

## LUCRARE DE LABORATOR NR. 3

**Tema:** Moștenirea și compoziția

**Scopul lucrării:**

- studierea moștenirii, avantajele și dezavantajele
- studierea compoziției
- studierea regulilor de definire a moștenirii și compoziției
- studierea formelor de moștenire
- studierea inițializatorilor
- principiul de substituție
- moștenire și compoziție

### Noțiuni de bază

Derivarea permite definirea într-un mod simplu, eficient și flexibil a unor clase noi prin adăugarea unor funcționalități claselor deja existente, fără să fie necesară reprogramarea sau recompilarea acestora. Clasele derivate exprimă relații ierarhice între conceptele pe care acestea le reprezintă și asigură o interfață comună pentru mai multe clase diferite.

Moștenirea poate fi abordată din două puncte de vedere: al elaboratorului și al utilizatorului clasei.

Din punctul de vedere al elaboratorului, moștenirea abordează comportarea și proprietățile clasei derivate ca extensie a proprietăților clasei de bază. Din punct de vedere al utilizatorului – moștenirea semnifică existența unui șir de clase parțial intershibabile cu o interfață unică. (Inginerii TI și inginerii mecanici sunt ambii, în primul rând, ingineri).

#### Avantajele moștenirii:

- micșorarea volumului de cod și utilizarea lui repetată,
- micșorarea numărului de erori și a prețului de cost,
- interfață unică și substituție,
- mărirea vitezei de elaborare.

Un neajuns al moștenirii ar fi o oarecare micșorare a vitezei de executare a codului. Totuși eficiența nu trebuie să contrazică avantajele, deoarece cheltuielile reale nu sunt esențiale, ele fiind evitate parțial prin utilizarea funcțiilor.

#### Definirea și utilizarea moștenirii

O clasă care asigură proprietăți comune mai multor clase se definește ca o clasă de bază. O clasă derivată moștenește de la una sau mai multe clase de bază toate caracteristicile acestora, cărora le adaugă alte caracteristici noi, specifice ei.

În general, derivarea unei clase se specifică în felul următor:

```
class nume_derivata : specificator_acces nume_baza
{
    // corpul clasei
};
```

Specificatorul de acces poate fi unul din cuvintele-cheie: *public*, *private*, *protected*.

De exemplu:

```
class Animal{
    int NofLegs;
public:
    void Say(){ cout<<"!!!"; }
};
class Dog: public Animal{ // moștenire
...
};
void main(){
    Dog d;
    d.Say();
}
```

Din exemplu se vede că, funcția și variabila definite în clasa *Animal* sunt importate și pot fi utilizate de obiectul clasei *Dog*. Clasa derivată poate să supradefinească comportarea definită în clasa de bază. (Struțul și pinguinul nu zboară, deși sunt păsări, dar ornitoringul depune ouă, deși este mamifer).

### Principiul de substituție

Moștenirea mai presupune că, obiectul clasei derivate poate fi utilizat în locul obiectului clasei de bază:

```
void main() {  
    Animal *ptr = new Dog;}
```

În exemplul dat se declară un pointer spre clasa de bază – deci se așteaptă utilizarea obiectului clasei de bază, însă, pointerului i se atribuie obiectul clasei derivate.

### Formele moștenirii

În C++ există cinci forme diferite de moștenire. În lucrarea de față vor fi analizate trei tipuri, numite moșteniri simple care au protecția:

- public,
- private,
- protected,

Formele diferite de moștenire se utilizează pentru modificarea statutului de acces pentru elementele membru ale clasei.

Când o clasă este derivată dintr-o clasă de bază, clasa derivată moștenește toți membrii clasei de bază (cu excepția unora: constructori, destructor și funcția operator de asignare). Tipul de acces din clasa derivată la membrii clasei de bază este dat de specificatorul de acces. Dacă nu este indicat, specificatorul de acces este implicit *private*.

Când specificatorul de acces este *public*, toți membrii de tip *public* ai clasei de bază devin membri de tip *public* ai clasei derivate; toți membrii *protected* ai clasei de bază devin membri *protected* ai clasei derivate. Membrii *private* ai clasei de bază rămân *private* în clasa de bază și nu sunt accesibili membrilor clasei derivate. Această restricție de acces ar putea pare surprinzătoare, dar, dacă s-ar permite accesul dintr-o clasă derivată la membrii *private* ai clasei de bază, noțiunile de încapsulare și ascundere a datelor nu ar mai avea nici o semnificație.

