Лабораторна робота №4

Тема: Успадковування класів

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

Хід роботи

1.Ознайомився з матеріалом

2.Написав програму згідно наступного 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().

#include <iostream>

#include <string>

using namespace std;

class Publication

{

public:

void getData()

{

cout << "Name:\t" << name << "\nPrice:\t" << price << " grn" << endl;

}

void putData()

{

cout << "Name:\t";

cin >> this->name;

cout << "Price:\t";

cin >> this->price;

}

private:

string name;

float price;

};

class Book :public Publication

{

public:

void getData()

{

Publication::getData();

cout << "Pages:\t" << pages << endl;

}

void putData()

{

Publication::putData();

cout << "Pages:\t";

cin >> this->pages;

}

private:

int pages;

};

class Type :public Publication

{

public:

void getData()

{

Publication::getData();

cout << "Min:\t" << min << endl;

}

void putData()

{

Publication::putData();

cout << "Min:\t";

cin >> this->min;

}

private:

int min;

};

int main()

{

Book book;

book.putData();

Type type;

type.putData();

book.getData();

type.getData();

return 0;

}



3.Проаналізував результати роботи в прикладі.

4. Для заданого індивідуального завдання написати програму з використанням наслідування.

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.

#include <iostream>

using namespace std;

class PairNum

{

protected:

int a, b, c, d;

public:

PairNum() : a(1), b(2), c(3), d(4)

{}

PairNum(int a, int b, int c, int d) : a(a), b(b), c(c), d(d)

{}

~PairNum()

{}

void PairSum()

{

cout << "a+c= " << a + c << endl;

cout << "b+d= " << b + d << endl;

}

};

class MultiNum:PairNum

{

protected:

int olderPart1 = a \* 10;

int yongerPart1 = b;

int olderPart2 = c \* 10;

int yongerPart2 = d;

public:

MultiNum(int a, int b, int c, int d) : PairNum(a,b,c,d)

{}

MultiNum() :PairNum()

{}

int GetOlderPart1() {

return olderPart1;

}

void SetOlderPart1(int n)

{

olderPart1 = n;

}

int GetOlderPart2() {

return olderPart2;

}

void SetOlderPart2(int n)

{

olderPart2= n;

}

int GetYongerPart1() {

return yongerPart1;

}

void SetYongerPart1(int n)

{

yongerPart1 = n;

}

int GetYongerPart2() {

return yongerPart2;

}

void SetYongerPart2(int n)

{

yongerPart2 = n;

}

void ShowNum()

{

cout << "num1=" << olderPart1+yongerPart1 << endl;

cout << "num2=" << olderPart2+yongerPart2 << endl;

}

int Multiplie()

{

int multi = (olderPart1 + yongerPart1) \* (olderPart2 + yongerPart2);

return multi;

}

int Add()

{

int add = (olderPart1 + yongerPart1) + (olderPart2 + yongerPart2);

return add;

}

int Minus()

{

int min = (olderPart1 + yongerPart1) - (olderPart2 + yongerPart2);

return min;

}

};

int main()

{

PairNum pairNum(2, 4, 6, 8);

pairNum.PairSum();

PairNum pairNumDef;

pairNumDef.PairSum();

MultiNum multiNum(3, 3, 2, 2);

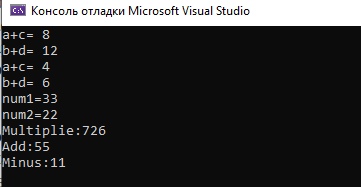
multiNum.ShowNum();

cout << "Multiplie:" << multiNum.Multiplie() << endl;

cout << "Add:" << multiNum.Add() << endl;

cout << "Minus:" << multiNum.Minus() << endl;

}



Висновки: на цій лабораторній роботі я ознайомився зі способами та механізмами успадкування класів та навчився використовувати їх для побудови об’єктно-орієнтованих програм.