

```
1  #ifndef _MENU_BASE_H
2  #define _MENU_BASE_H
3
4  #include "StdAfx.h"
5  #include "Resource.h"
6  #include "SMPBase.h"
7
8  #if _MSC_VER > 1000
9  #pragma once
10 #endif
11 #endif
12
13 /// Основной класс меню параметров модели
14 class MenuBase
15 {
16 public:
17     /// Коллекция контролов
18     IPropertyControlsPtr collection;
19     /// Поле "Координаты"
20     IPropertyPoint3DPtr coordinates;
21     /// Поле "Тип основания"
22     IPropertyListPtr surfaceType;
23     /// Поле "Тип размера"
24     IPropertyListPtr sizeType;
25     /// Поле "Величина размера"
26     IPropertyEditPtr size;
27     /// Поле "Высота"
28     IPropertyEditPtr height;
29     /// Поле "Угол наклона  $\alpha$ "
30     IPropertyEditPtr angleAlpha;
31     /// Поле "Радиус отверстия"
32     IPropertyEditPtr roundingRadius;
33     /// Поле "Наличие отверстия"
34     IPropertyTwinSwitcherPtr hasHole;
35     /// Поле "Радиус скругления"
36     IPropertyEditPtr holeRadius;
37     /// Создание кнопки "Применить"
38     IPropertyTextButtonPtr button;
39
40     MenuBase() { };
41     ~MenuBase() { }
42
43     /// <summary>
44     /// Получение модели объекта
45     /// </summary>
46     /// <returns>Модель объекта</returns>
47     virtual SMPBase* getModel() = NULL;
48     /// <summary>
49     /// Присвоение имени полю
50     /// </summary>
51     /// <param name="field">Передаваемое поле</param>
52     /// <param name="name">Имя поля</param>
53     void setFieldName(IPropertyEditPtr field, BSTR name);
54     /// <summary>
```

```
55     /// Подключение переданного поля
56     /// </summary>
57     /// <param name="field">Передаваемое поле</param>
58     void setFieldActive(IPropertyEditPtr field);
59     /// <summary>
60     /// Отключение переданного поля
61     /// </summary>
62     /// <param name="field">Передаваемое поле</param>
63     void setFieldInactive(IPropertyEditPtr field);
64     /// <summary>
65     /// Проверка условия подключения поля "Наличия отверстия"
66     /// </summary>
67     void checkHasHole();
68     void CreateMultiButton(ButtonTypeEnum type);
69
70 protected:
71     /// <summary>
72     /// Создание поля "Координаты"
73     /// </summary>
74     /// <param name="id">Номер элемента координат</param>
75     /// <param name="id_x">Номер элемента координаты X</param>
76     /// <param name="id_y">Номер элемента координаты Y</param>
77     /// <param name="id_z">Номер элемента координаты Z</param>
78     void initCoordinates(size_t id, size_t id_x, size_t id_y, size_t id_z);
79     /// <summary>
80     /// Создание поля "Тип основания"
81     /// </summary>
82     /// <param name="id">Номер элемента</param>
83     void initSurfaceType(size_t id);
84     /// <summary>
85     /// Создание поля "Тип размера"
86     /// </summary>
87     /// <param name="id">Номер элемента</param>
88     void initSizeType(size_t id);
89     /// <summary>
90     /// Создание поля "Величина размера"
91     /// </summary>
92     /// <param name="id">Номер элемента</param>
93     void initSize(size_t id);
94     /// <summary>
95     /// Создание поля "Высота"
96     /// </summary>
97     /// <param name="id">Номер элемента</param>
98     void initHeight(size_t id);
99     /// <summary>
100    /// Создание поля "Наличие отверстия"
101    /// </summary>
102    /// <param name="id">Номер элемента</param>
103    void initAngleAlpha(size_t id);
104    /// <summary>
105    /// Создание поля "Радиус отверстия"
106    /// </summary>
```

```
107     ///Создание поля "Угол наклона  $\alpha$ "
111     </summary>
112     ///Создание поля "Радиус скругления"
116     </summary>
117     ///Создание кнопки "Применить"
121     </summary>
122     ///
```