La utilizarea moștenirii de tip *private* a clasei de bază, toate elementele accesibile ( *public* și *protected*) din clasa de bază devin *private* în clasa derivată. Iar la o moștenire de tip *protected* – devin *protected*. Elementele cu acces *private* ale clasei de bază la orice formă de moștenire sunt inaccesibile în clasa derivată. În afară de aceasta, formele de moștenire se deosebesc prin faptul că moștenirea de tip *public* crează un subtip de date, adică corespunde principiului de substituție, iar moștenirea de tip *protected* și *private* – nu.

### Compoziția

Compoziția reprezintă încă un mecanism legat de POO, care desemnează relația dintre obiecte, pe când moștenirea este relația dintre clase.

Moștenirea realizează relația este "*is a*". *Cîinele* este *mamifer*, iar *mamiferul* – *animal*. Compoziția realizează relația *conține, are "has a"*. *Automobilul* conține *motor* și *roți*.

### Definirea compoziției

De fapt compoziția se aplică foarte larg, deoarece și variabilele predefinite au tip și se utilizează la definirea clasei. În cazul claselor utilizator, complexitatea constă în utilizarea constructorilor, definirea propriu-zisă fiind simplă:

```
class Car{  
    Engine e;  
};
```

### Listele de inițializare

Se știe că variabilele clasei de bază protejate cu modificatorul *private* nu sunt accesibile, prin urmare, ele nu pot fi inițializate în constructorul clasei derivate, mai mult ca atât, aceasta contrazice principiul de reutilizare a codului. Soluția este una – apelul constructorului clasei de bază. Acest lucru are loc dar numai pentru constructorii implicați și de copiere, generați de compilator. Restul constructorilor trebuie să se apeleze manual. Aceeași problemă apare și în cazul compoziției, de asemenea poate fi utilizat numai constructorul implicit și de copiere. Pentru soluționarea acestor probleme sunt utilizate listele de inițializare – care permit apelarea oricărui tip de constructor și efectuează orice inițializare.

```
class Engine{
    int power;
public:
    Engine(int p){power=p;}
};
class Transport{
    ...
public:
    Transport(char*);
};
class Car:public Transport{ // наследование
    Engine e;                // композиция
public:
    Car():Transport("automobile"),e(10){ }
};
```

Pentru apelul constructorului clasei de bază, după parantezele constructorului clasei derivate prin două puncte se scrie constructorul clasei de bază, fiind posibil transferul parametrilor din constructorul clasei derivate în constructorul clasei de bază. În cazul compoziției situația este asemănătoare dar se scrie numele variabilei și nu a constructorului.

### Ce alegem

Atât derivarea cât și compoziția reprezintă instrumente de reutilizarea a codului, apare o întrebare evidentă: când se utilizează moștenirea și când – compoziția. În acest sens, există multe recomandări diferite, însă cel mai simplu este să ne conducem de regula: se pune întrebarea: clasa nouă este un subtip (*Dog is an Animal*), dacă răspunsul este afirmativ, se utilizează moștenirea, în alt caz se pune întrebarea: clasa nouă reprezintă oare un container (*Car has a door*) – în acest caz se utilizează compoziția.

Există însă situații când nici această strategie nu este suficientă. De exemplu, clasa *mulțime* poate fi creată în baza clasei *listă*, dar nu se știe ce mecanism trebuie de folosit. Regulile în acest caz sunt mai complicate, se pune un set de întrebări:

- vor putea fi înlocuite obiectele clasei existente prin obiectele clasei noi?
- este necesar să se supradefinească careva funcție virtuală?
- prelucrează oare tipul nou aceleași mesaje ca și cel vechi?
- este oare clasa de bază abstractă?

Dacă răspunsurile sunt afirmative, se utilizează moștenirea.

