

```
1  /*
2  * Proyecto: ProyectoRectanguloMetodosSelectores
3  * Archivo: Rectangulo.h
4  * Autor: J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 7 de mayo de 2024, 08:16 AM
7  */
8
9
10 #ifndef RECTANGULO_H
11 #define RECTANGULO_H
12
13 #include <cmath>
14
15
16 class Rectangulo {
17 private:
18     char *nombre;
19     double base;
20     double altura;
21 public:
22     Rectangulo();
23     Rectangulo(const char*, double=0.0, double=0.0 );
24     Rectangulo(const class Rectangulo &rect);
25
26     virtual ~Rectangulo();
27     void SetAltura(double altura);
28     double GetAltura() const;
29     void SetBase(double base);
30     double GetBase() const;
31     double area();
32
33     void SetNombre(const char* nombre);
34     void GetNombre(char*) const;
35     void operator =(const class Rectangulo &rect);
36 };
37
38 #endif /* RECTANGULO_H */
39
40 /*
41 * Proyecto: ProyectoRectanguloMetodosSelectores
42 * Archivo: Rectangulo.cpp
43 * Autor: J. Miguel Guanira E. //miguel.guanira.
44 *
45 * Created on 7 de mayo de 2024, 08:16 AM
46 */
47
48 #include <iostream>
49 #include <iomanip>
50 using namespace std;
51 #include <cstring>
52 #include "Rectangulo.h"
53
54 Rectangulo::Rectangulo() {
55     cout<<"Constructor Defecto"<<endl;
56     nombre=nullptr;
57 }
58
59 Rectangulo::Rectangulo(const char*nom, double b, double h ) {
60     cout<<"Constructor con parametros"<<nom<<endl;
61     nombre = nullptr;
62     base = b;
63     altura = h;
64     SetNombre(nom);
65 }
66
```

```
67 Rectangulo::Rectangulo(const class Rectangulo& rect) {
68     nombre = nullptr;
69     *this = rect;
70 }
71
72 Rectangulo::~Rectangulo() {
73     cout<<"Ejecuto el destructor ";
74     if(nombre!=nullptr) {
75         cout<<nombre<<endl;
76         delete nombre;
77     }
78     else cout<<"Vacio:"<<endl;
79 }
80
81 void Rectangulo::SetAltura(double alt) {
82     altura = alt;
83 }
84
85 double Rectangulo::GetAltura() const {
86     return altura;
87 }
88
89 void Rectangulo::SetBase(double base) {
90     this->base = base;
91 }
92
93 double Rectangulo::GetBase() const {
94     return base;
95 }
96
97 double Rectangulo::area() {
98     return base*this->altura;
99 }
100
101
102 void Rectangulo::SetNombre(const char* cad) {
103     if(nombre!=nullptr)delete nombre;
104     nombre = new char[strlen(cad)+1];
105     strcpy(nombre,cad);
106 }
107
108 void Rectangulo::GetNombre(char*cad) const {
109     if(nombre ==nullptr)cad[0]=0;
110     strcpy(cad,nombre);
111 }
112
113 void Rectangulo::operator=(const class Rectangulo& rect) {
114     // base = rect.GetBase();
115     char cadena[100];
116     base = rect.base;
117     rect.GetNombre(cadena);
118     SetNombre(cadena);
119     altura = rect.altura;
120 }
121
122
123 /*
124  * Proyecto: ProyectoRectanguloMetodosSelectores
125  * Archivo: main.cpp
126  * Autor: J. Miguel Guanira E.//miguel.guanira.
127  *
128  * Created on 7 de mayo de 2024, 08:15 AM
129  */
130
131 #include <iostream>
132 #include <iomanip>
```

```
133 using namespace std;
134 #include "Rectangulo.h"
135 void f(){
136     class Rectangulo rect("RectanguloF");
137
138 }
139 int main(int argc, char** argv) {
140     class Rectangulo r1("Rect1"), r2("Rect2", 10), r3,
141         r4("Rect4", 1.5, 5.3), *pt;
142     pt = new class Rectangulo[5];
143     f();
144     delete [] pt;
145     // r2.SetAltura(4);
146     // r2.SetBase(12);
147     // cout<<r2.area()<<endl;
148     //
149     // char *n;
150     // r1.SetNombre("Juan");
151     // {}
152     // n = new char [50];
153     // r1.GetNombre(n);
154     // r2.SetNombre(n);
155
156     return 0;
157 }
158
```