

Lucrarea 2

Extensii ale limbajului C in C++

Scopul lucrării: Utilizarea extensiilor limbajului C in limbajul C++, care nu apartin paradigmei OOP.

Desfasurarea lucrării: Se vor scrie programe care utilizeaza extensii ale limbajului C, ori de cate ori este posibil:

- tipul de date `bool`;
- utilizarea referintelor la transmiterea parametrilor functiilor;
- operatorii `new` si `delete`;
- functii cu parametri cu valori implicite;
- functii supraancarcate;
- functii `template`.

Se vor utiliza obiectele `cin` si `cout`, functiile membre `get` si `put`, precum si operatorii de inserare/extractie `in`/`din` stream-uri, pentru citirea si scrierea datelor.

Se va utiliza programarea modulara, aplicatiile avand fisiere header de interfata si fisiere sursa de implementare pentru modulele dezvoltate, precum si un fisier sursa de utilizare. In cazul in care nu se specifica in mod explicit, se vor identifica si proiecta modulele corespunzatoare pentru rezolvarea fiecărei aplicatii.

Teme propuse:

1. (Tipul `bool`) Se citeste de la tastatura o secventa de siruri de caractere. Sa se dezvolte o aplicatie care determina apartenenta unui alt sir de caractere la secventa, iar in cazul in care sirul apartine secventei sa se contorizeze numarul de aparitii ale acestuia. Secventa se memoreaza intr-un tablou de siruri.
2. (Parametri de tip referinta) Se dau urmatoarele informatii despre studentii unei grupe, numarul acestora fiind cunoscut :
 - nume si prenume : sir de maxim 20 de caractere;
 - adresa, formata din:
 - nume oras : sir de maxim 15 caractere;
 - cod oras : numar intreg;
 - strada : sir de maxim 25 caractere;
 - numar : numar intreg;
 - numar de telefon : format din
 - prefix : numar intreg;
 - numar efectiv : numar intreg lung;
 - facultatea : cod format din 3 caractere;
 - varsta : numar intreg;
 - note : tablou de 10 numere intregi;
 - media de promovare;

- numar de absente la examene.

Sa se dezvolte o aplicatie care realizeaza urmatoarele operatii:

- citeste datele despre studentii dintr-o grupa;
- memoreaza datele citite intr-un tablou asociat grupei;
- afiseaza datele din tablou.

3. (Operatorii `new` si `delete`) Folosind aceleasi informatii despre studentii unei grupe din cadrul problemei 2, sa se modifice aplicatia precedenta, astfel incat memorarea studentilor sa se faca intr-un tablou cu alocare dinamica. Se vor implementa urmatoarele operatii:

- crearea tabloului;
- adaugarea unui nou student in tablou; in cazul in care tabloul nu mai are elemente disponibile, se efectueaza o realocare a acestuia de o dimensiune mai mare;
- modificarea informatiilor aferente unui student;
- stergerea unui student din tablou.

Fisierul de utilizare va contine in functia `main` un meniu care afiseaza operatiile amintite, citeste operatia selectata de utilizator si apeleaza functia specifica de tratare.

4. Sa se dezvolte aplicatia de la problema 3, adaugandu-se noi operatii asupra tabloului grupei de studenti:

- determinarea studentului (sau a studentilor) cu media cea mai mare;
- determinarea studentului (sau a studentilor) cu cele mai multe restante si absente de la examene;
- afisarea studentilor promovati:
 - in ordine alfabetica, dupa nume si prenume;
 - in ordinea descrescatoare a mediilor;
- afisarea studentilor restantieri:
 - in ordine alfabetica, dupa nume si prenume;
 - in ordinea crescatoare a numarului de restante si absente;

5. (Functiile `get` si `put`) Sa se dezvolte un modul pentru prelucrarea fisierelor sursa `.cpp` introduse de la tastatura. Se vor considera urmatoarele operatii:

- eliminarea comentariilor din fisier si afisarea fisierului modificat la terminalul standard de iesire;
- contorizarea numarului de cuvinte din fisier;
- contorizarea numarului de linii din fisier;

Fisierul de test contine apelul pe rand al celor trei operatii.

6. (Functii supraancarcate) Sa se dezvolte trei module pentru operatii cu tablouri: tablouri cu numere intregi, tablouri cu numere reale si tablouri siruri de caractere (se va utiliza clasa `string`, pentru care sunt supraancarcati operatorii `<<`, `>>`, precum si cei de comparare). Tablourile nu sunt ascunse in interiorul modulelor, ci sunt transmise ca parametri functiilor de prelucrare. Principalele operatii luate

in considerare sunt: citirea unui tablou de la tastatura, sortarea lui si afisarea lui pe ecran. Functiile corespunzatoare pentru fiecare operatie din cele trei module au acelasi nume (sunt functii supraancarcate). Sa se scrie un program de test, care sa utilizeze toate cele trei module (in cazul minimal, foloseste cate un tablou din fiecare tip, precum si utilizarea operatiilor specificate asupra fiecarui tablou).

7. (Functii `template`) In locul celor trei module din problema 6, aplicatia va contine un singur modul generic, ce utilizeaza functii `template` pentru cele trei operatii. Functia de test a aplicatiei trebuie sa fie la fel cu cea din aplicatia problemei 6.

8. (Argumente cu valori implicite) Un vector n -dimensional este un tablou de n componente reale. Adunarea a doi vectori se defineste astfel:

$$u = (x_1, \dots, x_n), v = (y_1, \dots, y_n)$$

$$u+v = (x_1+y_1, \dots, x_n+y_n)$$

Sa se scrie o functie *Add* cu parametri cu valori implicite, care sa permita adunarea a doi, trei sau patru vectori. Sa se scrie o functie de test care sa puna in evidenta modul de lucru al functie *Add* (in cazul minimal, se va utiliza aceeaasi functie pentru adunarea a 2, 3 si 4 vectori).