

POO – C++ - Laborator 2

Definirea si utilizarea tipurilor de data structură in C și C++

1. (ANSI C) Creați un tip de date structură numit *student* care să conțină: *număr matricol* de tip *int*, *nume* de tip șir de caractere *char[]*, *gen* de tip *char*, *nota* de tip *float* mobilă simplă precizie *float*, doi pointeri pe funcții pentru citire date *void (read*)(student*st)* și unul pentru afisare date *void (write*)(student*st)*.

Citiți de la tastatura (folosind *cin*) numărul de studenți și alocați dinamic memorie pentru un vector de studenți (folosiți *v = new student[n]*).

Definiți două funcții: *void ReadData(student*st){...}* și *void WriteData(student*st){...}* care citesc și afisează membrii unei variabile de tip structură *student*. Parcurgeți vectorul cu elemente de tip *student* și inițializați pointerii la funcții *void (read*)(student*st)* și *void (write*)(student*st)* cu funcțiile *void ReadData(student*st)* și *void WriteData(student*st)*.

Parcurgeți vectorul cu elemente de tip *student* și apelați funcțiile: *v[i].read(&v[i])* și *v[i].write(&v[i])* pentru citirea datelor și respectiv pentru afisarea datelor.

2. (C++) Declarați două funcții membre ale structurii *student* *void Read(void)* și *void Write(void)* și definiți aceste funcții în afara structurii: *void student::Read(void){...}* și *void student::Write(void){...}*.

Parcurgeți vectorul cu elemente de tip *student* și apelați funcțiile: *v[i].Read()* și *v[i].Write()* pentru citirea datelor și respectiv pentru afisarea datelor.

Analizați și discutați cele două abordări.

3. Rezolvați problema 8 din laboratorul 1 declarând ca membre ale structurii funcțiile pentru sortare alfabetică, sortare după lungime, citire și afisare de date.

obs. nu uitați dealocare zonelor de memorie (folosiți *delete []v*)