

# Лабораторна робота №3

**Тема:** Успадковування класів

**Мета:** ознайомитись зі способами та механізмами успадкування класів та навчитись використовувати їх для побудови об'єктно-орієнтованих програм.

## Завдання 1:

Уявіть собі видавничу компанію, яка торгує книгами і аудіо-записами цих книг. Створіть клас `publication`, в якому зберігаються назва (рядок) і ціна (тип `float`) книги. Від цього класу успадковуються ще два класи: `book`, який містить інформацію про кількість сторінок у книзі (типу `int`), і `ture`, який містить час запису книги у хвилинах (тип `float`). У кожному з цих трьох класів повинен бути метод `getdata()`, через який можна отримувати дані від користувача з клавіатури, і `putdata()`, призначений для виведення цих даних. Напишіть функцію `main()` програми для перевірки класів `book` і `ture`. Створіть їх об'єкти в програмі і запросіть користувача ввести і вивести дані з використанням методів `getdataQ` і `putdata()`.

## Код програми

```
#include <iostream>
#include <string.h>

using namespace std;

class publication
{
private:
    string name;
    float cost;
public:

    publication()
    {

    }

    void getdata()
    {
        cout<<"Book name\n";
        cin>>name;cout<<"Cost=\n";
        cin>>cost;

    }

    void putdata()
    {
        cout<<"NAME="<<name<<endl;
```

```

        cout<<"Cost="<<cost<<endl;

    }

};

class book : public publication
{
private:
int n;
public:
    book() : publication()
    {

    }

    void getdata()
    {
publication::getdata();
cout<<"Number page"<<endl;
        cin>>n;
    }

    void putdata()
    {
publication::putdata();
cout<<"Number page"<<n<<endl;
    }
};

class type : publication
{
private:
    float sec;
    float min;
    float hour;
public:
    type () : publication()
    {}
    void getdata()
    {
        publication::getdata();
        cout<<"Enter hour= ";
        cin>>hour;
        cout<<"Enter minutes= ";
        cin>>min;
        cout<<"Enter seconds= ";
        cin>>sec;
    }

    void putdata()
    {
publication::putdata();
publication::putdata();
        if(sec>60) {do{sec=sec-60; min++;}while(sec>60);}
        if(min>60) {do{min=min-60; hour++;}while(min>60);}
        if(hour>24) do{hour=hour-24;}while(hour>24);
        cout<<"time="<<hour<<":"<<min<<":"<<sec<<endl;
    }

};

int main()
{
cout<<"\tBuild Base Class\n";
    publication k;    //Створення базового класу

```

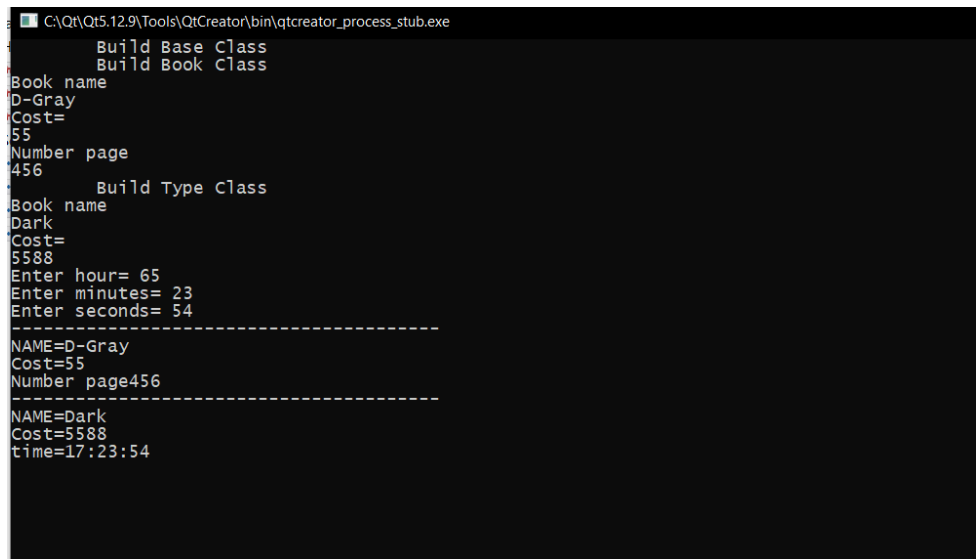
```

        cout<<"\tBuild Book Class\n";
        book l;          //Створення класу book
        l.getdata();
        l.putdata();
        cout<<"\tBuild Type Class\n";
        type j;          //Створення класу type
        j.getdata();
        j.putdata();

        return 0;
}

