**//Чистые виртуальные функции L9**

const double PI = 3.14159265; //Глобальное определение PI

#include <iostream.h>

class CContainer //Общий базовый класс для специфических контейнеров

{

public:

**//Функция для вычисления объема - без содержимого**

**//Это определение чистой ('pure') виртуальной функции, на что указывает '=0'**

**virtual double Volume() const = 0;**

//Функция для отображения объема

virtual void ShowVolume() const

{ cout << endl

<<"Volume equals" << Volume();

}

};

class CBox : public CContainer //Производный класс

{

public:

//Функция для отображения объема объекта

virtual void ShowVolume() const

{ cout << endl

<<"Volume CBox equals" << Volume();

}

//Функция для вычисления объема объекта CBox

**virtual double Volume() const**

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

//Конструктор

CBox (double lv = 1.0,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 CCan : public CContainer

{

public:

//Функция вычисления объема банки

**virtual double Volume() const**

**{return 0.25\*PI\*m\_Diameter\*m\_Diameter\*m\_Height;}**

//Конструктор

CCan (double hv = 4.0, double dv = 2.0)

:m\_Height(hv), m\_Diameter(dv) {}

protected:

double m\_Height;

double m\_Diameter;

};

int main(void)

{

//указатель на абстрактный базовый класс,

//инициализированный адресом объекта CBox

CContainer\* pC1 = new CBox(2.0, 3.0, 4.0);

//указатель на абстрактный базовый класс,

//инициализированный адресом объекта CCan

CContainer\* pC2 = new CCan(6.5, 3.0);

pC1->ShowVolume(); //вывод объемов двух объектов через указатели

pC2->ShowVolume();

cout << endl;

delete pC1; //освободить память в свободном хранилище

delete pC2;

return 0;

}

Volume CBox equals24

Volume equals45.9458