### **Întrebări de control:**

1. Prin ce se deosebesc formele de moștenire?
2. Ce reprezintă compoziția ?
3. Care sunt avantajele moștenirii?
4. Cum lucrează constructorii la moștenire?
5. Ce reprezintă principiul de substituție?
6. În ce cazuri se utilizează moștenirea și în care se utilizează compoziția?
7. Cum pot fi inițializate elementele containerului?

## Sarcina

### Varianta 1

- a) Să se creeze o ierarhie a claselor *joc – joc sportiv – volei*. Determinați constructorii, destructorul, operatorul de atribuire și alte funcții necesare.
- b) Să se creeze clasa *roata*, care conține rază. Determinați constructorii și metodele de acces. Creați clasa *automobil*, care conține roți și un câmp care reprezintă firma producătoare. Creați o clasă derivată *autocamion* care se deosebește prin tonaj. Determinați constructorii, destructorul și alte funcții necesare.

### Varianta 2

- a) Să se creeze clasa *student*, care are un nume, specialitate, anul de învățământ și balul mediu. Determinați funcția de definire, schimbare a datelor și de comparare. Pentru setarea câmpurilor textuale să se folosească operatorul *new*. Determinați constructorii, destructorul și alte funcții necesare. Creați clasa derivată *student-diplomant*, pentru care este definită tema de diplomă. De asemenea, este necesar de definit toate funcțiile necesare.
- b) Să se creeze clasa *camera*, care conține suprafață. Determinați constructorii și metodele de acces. Creați clasa *apartament cu o odaie*, care conține o odaie și o bucatărie (suprafața ei), etajul (camera este în clasa apartament cu o odaie). Determinați constructorii, metodele de acces. Definiți clasa derivată a apartamentelor cu o odaie cu adresă (un câmp adăugător - adresa). Determinați constructorii, destructorul și fluxul de ieșire.

### Varianta 3

- a) Să se creeze clasa *mobilă*, care conține informație despre preț, stil și domeniul de utilizare (oficiu, bucatărie și altă mobilă). Pentru setarea câmpurilor textuale să se folosească memoria dinamică. Definiți clasele derivate *masa* și *scaun*. Definiți constructorii, destructorul, operatorii de atribuire și alte funcții necesare.
- b) Să se creeze clasa *garaj*, care conține suprafața. Determinați constructorii și metodele de acces. Creați clasa *casa*, care conține odăi, o bucatărie (suprafața ei) și garaj. Definiți clasa derivată *vilă* (ca parametru adăugător – mărimea lotului de pământ). Determinați constructorii, destructorul și fluxul de ieșire.

### Varianta 4

- a) Să se creeze o ierarhie a claselor *om* și *colaborator*, care ocupă un post anumit și primește un salariu anumit. Să se supraîncarce operatorii pentru ieșiri și intrări de obiecte, constructorul de copiere, operatorii de atribuire utilizând funcțiile respective ale clasei de bază.
- b) Să se creeze clasa *carte de joc*, care conține grad și culoare. Cartea poate fi întoarsă și deschisă. Creați clasa – *butuc de cărți*, care conține cărți de joc. Creați două clase derivate de la butuc de cărți, în una cărțile pot fi scoase numai într-o ordine, iar în alta – aleator.

### Varianta 5

- a) Să se creeze clasa *lichid*, care conține denumirea (pointer spre char), densitatea. Definiți constructorii, destructorul, operatorii fluxului de intrare. Creați o clasă derivată – băuturi alcoolice, care conține tărie. Determinați funcțiile de redenumire a densității și a tăriei.
- b) Să se creeze clasa *buton*, care conține un text anumit. Definiți constructorii și metodele de acces. Creați clasa *fereastră*, care conține un buton și coordonatele ferestrei. Definiți

constructorii și destructorul. Definiți clasa derivată *fereastra cu buton și cu un mesaj*. Definiți constructorii, destructorul și operatorii fluxului de ieșire.