```

## Результат:



```

C:\Qt\Qt5.12.9\Tools\QtCreator\bin\qtcreator_process_stub.exe
Build Base Class
Build Book Class
Book name
D-Gray
Cost=
55
Number page
456
Build Type Class
Book name
Dark
Cost=
5588
Enter hour= 65
Enter minutes= 23
Enter seconds= 54
-----
NAME=D-Gray
Cost=55
Number page456
-----
NAME=Dark
Cost=5588
time=17:23:54

```

## Завдання 2:

До класів з попереднього завдання (попередньо зберігши окремо код) додайте базовий клас `sales`, в якому міститься масив, що складається з трьох значень типу `float`, куди можна записати загальну вартість проданих книг за останні три місяці. Включіть в клас методи `getdata()` для отримання значень вартості від користувача і `putdata()` для виведення цих цифр. Змініть класи `book` і `type` так, щоб вони стали похідними обох класів: `publication` і `sales`. Об'єкти класів `book` і `type` повинні вводити і виводити дані про продажі разом з іншими своїми даними. Напишіть функцію `main()` для створення об'єктів класів `book` і `type`, щоб протестувати можливості введення/виведення даних

## Код програми

```
#include <iostream>
#include <string.h>

using namespace std;

class publication
{
private:
    string name;
    float cost;
public:

    publication()
    {

    }
    void getdata()
    {
        cout<<"Book name\n";
        cin>>name;cout<<"Cost=\n";
        cin>>cost;

    }

    void putdata()
    {
        cout<<"NAME="<<name<<endl;
        cout<<"Cost="<<cost<<endl;

    }
};

class sales
{
private:
    int n[3];
public:
    sales () {}

    void getdata()
    {
        for(int i=0;i<3;i++)
        {

            cout<<"Sales mount"<<i+1<<"=";
            cin>>n[i];

        }
    }
    void putdata()
    {
        for(int i=0;i<3;i++)
        {
            cout<<"Salse mount"<<i+1<<"="<<n[i]<<endl;

        }
    }

};
```

```

class book : public publication, public sales
{
private:
    int n;
public:
    book() : publication(), sales()
    {

    }
    void getdata()
    {
        publication::getdata();
        cout<<"Number page"<<endl;
        cin>>n;
        sales::getdata();
    }

    void putdata()
    {
        publication::putdata();
        cout<<"Number page"<<n<<endl;
        sales::putdata();
    }
};

class type : publication, sales
{
private:
    float sec;
    float min;
    float hour;
public:
    type() : publication(), sales()
    {}
    void getdata()
    {
        publication::getdata();
        sales::getdata();
        cout<<"Enter hour= ";
        cin>>hour;
        cout<<"Enter minutes= ";
        cin>>min;
        cout<<"Enter seconds= ";
        cin>>sec;
    }

    void putdata()
    {
        publication::putdata();

        if(sec>60) {do{sec=sec-60; min++;}while(sec>60);}
        if(min>60) {do{min=min-60; hour++;}while(min>60);}
        if(hour>24) do{hour=hour-24;}while(hour>24);
        cout<<"time="<<hour<<": "<<min<<": "<<sec<<endl;
        sales::putdata();
    }
};

int main()
{
    cout<<"\tBuild Base Class\n";

```

```

publication k;    //Створення базового класу
cout<<"\tBuild Book Class\n";
book l;          //Створення класу book
l.getdata();

cout<<"\tBuild Type Class\n";
type j;          //Створення класу type
j.getdata();
cout<<"-----\n";

l.putdata();
cout<<"-----\n";
j.putdata();

return 0;
}

```

## Результат:

```

C:\Qt\Qt5.12.9\Tools\QtCreator\bin\qtcreator_process_stub.exe
      Build Base Class
      Build Book Class
Book name
Virt
Cost=
7895
Number page
555
Sales mount1=856
Sales mount2=4584
Sales mount3=55624
      Build Type Class
Book name
Next
Cost=
445
Sales mount1=789
Sales mount2=456
Sales mount3=123
Enter hour= 24
Enter minutes= 60
Enter seconds= 300
-----
NAME=Virt
Cost=7895
Number page555
Sales mount1=856
Sales mount2=4584
Sales mount3=55624
-----

