

```
1 #include "StdAfx.h"
2 #include "MenuBase.h"
3
4 extern int LibMessage(LPCTSTR str, int flags = MB_OK);
5
6 void MenuBase::initCoordinates(size_t id, size_t id_x, size_t id_y,
    size_t id_z) {
7     BSTR name = _T("Координаты");
8     this->coordinates = this->collection->Add
        (ControlTypeEnum::ksControlPoint3D);
9     this->coordinates->Name = name;
10    this->coordinates->Hint = name;
11    this->coordinates->Tips = name;
12    this->coordinates->Id = id;
13    IPropertyEditPtr X = this->coordinates->GetCoordinate(0);
14    IPropertyEditPtr Y = this->coordinates->GetCoordinate(1);
15    IPropertyEditPtr Z = this->coordinates->GetCoordinate(2);
16
17    X->Id = id_x;
18    X->Value = this->getModel()->coordinates.X;
19    X->CheckState = CheckStateEnum::ksCheckFixed;
20
21    Y->Id = id_y;
22    Y->Value = this->getModel()->coordinates.Y;
23    Y->CheckState = CheckStateEnum::ksCheckFixed;
24
25    Z->Id = id_z;
26    Z->Value = this->getModel()->coordinates.Z;
27    Z->CheckState = CheckStateEnum::ksCheckFixed;
28 }
29
30 void MenuBase::initSurfaceType(size_t id) {
31     BSTR name = _T("Тип основания");
32     this->surfaceType = this->collection->Add
        (ControlTypeEnum::ksControlListStr);
33     this->surfaceType->Name = name;
34     this->surfaceType->Hint = name;
35     this->surfaceType->Tips = name;
36     this->surfaceType->Id = id;
37     this->surfaceType->ReadOnly = true;
38     this->surfaceType->Add(_T("Круг"));
39     this->surfaceType->Add(_T("Треугольник"));
40     this->surfaceType->Add(_T("Квадрат"));
41     this->surfaceType->Add(_T("Пятиугольник"));
42     this->surfaceType->Add(_T("Шестиугольник"));
43     this->surfaceType->Add(_T("Тригон"));
44     this->surfaceType->Add(_T("Ромб 35"));
45     this->surfaceType->Add(_T("Ромб 55"));
46     this->surfaceType->Add(_T("Ромб 80"));
47     this->surfaceType->SetCurrentByIndex(this->getModel()-
        >surfaceType);
48 }
49
50 void MenuBase::initSizeType(size_t id) {
51     BSTR name = _T("Тип размера");
```

```
52     this->sizeType = this->collection->Add
        (ControlTypeEnum::ksControlListStr);
53     this->sizeType->Name = name;
54     this->sizeType->Hint = name;
55     this->sizeType->Tips = name;
56     this->sizeType->Id = id;
57     this->sizeType->ReadOnly = true;
58     this->sizeType->Add(_T("По радиусу описанной окружности"));
59     this->sizeType->Add(_T("По радиусу вписанной окружности"));
60     this->sizeType->Add(_T("По длине стороны"));
61     this->sizeType->SetCurrentByIndex(this->getModel()->sizeType);
62 }
63
64 void MenuBase::initSize(size_t id) {
65     BSTR name = _T("Величина размера");
66     this->size = this->collection->Add
        (ControlTypeEnum::ksControlEditLength);
67     this->size->Name = name;
68     this->size->Hint = name;
69     this->size->Tips = name;
70     this->size->Id = id;
71     this->size->Value = this->getModel()->size;
72 }
73
74 void MenuBase::initHeight(size_t id) {
75     BSTR name = _T("Высота");
76     this->height = this->collection->Add
        (ControlTypeEnum::ksControlEditLength);
77     this->height->Name = name;
78     this->height->Hint = name;
79     this->height->Tips = name;
80     this->height->Id = id;
81     this->height->Value = this->getModel()->height;
82 }
83
84 void MenuBase::initHasHole(size_t id) {
85     BSTR name = _T("Есть отверстие");
86     this->hasHole = this->collection->Add
        (ControlTypeEnum::ksControlTwinSwitcher);
87     this->hasHole->Name = name;
88     this->hasHole->Hint = name;
89     this->hasHole->Tips = name;
90     this->hasHole->Label2 = _T("Нет отверстия");
91     this->hasHole->Id = id;
92     this->hasHole->Value = (this->getModel()->hasHole ? 1 : 2);
93 }
94
95 void MenuBase::initHoleRadius(size_t id) {
96     BSTR name = _T("Радиус отверстия");
97     this->holeRadius = this->collection->Add
        (ControlTypeEnum::ksControlEditLength);
98     this->holeRadius->Name = name;
99     this->holeRadius->Hint = name;
100    this->holeRadius->Tips = name;
101    this->holeRadius->Id = id;
```

```
102     this->holeRadius->Value = this->getModel()->holeRadius;
103 }
104
105 void MenuBase::initAngleAlpha(size_t id) {
106     BSTR name = _T("Угол наклона α");
107     this->angleAlpha = this->collection->Add          ↗
        (ControlTypeEnum::ksControlEditAngle);
108     this->angleAlpha->Name = name;
109     this->angleAlpha->Hint = name;
110     this->angleAlpha->Tips = name;
111     this->angleAlpha->Id = id;
112     this->angleAlpha->Value = this->getModel()->angleAlpha;
113 }
114
115 void MenuBase::initRoundingRadius(size_t id) {
116     BSTR name = _T("Радиус скругления");
117     this->roundingRadius = this->collection->Add      ↗
        (ControlTypeEnum::ksControlEditLength);
118     this->roundingRadius->Name = name;
119     this->roundingRadius->Hint = name;
120     this->roundingRadius->Tips = name;
121     this->roundingRadius->Id = id;
122     this->roundingRadius->Value = this->getModel()->roundingRadius;
123 }
124
125 void MenuBase::initButton(size_t id) {
126     BSTR name = _T("Применить");
127     this->button = this->collection->Add             ↗
        (ControlTypeEnum::ksControlTextButton);
128     this->button->Name = name;
129     this->button->Hint = name;
130     this->button->Tips = name;
131     this->button->Id = id;
132     this->button->Value = this->getModel()->button;
133 }
134
135 void MenuBase::setFieldName(IPropertyEditPtr field, BSTR name) {
136     field->Name = name;
137     field->Hint = name;
138     field->Tips = name;
139 }
140
141 void MenuBase::setFieldActive(IPropertyEditPtr field) {
142     field->ReadOnly = false;
143     field->Enable = true;
144     field->Visible = true;
145 }
146
147 void MenuBase::setFieldInactive(IPropertyEditPtr field) {
148     field->ReadOnly = true;
149     field->Enable = false;
150     field->Visible = false;
151 }
152
153 void MenuBase::checkHasHole() {
154     if (this->hasHole->Value.intVal == 1) {
```

---

```
155         this->setFieldActive(field: this->holeRadius);
156     }
157     else {
158         this->setFieldInactive(field: this->holeRadius);
159     }
160 }
161
```