

```
1  #include "pch.h"
2  #include "CAppObject.h"
3
4  // Public constructors
5  CAppObject::CAppObject(CRect bounds, CString ID, BOOL active)
6  {
7      this->bounds = bounds;
8      this->ID = ID;
9      this->active = active;
10 }
11 CAppObject::~CAppObject()
12 {
13 }
14
15 void CAppObject::draw(CDC* pDc)
16 {
17 }
18
19 // Public message handlers
20 void CAppObject::OnSize(UINT nType, int cx, int cy)
21 {
22 }
23
24 BOOL CAppObject::OnLButtonUp(UINT nFlags, CPoint point)
25 {
26     return 0;
27 }
28 BOOL CAppObject::OnLButtonDown(UINT nFlags, CPoint point)
29 {
30     return 0;
31 }
32 void CAppObject::OnLButtonDblClk(UINT nFlags, CPoint point)
33 {
34 }
35 void CAppObject::OnRButtonUp(UINT nFlags, CPoint point)
36 {
37 }
38 void CAppObject::OnRButtonDown(UINT nFlags, CPoint point)
39 {
40 }
41 BOOL CAppObject::OnMouseMove(UINT nFlags, CPoint point)
42 {
43     return TRUE;
44 }
45 int CAppObject::OnRecieveText(CString text)
46 {
47     return 0;
48 }
49 void CAppObject::OnRecieveBackspace()
50 {
51 }
52 void CAppObject::OnRecieveReturn()
53 {
54 }
55 void CAppObject::OnRecieveTab()
56 {
```

```
57 }
58
59 void CAppObject::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
60 {
61 }
62
63 BOOL CAppObject::OnContextMenu(CWnd* pWnd, CPoint point)
64 {
65     return 0;
66 }
67
68 // Public getters, setters and checkers
69 CString CAppObject::getID()
70 {
71     return this->ID;
72 }
73 void CAppObject::setID(CString ID)
74 {
75     this->ID = ID;
76 }
77
78 CRect CAppObject::getBounds()
79 {
80     return this->bounds;
81 }
82 BOOL CAppObject::checkPointInBounds(CPoint point)
83 {
84     if (this->bounds.left <= point.x && bounds.right >= point.x &&
85         this->bounds.top <= point.y && this->bounds.bottom >=
86             point.y) {
87         return TRUE;
88     }
89     return FALSE;
90 }
91
92 void CAppObject::setBounds(CRect bounds)
93 {
94     this->bounds = bounds;
95 }
96 void CAppObject::setBounds(int x, int y, int cx, int cy)
97 {
98     this->bounds.left = x;
99     this->bounds.top = y;
100     this->bounds.right = x + cx;
101     this->bounds.bottom = y + cy;
102 }
103 void CAppObject::setPosition(CPoint point)
104 {
105     this->move(point.x - this->bounds.left, point.y - this->
106         >bounds.top);
107 }
108 void CAppObject::move(int x, int y)
109 {
110     this->bounds.left += x;
111     this->bounds.right += x;
```

---

```
111     this->bounds.top += y;
112     this->bounds.bottom += y;
113 }
114
115 BOOL CAppObject::isActive()
116 {
117     return this->active;
118 }
119 void CAppObject::setActive(BOOL val)
120 {
121     this->active = val;
122 }
123
```