

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕХНІЧНИЙ КОЛЕДЖ ТНТУ ІМЕНІ ІВАНА ПУЛЮЯ  
ВІДДІЛЕННЯ ЕЛЕКТРОННИХ АПАРАТІВ  
Циклова комісія програмних систем і комплексів

## **ЗВІТ**

про виконання лабораторних робіт з дисципліни:  
**«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

Студента 3 курсу групи КН-321  
спеціальності 122 «Комп'ютерні науки»

Попович О.В.  
(прізвище та ініціали)

Перевірив: Р.О. Слободян  
(підпис)

Тернопіль – 2020

# Лабораторна робота №4

**Тема:** Успадковування класів

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

## *Завдання 1.*

**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 `type`, 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 `type`. 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>
using namespace std;

class publication
{
protected:
    string name;
    float price;
};

class book : public publication
{
private:
    int numberOfPages;
public:
```

```

void getdata()
{
    cout << "Enter a book name " << endl;
    cin >> this->name;
    cout << "Enter a book price" << endl;
    cin >> this->price;
    cout << "Enter the number of pages in the book" << endl;
    cin >> this->numberOfPages;
}
void putdata()
{
    cout << "Book name --> " << name << endl;
    cout << "Book price --> " << price << endl;
    cout << "Number of pages in the book --> " << numberOfPages << endl;
}
};

class type : public publication
{
private:
    float recordingTime;
public:
    void getdata()
    {
        cout << "Enter an audio recording name "<<endl;
        cin >> this->name;
        cout << "Enter an audio recording price" << endl;
        cin >> this->price;
        cout << "Enter the recording time books in minutes"<<endl;
        cin >> this->recordingTime;
    }
    void putdata()
    {
        cout << "Audio recording name --> " <<name<<endl;
        cout << "Audio recording price --> " <<price<<endl;
        cout << "Recording time books in minutes --> " << recordingTime <<endl;
    }
};

int main()
{
    book myBook;
    type myType;

    myBook.getdata();
    cout << endl;
    myType.getdata();
    cout << endl;
    cout << "-----"<<endl;
    myBook.putdata();
    myType.putdata();
    cout << "-----" << endl;
    return 0;
}

```

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ**

```
Microsoft Visual Studio Debug Console

Enter a book name
C++_OOP
Enter a book price
1000
Enter the number of pages in the book
985

Enter an audio recording name
Audio_C++_OOP
Enter an audio recording price
1500
Enter the recording time books in minutes
870

-----
Book name --> C++_OOP
Book price --> 1000
Number of pages in the book --> 985
Audio recording name --> Audio_C++_OOP
Audio recording price --> 1500
Recording time books in minutes --> 870
-----

C:\Users\ASUS\source\repos\ConsoleApplication23\Debug\
```

## Завдання 2.

**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>
using namespace std;

class sales
```

```

{
protected :
    const int size = 3;
    float *arr;
public:
    sales()
    {arr = new float[size];}

    void getdata()
    {
        cout << "Enter price last three book";
        cin >> arr[0]>>arr[1]>>arr[2];
    }

    void putdata()
    {
        cout << arr[0] << " " << arr[1] << " " << arr[2] << endl;
    }
};

class publication
{
protected:
    string name;
    float price;
};

class book : public publication, public sales
{
private:
    int numberOfPages;
public:
    book()
    {
        arr = new float[size];
    }
    void getdata()
    {
        cout << "Enter a book name " << endl;
        cin >> this->name;
        cout << "Enter price last three book";
        cin >> arr[0] >> arr[1] >> arr[2];
    }
    void putdata()
    {
        cout << "Book name --> " << name << endl;
        cout << "Price last three book --> ";
        cout << arr[0] << " " << arr[1] << " " << arr[2] << endl;
    }
};

class type : public publication, public sales
{
private:
    float recordingTime;
public:
    type()
    {

