```cpp
#include "stdafx.h"
#include "MenuProfile.h"


void MenuProfile::init(IPropertyControlsPtr collection) {
    this->model = new SMPProfile;
    this->collection = collection;

    this->initCoordinates(Elements::e_coordinates,
        Elements::e_X, Elements::e_Y, Elements::e_Z);
    this->initSurfaceType(Elements::e_surfaceType);
    this->initSizeType(Elements::e_sizeType);
    this->initSize(Elements::e_size);
    this->initHeight(Elements::e_height);
    this->initHasHole(Elements::e_hasHole);
    this->initHoleRadius(Elements::e_holeRadius);
    this->initAngleAlpha(Elements::e_angleAlpha);
    this->initRoundingRadius(Elements::e_roundingRadius);
    this->initEvolutionLength(Elements::e_cutLength);
    this->initEvolutionAngleGamma(Elements::e_cutAngleGamma);
    this->initEvolutionRadius(Elements::e_cutRadius);
    this->initEvolutionRoundingType(Elements::e_cutRoundingType);
    this->initEvolutionFilletRadius(Elements::e_cutFilletRadius);
    this->initEvolutionChamferLength(Elements::e_cutChamferLength);
    this->initEvolutionChamferAngle(Elements::e_cutChamferAngle);
    this->initButton(Elements::e_button);

    this->checkHasHole();
    this->checkEvolutionRoundingType();
}

bool MenuProfile::updateMenuParameters() {
    this->model->coordinates.X = this->coordinates->GetCoordinate(0)->Value;
    this->model->coordinates.Y = this->coordinates->GetCoordinate(1)->Value;
    this->model->coordinates.Z = this->coordinates->GetCoordinate(2)->Value;
    this->model->surfaceType = (enum SurfaceType)this->surfaceType->Find(this->surfaceType->Value);
    this->model->sizeType = (enum SizeType)this->sizeType->Find(this->sizeType->Value);
    this->model->size = this->size->Value;
    this->model->height = this->height->Value;
    this->model->angleAlpha = this->angleAlpha->Value;
    this->model->roundingRadius = this->roundingRadius->Value;
    this->model->hasHole = (this->hasHole->Value.intVal == 1);
    this->model->holeRadius = this->holeRadius->Value;
    this->model->cutLength = this->cutLength->Value;
    this->model->cutAngleGamma = this->cutAngleGamma->Value;
    this->model->cutRadius = this->cutRadius->Value;
    this->model->cutRoundingType = (this->cutRoundingType->Value.intVal == 1);
    this->model->cutFilletRadius = this->cutFilletRadius->Value;
```

```
49      this->model->cutChamferLength = this->cutChamferLength->Value;
50      this->model->cutChamferAngle = this->cutChamferAngle->Value;
51
52      return this->model->updateParameters();
53  }
54
55  void MenuProfile::initEvolutionLength(size_t id) {
56      BSTR name = _T("Длина режущей кромки");
57      this->cutLength = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
58      this->cutLength->Name = name;
59      this->cutLength->Hint = name;
60      this->cutLength->Tips = name;
61      this->cutLength->Id = id;
62      this->cutLength->Value = this->model->cutLength;
63  }
64
65  void MenuProfile::initEvolutionAngleGamma(size_t id) {
66      BSTR name = _T("Угол режущей кромки γ");
67      this->cutAngleGamma = this->collection->Add
          (ControlTypeEnum::ksControlEditAngle);
68      this->cutAngleGamma->Name = name;
69      this->cutAngleGamma->Hint = name;
70      this->cutAngleGamma->Tips = name;
71      this->cutAngleGamma->Id = id;
72      this->cutAngleGamma->Value = this->model->cutAngleGamma;
73  }
74
75  void MenuProfile::initEvolutionRadius(size_t id) {
76      BSTR name = _T("Радиус режущей кромки");
77      this->cutRadius = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
78      this->cutRadius->Name = name;
79      this->cutRadius->Hint = name;
80      this->cutRadius->Tips = name;
81      this->cutRadius->Id = id;
82      this->cutRadius->Value = this->model->cutRadius;
83  }
84
85  void MenuProfile::initEvolutionRoundingType(size_t id) {
86      BSTR name = _T("Скругление режущей кромки");
87      this->cutRoundingType = this->collection->Add
          (ControlTypeEnum::ksControlTwinSwitcher);
88      this->cutRoundingType->Name = name;
89      this->cutRoundingType->Hint = name;
90      this->cutRoundingType->Tips = name;
91      this->cutRoundingType->Label2 = _T("Фаска режущей кромки");
92      this->cutRoundingType->Id = id;
93      this->cutRoundingType->Value = (this->model->cutRoundingType ? 1 :
          2);
94  }
95
96  void MenuProfile::initEvolutionFilletRadius(size_t id) {
97      BSTR name = _T("Радиус скругления режущей кромки");
98      this->cutFilletRadius = this->collection->Add
```

```
        (ControlTypeEnum::ksControlEditLength);
 99     this->cutFilletRadius->Name = name;
100     this->cutFilletRadius->Hint = name;
101     this->cutFilletRadius->Tips = name;
102     this->cutFilletRadius->Id = id;
103     this->cutFilletRadius->Value = this->model->cutFilletRadius;
104 }
105
106 void MenuProfile::initEvolutionChamferLength(size_t id) {
107     BSTR name = _T("Фаска режущей кромки");
108     this->cutChamferLength = this->collection->Add
        (ControlTypeEnum::ksControlEditLength);
109     this->cutChamferLength->Name = name;
110     this->cutChamferLength->Hint = name;
111     this->cutChamferLength->Tips = name;
112     this->cutChamferLength->Id = id;
113     this->cutChamferLength->Value = this->model->cutChamferLength;
114 }
115
116 void MenuProfile::initEvolutionChamferAngle(size_t id) {
117     BSTR name = _T("Угол фаски режущей кромки");
118     this->cutChamferAngle = this->collection->Add
        (ControlTypeEnum::ksControlEditAngle);
119     this->cutChamferAngle->Name = name;
120     this->cutChamferAngle->Hint = name;
121     this->cutChamferAngle->Tips = name;
122     this->cutChamferAngle->Id = id;
123     this->cutChamferAngle->Value = this->model->cutChamferAngle;
124 }
125
126 void MenuProfile::checkEvolutionRoundingType() {
127     if (this->cutRoundingType->Value.intVal == 1) {
128         this->setFieldActive(field: this->cutFilletRadius);
129         this->setFieldInactive(field: this->cutChamferLength);
130         this->setFieldInactive(field: this->cutChamferAngle);
131     }
132     else {
133         this->setFieldInactive(field: this->cutFilletRadius);
134         this->setFieldActive(field: this->cutChamferLength);
135         this->setFieldActive(field: this->cutChamferAngle);
136     }
137 }
```