

```
1 #include "StdAfx.h"
2 #include "MenuBase.h"
3 #include "SMPConditions.h"
4
5 #ifndef _MENU_CONDITIONS_H
6 #define _MENU_CONDITIONS_H
7
8 #if _MSC_VER > 1000
9 #pragma once
10 #endif
11 #endif
12
13 /// Класс меню параметров модели по режимам резания
14 class MenuConditions : public MenuBase
15 {
16 public:
17     /// Модель объекта
18     SMPConditions* model;
19     /// Поле "Подача S"
20     IPropertyEditPtr parameterS;
21     /// Поле "Глубина t"
22     IPropertyEditPtr parameterT;
23     /// Поле "Угол λд"
24     IPropertyEditPtr angleLambdaD;
25     /// Поле "Угол η – автоматически или вручную"
26     IPropertyTwinSwitcherPtr angleEtaAuto;
27     /// Поле "Угол η"
28     IPropertyEditPtr angleEta;
29     /// Поле "Радиус режущей кромки"
30     IPropertyEditPtr cutRadius;
31
32     MenuConditions() { };
33     ~MenuConditions() { }
34
35     /// Получение модели объекта
36     SMPBase* getModel() override { return this->model; }
37     /// <summary>
38     /// Создание полей меню
39     /// </summary>
40     /// <param name="collection">Коллекция контролов</param>
41     void init(IPropertyControlsPtr collection);
42     /// Обновление параметров меню
43     bool updateMenuParameters();
44     /// Проверка условия подключения поля "Угол η"
45     void checkAngleEtaAuto();
46     /// Проверка условия подсчёта значения поля "Угол η"
47     bool findAngleEta();
48
49     /// Список номеров полей
50     enum Elements
51     {
52         e_coordinates = 200000, /// Координаты центра основания
53         e_X,                    /// Координата X
54         e_Y,                    /// Координата Y
55         e_Z,                    /// Координата Z
```

```
56         e_surfaceType,          ////// Тип основания
57         e_sizeType,             ////// Тип размера
58         e_size,                 ////// Величина размера
59         e_height,               ////// Высота
60         e_angleAlpha,           ////// Угол наклона  $\alpha$ 
61         e_roundingRadius,       ////// Радиус скругления
62         e_hasHole,              ////// Наличие отверстия
63         e_holeRadius,           ////// Радиус отверстия
64         e_parameterS,           ////// Подача S
65         e_parameterT,           ////// Глубина t
66         e_angleLambdaD,         ////// Угол  $\lambda_d$ 
67         e_angleEtaAuto,         ////// Угол  $\eta$  – автоматически или вручную
68         e_angleEta,             ////// Угол  $\eta$ 
69         e_cutRadius,            ////// Радиус режущей кромки
70         e_button                ////// Кнопка "Применить"
71     };
72
73     protected:
74         ////// <summary>
75         ////// Создание поля "Подача S"
76         ////// </summary>
77         ////// <param name="id">Номер элемента</param>
78         void initParameterS(size_t id);
79         ////// <summary>
80         ////// Создание поля "Глубина t"
81         ////// </summary>
82         ////// <param name="id">Номер элемента</param>
83         void initParameterT(size_t id);
84         ////// <summary>
85         ////// Создание поля "Угол  $\lambda_d$ "
86         ////// </summary>
87         ////// <param name="id">Номер элемента</param>
88         void initAngleLambdaD(size_t id);
89         ////// <summary>
90         ////// Создание поля "Угол  $\eta$  – автоматически или вручную"
91         ////// </summary>
92         ////// <param name="id">Номер элемента</param>
93         void initAngleEtaAuto(size_t id);
94         ////// <summary>
95         ////// Создание поля "Угол  $\eta$ "
96         ////// </summary>
97         ////// <param name="id">Номер элемента</param>
98         void initAngleEta(size_t id);
99         ////// <summary>
100        ////// Создание поля "Радиус режущей кромки"
101        ////// </summary>
102        ////// <param name="id">Номер элемента</param>
103        void initCutRadius(size_t id);
104    };
105
```