```

```

        arr = new float[size];
    }
    void getdata()
    {
        cout << "Enter an audio recording name "<<endl;
        cin >> this->name;
        cout << "Enter price last three audio";
        cin >> arr[0] >> arr[1] >> arr[2];
    }
    void putdata()
    {
        cout << "Audio recording name --> " <<name<<endl;
        cout << "Price last three audio --> ";
        cout << arr[0] << " " << arr[1] << " " << arr[2] << endl;
    }
};

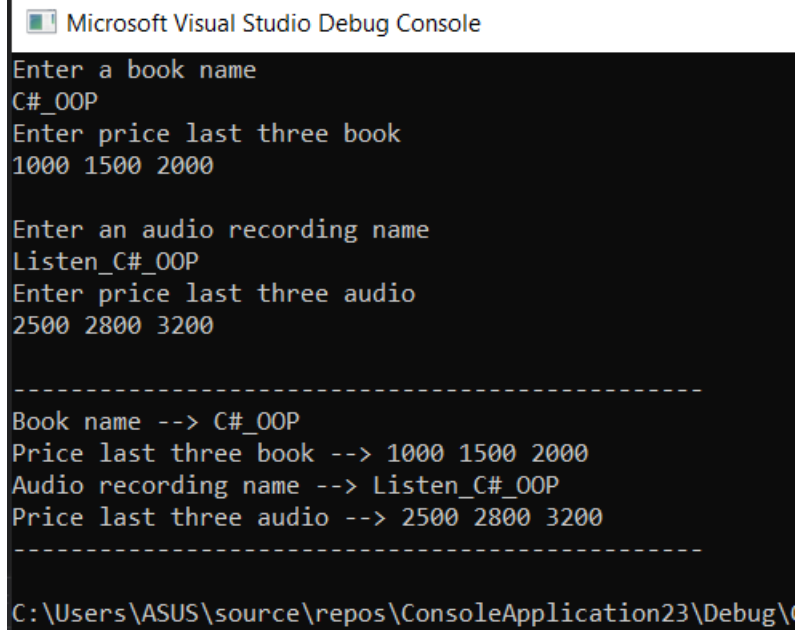
int main()
{
    book myBook;
    type myType;

    myBook.getdata();
    cout << endl;
    myType.getdata();
    cout << endl;
    cout << "-----"<<endl;
    myBook.putdata();
    myType.putdata();
    cout << "-----" << endl;

    return 0;
}

```

## РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ



```

Microsoft Visual Studio Debug Console

Enter a book name
C#_OOP
Enter price last three book
1000 1500 2000

Enter an audio recording name
Listen_C#_OOP
Enter price last three audio
2500 2800 3200

-----
Book name --> C#_OOP
Price last three book --> 1000 1500 2000
Audio recording name --> Listen_C#_OOP
Price last three audio --> 2500 2800 3200
-----

C:\Users\ASUS\source\repos\ConsoleApplication23\Debug\

```

## Завдання 3.

3. Створити клас ПАРА ЦІЛИХ ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення та додавання пар за формулою:  $(a,b)+(c,d)=(a+c,b+d)$ . Створити похідний клас БАГАТОРОЗРЯДНЕ ЧИСЛО з полями: старша частина, молодша частина. Визначити конструктори за замовчуванням і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, додавання, віднімання та множення багаторозрядних чисел.

### КОД ПРОГРАМИ

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

class PairOfIntegers
{
private:
    int a, b, c, d;
public:
    PairOfIntegers() {}
    PairOfIntegers(int a, int b, int c, int d)
    {
        this->a = a;
        this->b = b;
        this->c = c;
        this->d = d;
    }
    int getA() { return a; } void setA() { this->a=a; }
    int getB() { return b; } void setB() { this->b = b; }
    int getC() { return c; } void setC() { this->c = c; }
    int getD() { return d; } void setD() { this->d = d; }

    void setValue()
    {
        cout << "A = "; cin >> a; cout << endl;
        cout << "B = "; cin >> b; cout << endl;
        cout << "C = "; cin >> c; cout << endl;
        cout << "D = "; cin >> d; cout << endl;
    }

    void addPair()
    {
        string s = "2.2";
        int numberD1 = 0, numberD2 = 0;
        int bClone = b, dClone = d;
        while (bClone !=0)
        {
            bClone/= 10;
            numberD1++;
        }
        while (dClone != 0)
        {
```