### Varianta 6

a) Să se creeze clasa *om*, care are numele (pointer spre char), vârsta, greutatea. Determinați constructorii, destructorul și operatorul de atribuire. Creați clasa derivată - *matur*, care are numărul pașaportului. Definiți constructorii: impliciți, cu parametri, destructorul, operatorii fluxului de ieșire. Definiți funcțiile de resetare a vârstei și a numărului pașaportului.

b) Definiți clasa *vaca* care este compusă din următoarele câmpuri: numărul de identificare – trebuie să fie garantat unic (pentru care se utilizează un contor static), cantitatea medie de lapte, vârsta, numele și rasa. Pentru setarea câmpurilor textuale utilizați operatorul new. Definiți clasa *cireadă*, care constă dintr-un număr nelimitat de vaci. Definiți metodele de introducere, eliminare, de determinare a cantității medii de lapte în cireadă și cantitatea totală de lapte în cireadă și alte funcții necesare.

### Varianta 7

a) Să se creeze o ierarhie a claselor *clădire –clădire administrativă – clădire de locuit*. Să se supraîncarce fluxul de ieșire și fluxul de intrare, să se definească constructorul de copiere, operatorul de atribuire prin funcțiile corespunzătoare ale clasei de bază.

b) Să se creeze clasa *student*, care are nume, specialitate, anul de studiu și balul mediu. Determinați funcția de definire, schimbare a datelor și comparare. Pentru setarea câmpurilor textuale se folosește operatorul new. Definiți constructorii, destructorul și alte funcții necesare. Creați clasa *grupa*, care conține studenți (un număr nelimitat). Definiți metodele de adăugare și eliminare a studenților, de determinare a balului mediu pe grupă, constructorii, destructorul și alte funcții necesare.

### Varianta 8

a) Să se creeze ierarhia de clase *bloc de studiu* - clasa de bază abstractă și *instituție preșcolară, instituție școlară medie și instituție de învățământ superior* – clase derivate. Să se supraîncarce fluxul de ieșire și fluxul de intrare, să se definească constructorul de copiere, operatorul de atribuire prin funcțiile corespunzătoare ale clasei de bază.

b) Să se creeze clasa *motor*, care are o firmă producătoare, tip și putere. Determinați funcțiile de setare, de schimbare a parametrilor motoarelor. Creați o ierarhie a claselor: *corabie* – clasa de bază și *vapor pentru pasageri* – derivată. Corabia are un motor, capacitate de încărcare, măsurător de apă, denumire, portul unde este înscris. Pentru setarea câmpurilor textuale, utilizați operatorul new.

### Varianta 9

a) Să se creeze o ierarhie a claselor *presă – ziar, revistă și ziar electronic*. Determinați câmpurile: denumirea ziarului, tirajul, indexul de abonare, periodicitatea de publicare. Pentru setarea câmpurilor textuale, utilizați operatorul new. Definiți fluxul de ieșire și fluxul de intrare, constructorul de copiere, operatorul de atribuire prin funcția corespunzătoare a clasei de bază.

b) Să se creeze clasa *Procesor*, care conține informația despre denumirea procesorului, frecvența lui, tehnologiile de producere utilizate și mărimea memoriei. Determinați clasa *computer*, care

este compusă dintr-un procesor și alte componente. Pentru setarea câmpurilor textuale, utilizați operatorul *new*. Definiți constructorii, funcția fluxului de ieșire și alte funcții necesare.

### Varianta 10

- a) Să se creeze o ierarhie a claselor *transport* – *transport aerian* – *elicopter*. Definiți fluxul de ieșire și fluxul de intrare, constructorul de copiere, operatorul de atribuire prin funcția corespunzătoare a clasei de bază.
- b) Definiți clasa *element chimic*, care conține informația despre denumirea elementului și proprietățile lui chimice. Determinați clasa *medicamente*, care conține un număr diferit de elemente chimice și în cantități diferite. Determinați constructorii, funcțiile fluxului de ieșire și alte funcții necesare.

### Varianta 11

- a) Să se creeze o ierarhie a claselor *senzor* - clasa de bază abstractă și senzori de temperatură, de umiditate și de măsurare a vitezei vântului. Pentru fiecare clasă să se definească unitățile sale de măsură și metoda de citire a datelor despre starea mediului înconjurător. Să se supraîncarce fluxul de ieșire și fluxul de intrare, constructorul de copiere, operatorul de atribuire prin funcțiile corespunzătoare ale clasei de bază.
- b) Să se creeze clasa *dispozitiv de colectare a informației* despre starea timpului care este compusă din senzori (conform sarcinei a). Pentru citirea valorilor să se creeze clasa *generatorul de valori* pentru fiecare senzor. Să se arate funcționarea dispozitivului.

