**Laborator recapitulativ – Agregare, derivare, functii virtuale, interfete**

Se da urmatoarea structura de clase - clasa de baza este Piesa cu atributele cod si pret din care sunt derivate clasele Piesa\_A - care are in plus atributul gauri si Piesa\_B care are in plus atributul culoare.

class Piesa\_B: public Piesa{

string culoare;

};

class Piesa\_A: public Piesa{

int gauri;

};

Piesele de tip A si B sunt folosite pentru a realiza un Ansamblu (agregare) in felul urmator: se va folosi o piesa de tip A (prelucrata prin gaurire) – care are un anumit nr. de gauri si o serie de piese de tip B (prelucrate prin vopsire) care vor fi montate in gaurile piesei de tip A.

class Piesa{

protected: char cod[4];

int pret;

};

class Ansamblu{

Piesa\_A pa;

Piesa\_B \*vec; // are ca nr. de elemente nr. de gauri ale lui pa

int nr\_caracteristici;

string \*caracteristici; //lungimea vectorului e nr\_caracteristici

}; // pretul este dat de suma preturilor pieselor componente

**Cerinte:**

1. Realizati un vector care sa contina obiecte de tip: Piesa\_A, Piesa\_B si Ansamblu cu valori alese de voi.

2. Sortati-l dupa pret.

3. Parcurgeti-l si afisati **toate** atributele obiectelor intr-un fisier (pe fiecare linie – atributele unui obiect).

**Observatii:**

Implementati toate metodele care, generate automat, nu ar functiona corect.

Tineti cont de regulile de reutilizare a codului in contextul derivarii si agregarii.