**// Встроенные функции. Конструктор по умолчанию.**

#include <iostream>

class CBox

{

public:

double m\_Length;

double m\_Width;

double m\_Height;

**double Volume();**

CBox( double l, double w, double h)

{

m\_Length=l;

m\_Width=w;

m\_Height=h;

}

**CBox()**

**{}**

};

**inline** double CBox::Volume()

{

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

}

using namespace std;

int main()

{

CBox box1 (78.0,24.0,18.0);

CBox box2; // Использование конструктора по умолчанию

CBox box3 (8.0,5.0,1.0);

cout<<"V box1 ="<< box1.Volume()<<endl;

box2.m\_Height=box1.m\_Height-10;

box2.m\_Length=box1.m\_Length/2;

box2.m\_Width=box1.m\_Length\*0.25;

cout<<"V box2 ="<< box2.Volume()<<endl;

cout<<"V box3 ="<< box3.Volume()<<endl;

return 0;

}

# // Спецификаторы доступа public и private

#include <iostream>

using namespace std;

**class CBox**

{

**private:**

double m\_Length; //Длина ящика в дюймах

double m\_Width; //Ширина ящика в дюймах

double m\_Height; //Высота ящика в дюймах

**public:**

CBox (double l , double w , double h )

{

m\_Length=l;

m\_Width=w;

m\_Height=h;

}

double Volume()

{

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

}

};

int main ()

{

CBox box2(1.0,1.0,1.0); //объявление box2

CBox match(2.2,1.1,0.5); // объявление match

cout <<endl

<<" V match ="

<< match.Volume(); //Вычислить объем match

cout <<endl

<<" V box2 ="

<< box2.Volume();

cout <<endl;

return 0;

}

**// Определение конструктора с использованием списка инициализации**

#include <iostream>

using namespace std;

**class CBox**

**{**

**private:**

double m\_Length; //Длина ящика в дюймах

double m\_Width; //Ширина ящика в дюймах

double m\_Height; //Высота ящика в дюймах

**public:**

CBox(): // Определение конструктора с использованием списка инициализации

m\_Length(1.0),m\_Width(2.0), m\_Height(1.0)

{

}

double Volume()

{

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

}

**};**

int main ()

{

CBox box2;

cout <<endl <<" V box2 =" << box2.Volume();

cout <<endl;

return 0;

}

#include <iostream>

using namespace std;

**class CBox**

**{**

**private:**

double m\_Length; //Длина ящика в дюймах

double m\_Width; //Ширина ящика в дюймах

double m\_Height; //Высота ящика в дюймах

**public:**

CBox():

m\_Length(1.0), m\_Width(2.0), m\_Height(1.0)

{

}

CBox(double l, double w, double h):

m\_Length(l),

m\_Width(w),

m\_Height(h)

{

}

double Volume()

{

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

}

**};**

int main ()

{

CBox box2;

CBox match(2.2,1.1,0.5);

cout <<endl

<<" V match ="

<< match.Volume();

cout <<endl

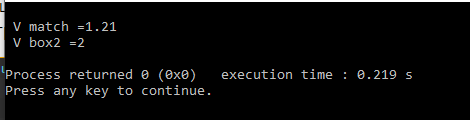
<<" V box2 ="

<< box2.Volume();

cout <<endl;

return 0;

}



**// Доступ к приватным членам класса**

#include <iostream>

using namespace std;

**class CBox**

{

**private:**

double m\_Length; //Длина ящика в дюймах

double m\_Width; //Ширина ящика в дюймах

double m\_Height; //Высота ящика в дюймах

**public:**

CBox():

m\_Length(1.0),m\_Width(2.0), m\_Height(1.0)

{

}

CBox(double l,double w,double h):

m\_Length(l),

m\_Width(w),

m\_Height(h)

{

}

double Volume()

{

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

}

**double GetLength()**

{

return m\_Length;

}

};

int main ()

{

CBox box2; //объявление box2 без инициализации

CBox match(2.2,1.1,0.5); // объявление match

//cout<< box2. m\_Lenght = 4.0; //Раскомментировать следующую строку для получения ошибки

cout<<"m\_Length : " <<box2.**GetLength**()<<endl;

cout <<endl

<<" V match ="

<< match.Volume();

cout <<endl

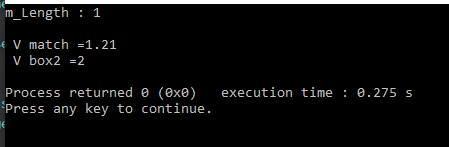
<<" V box2 ="

<< box2.Volume();

cout <<endl;

return 0;

}



**// Дружественные функции класса**

#include <iostream>

using namespace std;

class CBox //Определение класса в глобальном контексте

{

private:

double m\_Length; //Длина ящика в дюймах

double m\_Width; //Ширина ящика в дюймах

double m\_Height; //Высота ящика в дюймах

public:

CBox():

m\_Length(1.0),m\_Width(2.0), m\_Height(1.0)

{

}

CBox(double l,double w,double h):

m\_Length(l),

m\_Width(w),

m\_Height(h)

{

}

double Volume()

{

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

}

double GetLength()

{

return m\_Length;

}

**//Дружественная функция**

friend double BoxSurface(CBox fbox);

};

double BoxSurface(CBox fbox)

{

return 2.0\*(fbox.m\_Length\*fbox.m\_Width+

fbox.m\_Width\*fbox.m\_Height+

fbox.m\_Height\*fbox.m\_Length);

}

int main ()

{

CBox box2;//(1.0,1.0,1.0); //объявление box2 без инициализации

CBox match(2.2,1.1,0.5); // объявление match

//cout<< box2. m\_Lenght = 4.0;

cout<<"m\_Length : " <<box2.GetLength()<<endl;

cout <<endl

<<" V match ="

<< match.Volume(); //Вычислить объем match

cout <<endl

<<" Surface match ="

<< BoxSurface(match);

//Раскомментировать следующую строку для получения ошибки

cout <<endl

<<" V box2 ="

<< box2.Volume();

cout <<endl

<<" Surface box2 ="

<< BoxSurface(box2);

cout <<endl;

return 0;

}