```

```

C:\Qt\Qt5.12.9\Tools\QtCreator\bin\qtcreator_process_stub.exe
445
Sales mount1=789
Sales mount2=456
Sales mount3=123
Enter hour= 24
Enter minutes= 60
Enter seconds= 300
-----
NAME=Virt
Cost=7895
Number page555
Sales mount1=856
Sales mount2=4584
Sales mount3=55624
-----
NAME=Next
Cost=445
time=1:4:60
Sales mount1=789
Sales mount2=456
Sales mount3=123

```

# Завдання 3:

Створити клас ЧЕТВІРКА ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення. Створити похідний клас КУБІЧНЕ РІВНЯННЯ з полями-коефіцієнтами. Визначити конструктори за замовчуванням і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення коренів рівняння.

## Код програми

```
#include <iostream>
#include <math.h>
using namespace std;

class fournumber
{
protected:
    double a,b,c,d;
public:
    fournumber(): a(0),b(0),c(0),d(0)
    {}
    fournumber(double _a,double _b,double _c,double _d): a(_a),b(_b),c(_c),d(_d)
    {}
    void putA()
    {
        cout<<"a=";
        cin>>a;
    }
    void putB()
    {
        cout<<"b=";
        cin>>b;
    }
    void putC()
    {
        cout<<"c=";
        cin>>c;
    }
    void putD()
    {
        cout<<"d=";
        cin>>d;
    }
    double getA()
    {
        return a;
    }
    double getB()
    {
        return b;
    }
    double getC()
    {
        return c;
    }
    double getD()
    {
        return d;
    }
}
```

```

    }
    };

class kub : fournumber
{
private:
    double x;
public:
    kub() : fournumber(), x(0)
    {}

    kub(double _x) : fournumber(), x(_x)
    {}

    kub(double _A, double _B, double _C, double _D, double
_x) : fournumber(_A, _B, _C, _D), x(_x)
    {}

    void putX()
    {
        cout<<"x=";
        cin>>x;
    }
    double getX()
    {
        return this->x;
    }
    void math()
    {
        double _a, _b, _c, _d, s;
        _a=fournumber::getA();
        _b=fournumber::getB();
        _c=fournumber::getC();
        _d=fournumber::getD();
        s=( _a*pow(x,3) )+( _b*pow(x,2) )+( _c*x)+d;
        cout<<"Kub="<<s<<endl;
    }

};

int main()
{

    fournumber k;

    kub l(1,1,1,1,1);
    l.math();

    kub(1);
    k.putA();
    k.putB();
    k.putC();
    k.putD();
    l.math();

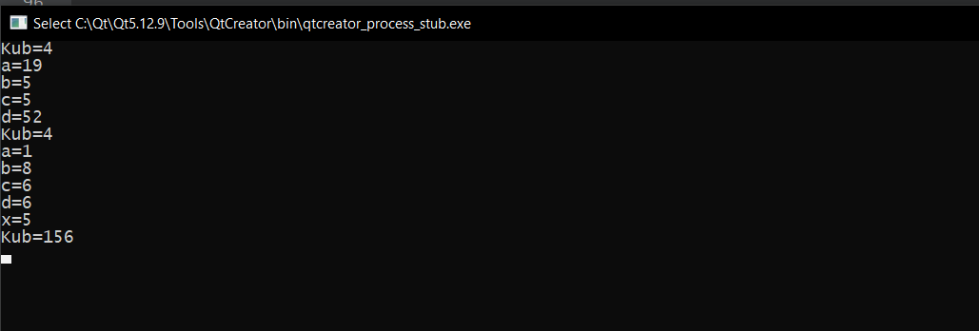
    kub();
    k.putA();
    k.putB();
    k.putC();
    k.putD();
    l.putX();
    l.math();

```



```
    return 0;  
}
```

## Результат:



```
Select C:\Qt\Qt5.12.9\Tools\QtCreator\bin\qtcreator_process_stub.exe  
Kub=4  
a=19  
b=5  
c=5  
d=52  
Kub=4  
a=1  
b=8  
c=6  
d=6  
x=5  
Kub=156  
■
```

## Висновок:

Ознайомився зі способами та механізмами успадкування класів та навчився використовувати їх для побудови об'єктно-орієнтованих програм.