## Практические задания для Урока 6 Модуля 3

1. Создайте класс Average с полями double m\_value, int count с доступом private. Создайте конструктор, который инициирует поля m\_value, count нулями, создайте метод add который увеличивает m\_value на получаемый аргумент, count увеличивает на 1, создайте метод getValue, который возвращает частное m\_value и count.

Для класса Average создайте дружественную функцию Reset, которая присваивает m\_value и int count нули.

Напишите программу, которая создает объект класса Average, вызовите у объекта два раза метод add (с параметром 5 и с параметром 10). Выведите в отдельной строке результат вызова метода getValue. Вызовите функцию reset. Вызовите у объекта два раза метод add (с параметром 2 и с параметром 3). Выведите в отдельной строке результат вызова метода getValue.

1. Создайте класс Rectangle с полями width, height с доступом private. Создайте конструктор с аргументами, который инициирует поля переданными значениями аргументов. Создайте дружественный класс Area, для которого создайте метод getValue, который возвращает значение площади для объекта класса Rectangle.

Напишите программу, которая запрашивает ввод с клавиатуры ширины и длины прямоугольника в отдельных строках, создает объект класса Rectangle и инициализирует его введенными значениями. Создайте объект класса Area, выведите на экран результат работы метода getValue.

1. Создайте базовый класс Avto с защищенными полями char brand[128] и char name[128]. Создайте конструктор класса, который инициирует данные поля получаемыми параметрами.

Создайте класс Car- наследник класса Avto, с дополнительным полем speed с правами доступа private, создайте конструктор класса, который будет инициализировать поле speed получаемым параметром. Создайте метод класса info для вывода информации на экран в формате: “Марка:brand Модель: Скорость:speed”.

Напишите программу, которая будет запрашивать ввод с клавиатуры (в отдельных строках) значение марки, модели и скорости. Создайте объект класса Car, инициализируйте его введенными с клавиатуры значениями. Вызовите у объекта класса Car метод info.

## Решения.

1.

#include <iostream>

using namespace std;

class Average

{

private:

double m\_value;

int count;

public:

Average() { m\_value = 0; count=0;}

void add(int value) { m\_value += value; count++;}

double getValue(){return m\_value/count;}

// Сделаем функцию reset() другом этого класса

friend void reset(Average &average);

};

// reset() теперь является другом класса Average

void reset(Average &average)

{

// и может получить доступ к закрытым данным объектов Average

average.m\_value = 0;

average.count = 0;

}

int main()

{

Average a;

a.add(5);

a.add(10);

cout<<a.getValue()<<endl;

reset(a);// сбрасываем накапливающий сумматор в 0

a.add(2);

a.add(3);

cout<<a.getValue()<<endl;

return 0;

}

2.

#include <iostream>

using namespace std;

class Rectangle {

int width, height;

public:

Rectangle (int x, int y)

{width=x; height=y;}

friend class Area;

};

class Area

{

public:

int getValue(Rectangle &r) {return r.width \* r.height;}

};

int main () {

int w,h;

cin>>w;

cin>>h;

Rectangle bar (w,h);

Area a;

cout << a.getValue(bar) << '\n';

return 0;

}

3.

#include <iostream>

#include <cstring>

using namespace std;

class Avto

{

public:

Avto(char b[], char n[]) { strcpy(brand,b);strcpy(name,n);}

protected:

char brand[128];

char name[128];

};

class Car: public Avto

{

int speed;

public:

Car(char b[], char n[], int s):Avto(b,n) {speed=s;};

void info() {cout<<"Марка:"<<brand<<" Модель:"<<name<<" Скорость:"<<speed;};

};

int main()

{

char brand[128];

char name[128];

int speed;

cin>>brand;

cin>>name;

cin>>speed;

Car a(brand,name,speed);

a.info();

}