### Varianta 12

- a) Să se creeze clasa *student*, care are un nume, specialitate, anul de învățământ și balul mediu. Determinați funcția de definire, schimbare a datelor și de comparare. Pentru setarea câmpurilor textuale să se folosească operatorul *new*. Determinați constructorii, destructorul și alte funcții necesare. Creați clasa derivată *student-diplomant*, pentru care este definită tema de diplomă. De asemenea, este necesar de definit toate funcțiile necesare.
- b) Să se creeze clasa *camera*, care conține suprafață. Determinați constructorii și metodele de acces. Creați clasa *apartament cu o odaie*, care conține o odaie și o bucatărie (suprafața ei), etajul (camera este în clasa apartament cu o odaie). Determinați constructorii, metodele de acces. Definiți clasa derivată a apartamentelor cu o odaie cu adresă (un câmp adăugător - adresă). Determinați constructorii, destructorul și fluxul de ieșire.

### Varianta 13

- a) Să se creeze clasa *mobilă*, care conține informație despre preț, stil și domeniul de utilizare (oficiu, bucatărie și altă mobilă). Pentru setarea câmpurilor textuale să se folosească memoria dinamică. Definiți clasele derivate *masa* și *scaun*. Definiți constructorii, destructorul, operatorii de atribuire și alte funcții necesare.
- b) Să se creeze clasa *garaj*, care conține suprafața. Determinați constructorii și metodele de acces. Creați clasa *casa*, care conține odăi, o bucatărie (suprafața ei) și garaj. Definiți clasa derivată *vilă* (ca parametru adăugător – mărimea lotului de pământ). Determinați constructorii, destructorul și fluxul de ieșire.

#### Varianta 14

- a) Să se creeze o ierarhie a claselor *om* și *colaborator*, care ocupă un post anumit și primește un salariu anumit. Să se supraîncarce operatorii pentru ieșiri și intrări de obiecte, constructorul de copiere, operatorii de atribuire utilizând funcțiile respective ale clasei de bază.
- b) Să se creeze clasa *carte de joc*, care conține grad și culoare. Cartea poate fi întoarsă și deschisă. Creați clasa – *butuc de cărți*, care conține cărți de joc. Creați două clase derivate de la butuc de cărți, în una cărțile pot fi scoase numai într-o ordine, iar în alta – aleator.

#### Varianta 15

- a) Să se creeze clasa *lichid*, care conține denumirea (pointer spre char), densitatea. Definiți constructorii, destructorul, operatorii fluxului de intrare. Creați o clasă derivată – băuturi alcoolice, care conține tărie. Determinați funcțiile de redenumire a densității și a tăriei.
- b) Să se creeze clasa *buton*, care conține un text anumit. Definiți constructorii și metodele de acces. Creați clasa *fereastră*, care conține un buton și coordonatele ferestrei. Definiți constructorii și destructorul. Definiți clasa derivată *fereastră cu buton și cu un mesaj*. Definiți constructorii, destructorul și operatorii fluxului de ieșire.

#### Varianta 16

- a) Să se creeze clasa *om*, care are numele (pointer spre char), vârsta, greutatea. Determinați constructorii, destructorul și operatorul de atribuire. Creați clasa derivată - *matur*, care are numărul pașaportului. Definiți constructorii: implicați, cu parametri, destructorul, operatorii fluxului de ieșire. Definiți funcțiile de resetare a vârstei și a numărului pașaportului.
- b) Definiți clasa *vaca* care este compusă din următoarele câmpuri: numărul de identificare – trebuie să fie garantat unic (pentru care se utilizează un contor static), cantitatea medie de lapte, vârsta, numele și rasa. Pentru setarea câmpurilor textuale utilizați operatorul new. Definiți clasa *cireadă*, care constă dintr-un număr nelimitat de vaci. Definiți metodele de introducere, eliminare, de determinare a cantității medii de lapte în cireadă și cantitatea totală de lapte în cireadă și alte funcții necesare.

