

```
1  #ifndef _SMP_PROFILE_H
2  #define _SMP_PROFILE_H
3
4  #include "SMPBase.h"
5  #include <PArray.h>
6
7  #if _MSC_VER > 1000
8  #pragma once
9  #endif
10 #endif
11
12 /// Класс СМП по профилю режущей кромки
13 class SMPPProfile : public SMPBase
14 {
15 public:
16     /// Длина режущей кромки
17     double cutLength = 2;
18     /// Угол  $\gamma$  режущей кромки
19     double cutAngleGamma = 60;
20     /// Радиус режущей кромки
21     double cutRadius = 2;
22     /// Тип режущей кромки (0 - фаска, 1 - скругление)
23     bool cutRoundingType = true;
24     /// Радиус скругления режущей кромки
25     double cutFilletRadius = 0.1;
26     /// Длина фаски режущей кромки
27     double cutChamferLength = 0.3;
28     /// Угол фаски режущей кромки
29     double cutChamferAngle = 45;
30     /// Обновление параметров меню
31     bool updateParameters();
32     /// Создание модели
33     void createModel();
34     /// Получение итоговой длины режущей кромки
35     double getCutLength();
36
37 protected:
38     /// Номер объекта
39     static size_t objectId;
40     /// Добавочная длина режущей кромки
41     double cutLength_delta = 0;
42     /// Первый катет фаски режущей кромки
43     double cutChamferLength_cat1 = 0;
44     /// Второй катет фаски режущей кромки
45     double cutChamferLength_cat2 = 0;
46     /// Обновление параметров меню для основания круга
47     void updateParametersCircle(double* radius_delta);
48     /// Обновление параметров меню для основания правильного
        многоугольника
49     void updateParametersNGon(double* radius_delta);
50     /// Обновление параметров меню для основания тригона
51     void updateParametersTrigon(double* radius_delta);
52     /// Обновление параметров меню для основания ромба
53     void updateParametersRhombus(double* radius_delta);
```

```
54     /// Обновление параметров меню для режущей кромки
55     void updateParametersCut(double* radius_delta);
56     /// Добавление режущей кромки
57     void addCutEvolution();
58     /// Нахождение граней для режущей кромки
59     IFacePtr findEvolutionFaces();
60     /// Нахождение рёбер для режущей кромки
61     void findEvolutionEdges(PArray<IModelObjectPtr>* edges_array,      ↗
        IFacePtr face_base);
62     /// Создание плоскости для режущей кромки
63     ILocalCoordinateSystemPtr createEvolutionCoordinateSystem();
64     /// Создание эскиза для режущей кромки
65     bool createEvolutionSketch(ISketchPtr cut_sketch);
66     /// Создание скругления для режущей кромки
67     void addEvolutionEdgesRounding(PArray<IModelObjectPtr>*          ↗
        fillet_objects);
68     /// Нахождение рёбер для скругления на режущей кромке
69     void findEvolutionRoundingEdges(PArray<IModelObjectPtr>*        ↗
        object_array);
70     /// Проверка грани для режущей кромки
71     bool checkEvolutionRoundingFace(IFacePtr face);
72 };
73
```