16 марта, Повышев таки пришёл.

struct point {  
int x,y;

}

void print (point x)

упрощаем

struct point {

int x,y;

void print();

}

void point::print() {  
}

point a;

a.x = 100;

Класс — пользовательский тип

class point {

x,y;

public:

void print()

{

/\*\*/

}

void set x (int \_x) {

x = \_x;

}

bool set y (int \_y) {

if((\_y > 10)||(\_y < -10)) return 1;

y = \_y;

return 0;

}

get x() {

return x;

}

get y() {

return y;

}

}

/\*

public

private

protected

\*/

point a,b;

a.setx(3);

b.sety(5);

int k = a.getx();

k = b.getx();

/\*\*\*/

class x {

int k;

public:

void set(int a) {

this->k = a;

}

}

x \* const this;

x b;

b.set(3);

/\*\*\*/

class Q {

Q \* p;

Q \* s;

public:

void append (Q \* t)

{

t -> s = s;

t -> p = this;

s -> p = p;

s = p;

}

/\*\*/

class C1 {

int k;

public:

void set(int x) {  
k = x;

}

int get() const {

return k;

};

}

void m (c1& t1, const c1& t2)

{

t1.set(5);

int I = t1.get;

t2.set(3); //нельзя ибо константа

int j = t2.get();  
}

/\*\*/

class c2

{

int x;

public:

void s1(int i) const

{x =()}; //низя

void s2(int i) const {

((c2\*)this)->x=i;

}

}

//ЗАЧЕМ ЭТО НАДО, ЗА ЧТО НАС УЧАТ ГОВНОКОДИТЬ

struct p { int a; };

class c3 {  
p\* x;

public:

void s(int i) const

{

x ->a = i;

}

}

Конструктор — член класса, отвечающий за первичную инициализацию данных.

class c3

{

int a,b,c;

public:

c3(int \_a, int \_b, int \_c) {

a = \_a ...

}

с3(int x) {

a = x; b = x; c = c+3;

}

c3(){a=0;b=0;c=0;}

c3(const c3& t);

}

с3 t1 = c3(5,n,3);

c3 t2(3);

c3 t3;

c3 t4 = t1;

c3\* t5 = new c3(5);

c3\* t6 = t5;

c3\* t7 = new c3(t5);

/\*\*/

class c4 {

p \* x;

public;

c4 (int i) { };

~c4() {delete x;}

}

class c5

{

int a;

public:

/\*

void m()

{a++;}  
\*/

inline void c5::m() {

a++;

}

}

c5 t;

t.m();

///\*

class point {

}

class vector {

}

 friend voidmove()

friend voidmove()

void move(point &, vector &)