicit■ utilizatorului s■ introduc■ un num■r întreg.

- Num■rul introdus este citit folosind `cin` ■i apoi afi■at folosind `cout`.

3. Utilizarea unei instruc■iuni if-else

```
#include <iostream>
using namespace std;

int main() {
    int varInt;
    cout << "Introduce■i un num■r întreg: ";
```

```
    cin >> varInt; // Cite■te un num■r întreg de la tastatur■

    // Verificarea valorii lui varInt
    if (varInt > 10) {
        cout << "Num■rul este mai mare decât 10." << endl;
    } else {
        cout << "Num■rul este mai mic sau egal cu 10." << endl;
    }

    return 0;
}
```

Explica■ii:

- Se verific■ dac■ num■rul introdus de utilizator este mai mare sau mai mic decât 10 folosind o instruc■iune if-else.

4. Utilizarea unei bucle for

```
#include <iostream>
using namespace std;

int main() {
    // Bucl■ for pentru a afi■a mesajul de mai multe ori
    for (int i = 0; i < 5; i++) {
        cout << "Aceasta este itera■ia " << i+1 << endl;
    }

    return 0;
}
```

Explica■ii:

- O bucl■ for este utilizat■ pentru a afi■a un mesaj de 5 ori, incrementând un contor de la 0 la 4.

5. Definirea ■i apelarea func■iilor

```
#include <iostream>
using namespace std;

// Func■ie f■r■ parametri care afi■eaz■ un mesaj
void mesaj() {
    cout << "Aceasta este o func■ie f■r■ parametri." << endl;
}

// Func■ie care calculeaz■ suma a dou■ numere întregi ■i returneaz■ rezultatul
int suma(int a, int b) {
    return a + b;
}

int main() {
    mesaj(); // Apelarea func■iei f■r■ parametri
    cout << "Suma celor dou■ numere este: " << suma(5, 10) << endl; // Apelarea func■iei suma

    return 0;
}
```

Explica■ii:

- Se definesc dou■ func■ii: `mesaj`, care afi■eaz■ un mesaj, ■i `suma`, care calculeaz■ suma a dou■ numere.

- Ambele func■ii sunt apelate în func■ia `main`.

6. Definirea ■i utilizarea unui tablou

```
#include <iostream>
using namespace std;

int main() {
    // Definirea unui tablou ■i afi■area valorilor din acesta
    int tablou[5] = {1, 2, 3, 4, 5}; // Tablou de numere întregi
    for (int i = 0; i < 5; i++) {
        cout << "Valoarea la indexul " << i << " este: " << tablou[i] << endl;
    }

    return 0;
}
```

Explica■ii:

- Se declar■ un tablou de întregi ■i se afi■eaz■ fiecare element folosind o bucl■ for.

7. Definirea ■i utilizarea unei structuri

```
#include <iostream>
#include <cstring>
using namespace std;

// Structur■ pentru reprezentarea unui student
struct Student {
    char nume[50];  // Numele studentului
    int nota;      // Nota studentului
};

int main() {
    // Definirea ■i utilizarea unei structuri
    Student student;
    strcpy(student.nume, "Ion Popescu"); // Setarea numelui studentului
    student.nota = 10; // Setarea notei studentului
    cout << "Student: " << student.nume << ", Nota: " << student.nota << endl; // Afi■area detaliilo

    return 0;
}
```

Explica■ii:

- Se define■te o structur■ pentru a reprezenta un student ■i se afi■eaz■ detaliile acestuia.

8. Utilizarea unui pointer

```
#include <iostream>
using namespace std;

int main() {
    int varInt = 10;
    int *ptr = &varInt; // Pointer care memoreaz■ adresa lui varInt
    cout << "Adresa lui varInt: " << ptr << endl; // Afi■area adresei lui varInt
    cout << "Valoarea la care pointeaz■ ptr: " << *ptr << endl; // Afi■area valorii la care pointeaz

    return 0;
}
```

Explica■ii:

- Se declar■ un pointer care memoreaz■ adresa unei variabile ■i se afi■eaz■ adresa ■i valoarea la care pointeaz■ pointerul.

9. Modificarea valorii prin pointer

```
#include <iostream>
using namespace std;

int main() {
    int varInt = 10;
    int *ptr = &varInt; // Pointer care memoreaz■ adresa lui varInt
    *ptr = 20; // Modificarea valorii variabilei prin pointer
    cout << "Noua valoare a variabilei este: " << varInt << endl; // Afi■area noii valori a lui varI

    return 0;
}
```

Explica■ii:

- Se modific■ valoarea unei variabile folosind un pointer ■i se afi■eaz■ noua valoare.

10. Alocarea ■i eliberarea memoriei

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    // Alocarea dinamic■ a memoriei
    int *dinamic = (int *)malloc(sizeof(int) * 5); // Alocare dinamic■ pentru un tablou de 5 întregi
    if (dinamic == NULL) {
        cout << "Alocarea memoriei a e■uat." << endl;
        return 1; // Întoarce 1 în caz de eroare
    }

    // Eliberarea memoriei alocate dinamic
    free(dinamic); // Eliberarea memoriei alocate

    return 0;
}
```

Explica■ii:

- Se aloc■ ■i elibereaz■ dinamic memorie pentru un tablou de întregi folosind `malloc` ■i `free`.