//Использование указателей на объекты классов l7

#include <iostream.h>

class CBox //Базовый класс

{

public:

//Функция для отображения объема объекта

void ShowVolume() const

{

cout << endl <<"Volume of CBox equals " << Volume();

}

**//Функция вычисления объема объекта CBox**

**virtual double Volume() const**

**{ return m\_Length\* m\_Width\* m\_Height; }**

//Конструктор

CBox(double lv = 1., double wv = 1.0, double hv = 1.0)

: m\_Length(lv), m\_Width(wv), m\_Height(hv) { }

protected:

double m\_Length;

double m\_Width;

double m\_Height;

};

class CGlassBox : public CBox //Производный класс

{

public:

**//функция производного классадля вычисления объема CGlassBox резервирующая 15% на упаковку**

**virtual double Volume()const**

**{ return 0.85\*m\_Length\* m\_Width\* m\_Height; }**

//Конструктор

CGlassBox (double lv, double wv, double hv):CBox(lv, wv, hv) { }

};

//Использование указателя базового класса для вызова виртуальной функции

#include <iostream.h>

int main()

{

CBox myBox (2.0, 3.0, 4.0); //Объявление базового ящика

CGlassBox myGlassBox(2.0, 3.0, 4.0); //Объявление произв.ящика

**CBox\* pBox = 0;//Объявление указателя на объекты базового класса**

**pBox = &myBox; //Установить указатель на адрес базового объекта**

**pBox ->ShowVolume(); //Отобразить объем базового ящика**

**pBox = &myGlassBox; //Установить указатель на адрес объекта //произв. класса**

**pBox ->ShowVolume(); //Отобразить объем производного ящика**

cout << endl;

return 0;

}

Результат

Volume of CBox equals 24

Volume of CBox equals 20.4

//Использование ссылок с виртуальными функциями l8

#include <iostream.h>

class CBox //Базовый класс

{

public:

//Функция для отображения объема объекта

void ShowVolume() const

{

cout << endl <<"Volume of CBox equals " << Volume();

}

**//Функция вычисления объема объекта CBox**

**virtual double Volume() const**

**{ return m\_Length\* m\_Width\* m\_Height; }**

//Конструктор

CBox(double lv = 1., double wv = 1.0, double hv = 1.0)

: m\_Length(lv), m\_Width(wv), m\_Height(hv) { }

protected:

double m\_Length;

double m\_Width;

double m\_Height;

};

class CGlassBox : public CBox //Производный класс

{

public:

**//функция производного классадля вычисления объема CGlassBox резервирующая 15% на упаковку**

**virtual double Volume()const**

**{ return 0.85\*m\_Length\* m\_Width\* m\_Height; }**

//Конструктор

CGlassBox (double lv, double wv, double hv):CBox(lv, wv, hv) { }

};

//Использование ссылки для вызова виртуальной функции

**void Output (const CBox& aBox); //Прототип функции**

int main()

{

CBox myBox (2.0, 3.0, 4.0); //Объявление базового ящика

CGlassBox myGlassBox(2.0, 3.0, 4.0); //Объявление произв. ящика

Output (myBox); //Вывод объема объекта базового класса

Output (myGlassBox); //Вывод объема объекта произв. класса

cout << endl;

return 0;

}

**void Output (const CBox& aBox)**

**{**

**aBox.ShowVolume();**

**}**

Результат

Volume of CBox equals 24

Volume of CBox equals 20.4

