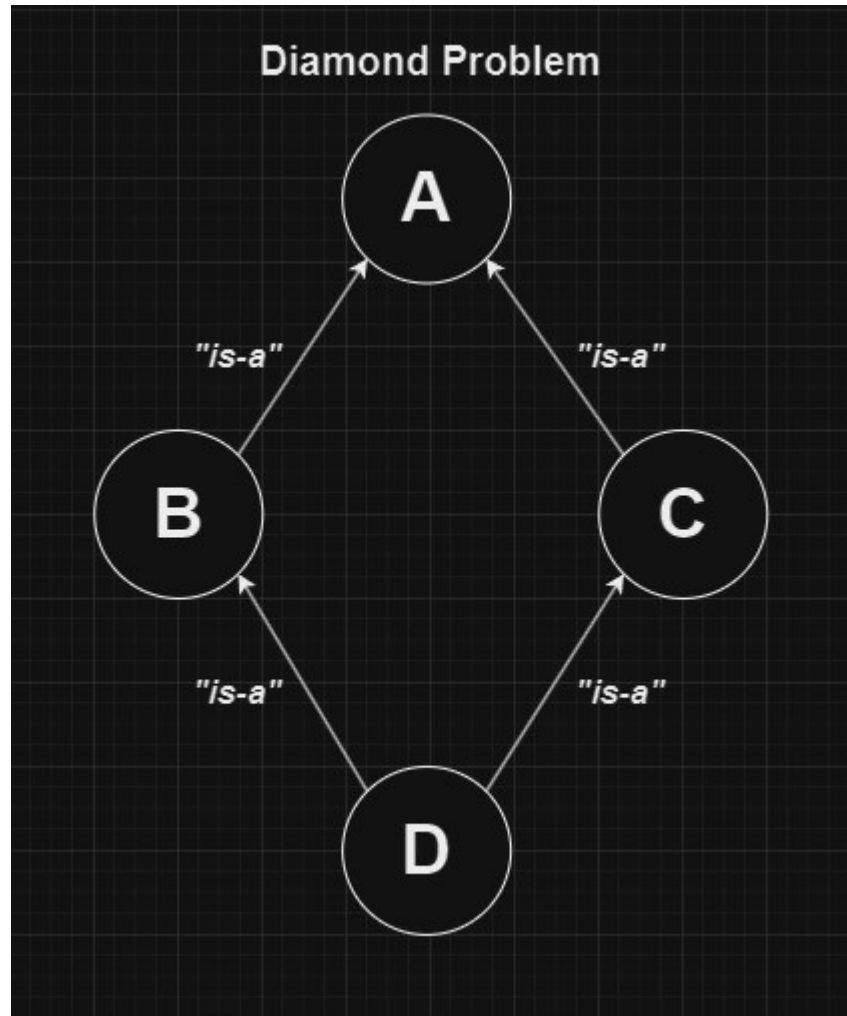


## Laborator 12

### ~Problema diamantului~



---

Exercițiu 1 – Citiți rapid laboratorul 12 postat pe OCW (10-15 min)

---

## Exercițiu 2

---

```
//clasa de baza "A"
class Hartie {
    char* m_numeFabrica;
    int m_pret;
    // Destructorul trebuie sa fie virtual (nu virtual pur) !!
};

class Carte: virtual public Hartie {
    float m_greutate;
    int m_pret; // diferit fata de m_pret din Hartie, e pretul propriu clasei !!!!!!!!

    //clasa are 4 atribute dintre care 2 sunt preturi D I F E R I T E
public:
    //Constructorii (fara parametrii, cu parametrii si de copiere) + Destructor
    //Operator=
};

class Coperta: virtual public Hartie {
    float m_greutate;
    int m_pret; // diferit fata de m_pret din Hartie, e pretul propriu clasei !!!!!!!!

    // si aceasta clasa are 4 atribute dintre care 2 sunt preturi D I F E R I T E
public:
    //ca mai sus
};

class CarteSiCoperta: public Carte, public Coperta {
    char* m_titluCarte;
    // pretului unui obiect de acest tip va fi:
    // pret total = pret carte + pret coperta + pret hartie
    // adica m_pret D I N F I E C A R E C L A S A P A R I N T E A A C E S T E I C L A S E
public:
    //ca mai sus, aveti grija ca trebuie apelat direct constructorul clasei de baza "A",
    Hartie, cititi cu atentie materialele de pe OCW
};
```

**Cerințe:**

1. Implementați toate metodele astfel încât programul să funcționeze corect
2. Creați în main un array de minim **5** elemente neomogene (cel puțin 1 din fiecare clasă, vedeți că în acest laborator fiecare clasă poate fi instanțiată, nu mai avem interfețe) ← FĂRĂ STL
3. Afișați elementele array-ului (prin metode specifice de afișare sau implementând operatorul <<)
4. Sortați array-ul după prețul specific fiecărei clase (țineți cont de cum este calculat prețul în clasa finală CarteSiCoperta) și îl mai afișați o dată
5. Eliberați memoria corespunzător

**Obs1:** Amintiți-vă de metodele virtuale despre care am vorbit în laboratoarele trecute pentru a rezolva problema afișării unui vector neomogen. Chiar dacă nu mai avem o clasă de tip interfață, principiile metodelor virtuale încă se aplică, dar nu mai trebuie să fie virtual *pure*

**Obs2:** Chiar dacă nu mai avem o interfață, orice obiect poate fi instanțiat într-un pointer la clasa lui de bază precum copii moștenitori ai acestuia:

A\* obj1 = new A();  
A\* obj2 = new B(); etc

**Punctaj:**

Exercițiul 1 – 0 puncte

Exercițiul 2 - Cerința 1 - 3 puncte

Exercițiul 2 - Cerința 2 - 3 puncte

Exercițiul 2 - Cerința 3 - 1 puncte

Exercițiul 2 - Cerința 4 - 1 puncte

Exercițiul 2 - Cerința 5 - 1 puncte

1 punct din oficiu