#### Varianta 17

- a) Să se creeze o ierarhie a claselor *clădire* – *clădire administrativă* – *clădire de locuit*. Să se supraîncarce fluxul de ieșire și fluxul de intrare, să se definească constructorul de copiere, operatorul de atribuire prin funcțiile corespunzătoare ale clasei de bază.
- b) Să se creeze clasa *student*, care are nume, specialitate, anul de studiu și balul mediu. Determinați funcția de definire, schimbare a datelor și comparare. Pentru setarea câmpurilor textuale se folosește operatorul new. Definiți constructorii, destructorul și alte funcții necesare. Creați clasa *grupa*, care conține studenți (un număr nelimitat). Definiți metodele de adăugare și eliminare a studenților, de determinare a balului mediu pe grupă, constructorii, destructorul și alte funcții necesare.

#### Varianta 18

- a) Să se creeze o ierarhie a claselor *joc* – *joc sportiv* – *volei*. Determinați constructorii, destructorul, operatorul de atribuire și alte funcții necesare.

b) Să se creeze clasa *roata*, care conține rază. Determinați constructorii și metodele de acces. Creați clasa *automobil*, care conține roți și un câmp care reprezintă firma producătoare. Creați o clasă derivată *autocamion* care se deosebește prin tonaj. Determinați constructorii, destructorul și alte funcții necesare.

### Varianta 19

a) Să se creeze clasa *student*, care are un nume, specialitate, anul de învățământ și balul mediu. Determinați funcția de definire, schimbare a datelor și de comparare. Pentru setarea câmpurilor textuale să se folosească operatorul *new*. Determinați constructorii, destructorul și alte funcții necesare. Creați clasa derivată *student-diplomant*, pentru care este definită tema de diplomă. De asemenea, este necesar de definit toate funcțiile necesare.

b) Să se creeze clasa *camera*, care conține suprafață. Determinați constructorii și metodele de acces. Creați clasa *apartament cu o odaie*, care conține o odaie și o bucatărie (suprafața ei), etajul (camera este în clasa apartament cu o odaie). Determinați constructorii, metodele de acces. Definiți clasa derivată a apartamentelor cu o odaie cu adresă (un câmp adăugător - adresa). Determinați constructorii, destructorul și fluxul de ieșire.

### Varianta 20

a) Să se creeze clasa *mobilă*, care conține informație despre preț, stil și domeniul de utilizare (oficiu, bucatărie și altă mobilă). Pentru setarea câmpurilor textuale să se folosească memoria dinamică. Definiți clasele derivate *masa* și *scaun*. Definiți constructorii, destructorul, operatorii de atribuire și alte funcții necesare.

b) Să se creeze clasa *garaj*, care conține suprafața. Determinați constructorii și metodele de acces. Creați clasa *casa*, care conține odăi, o bucatărie (suprafața ei) și garaj. Definiți clasa derivată *vilă* (ca parametru adăugător – mărimea lotului de pământ). Determinați constructorii, destructorul și fluxul de ieșire.

### Varianta 21

a) Să se creeze o ierarhie a claselor *om* și *colaborator*, care ocupă un post anumit și primește un salariu anumit. Să se supraîncarce operatorii pentru ieșiri și intrări de obiecte, constructorul de copiere, operatorii de atribuire utilizând funcțiile respective ale clasei de bază.

b) Să se creeze clasa *carte de joc*, care conține grad și culoare. Cartea poate fi întoarsă și deschisă. Creați clasa – *butuc de cărți*, care conține cărți de joc. Creați două clase derivate de la butuc de cărți, în una cărțile pot fi scoase numai într-o ordine, iar în alta – aleator.

### Varianta 22

a) Să se creeze clasa *lichid*, care conține denumirea (pointer spre char), densitatea. Definiți constructorii, destructorul, operatorii fluxului de intrare. Creați o clasă derivată – băuturi alcoolice, care conține tărie. Determinați funcțiile de redenumire a densității și a tăriei.

b) Să se creeze clasa *buton*, care conține un text anumit. Definiți constructorii și metodele de acces. Creați clasa *fereastră*, care conține un buton și coordonatele ferestrei. Definiți constructorii și destructorul. Definiți clasa derivată *fereastră cu buton și cu un mesaj*. Definiți constructorii, destructorul și operatorii fluxului de ieșire.