```

        dClone /= 10;
        numberD2++;
    }

    double numberDividedB = pow(10, numberD1);
    double numberDividedD = pow(10, numberD2);

    double dividedB = b / numberDividedB;
    double dividedD = d / numberDividedD;
    double res1 = (a + c) + (dividedB+ dividedD);
    cout << "(a, b) + (c, d) --> " << res1 << endl;
    cout << "(a + c, b + d) --> " << a + c << "." << d + b << endl;
}

~PairOfIntegers()
{
}

};

class MultiDigitsNumber : public PairOfIntegers
{
private:
    short int seniorPart = 0;
    short int juniorPart = 0;
public:

    MultiDigitsNumber(short int seniorPart, short int juniorPart)
    {
        this->seniorPart = seniorPart;
        this->juniorPart = juniorPart;
    }
    MultiDigitsNumber() {}

    short int getSeniorPart() { return seniorPart; }
    short int getJuniorPart() { return juniorPart; }

    void setSeniorPart(short int seniorPart) { this->seniorPart = seniorPart; }
    void getJuniorPart(short int juniorPart) { this->juniorPart = juniorPart; }

    void setValue()
    {
        cout << "Senior part = " ; cin >> seniorPart; cout << endl;
        cout << "Junior part = " ; cin >> juniorPart; cout << endl;
    }

    long long oneNumber()
    {
        short int cloneJuniorPart = juniorPart;
        int digitNumber = 0;

        while (cloneJuniorPart !=0)
        {
            cloneJuniorPart /= 10;
            digitNumber++;
        }

        long long int Unumber = (long long )seniorPart * pow(10, digitNumber) +
(long long)juniorPart;
    }

```



```

        return Unumber;
    }

    void addSomeNumber()
    {
        long long Unumber = oneNumber();
        short int value = 0;
        cout << "Input number ";
        cin >> value; cout << endl;
        cout << "-----> " << seniorPart << juniorPart << " + " << value << " = " <<
Unumber + (long long)value << endl;
        cout << "-----" << endl;
    }
    void subtractSomeNumber()
    {
        long long Unumber = oneNumber();
        short int value = 0;
        cout << "Input number ";
        cin >> value; cout << endl;
        cout << "-----> " << seniorPart << juniorPart << " - " << value << " = "
<< Unumber - (long long)value << endl;
        cout << "-----" << endl;
    }
    void multiplySomeNumber()
    {
        long long Unumber = oneNumber();
        short int value = 0;
        cout << "Input number ";
        cin >> value; cout << endl;
        cout << "-----> " << seniorPart << juniorPart << " * " << value << " = "
<< Unumber * (long long)value << endl;
        cout << "-----" << endl;
    }

    ~MultiDigitsNumber() {}
};

int main()
{
    PairOfIntegers p;
    p.setValue();
    p.addPair();
    cout << endl << "-----" << endl;

    MultiDigitsNumber m;
    m.setValue();
    m.addSomeNumber();
    m.subtractSomeNumber();
    m.multiplySomeNumber();

    return 0;
}

```

## РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

```
Microsoft Visual Studio Debug Console

A = 160

B = 212

C = 129

D = 139

(a, b) + (c, d) --> 289.351
(a + c, b + d) --> 289.351

-----
Senior part = 19

Junior part = 200

Input number 1999

-----> 19200 + 1999 = 21199
-----
Input number 15784

-----> 19200 - 15784 = 3416
-----
Input number 19

-----> 19200 * 19 = 364800
-----

C:\Users\ASUS\source\repos\ConsoleApplication1
```

**Висновок:** ознайомився зі способами та механізмами успадкування класів та навчився використовувати їх для побудови об'єктно-орієнтованих програм.