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


void MenuConditions::init(IPropertyControlsPtr collection) {
    this->model = new SMPConditions;
    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->initParameterS(Elements::e_parameterS);
    this->initParameterT(Elements::e_parameterT);
    this->initAngleLambdaD(Elements::e_angleLambdaD);
    this->initAngleEtaAuto(Elements::e_angleEtaAuto);
    this->initAngleEta(Elements::e_angleEta);
    this->initCutRadius(Elements::e_cutRadius);
    this->initButton(Elements::e_button);

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

bool MenuConditions::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->parameterS = this->parameterS->Value;
    this->model->parameterT = this->parameterT->Value;
    this->model->angleLambdaD = this->angleLambdaD->Value;
    this->model->angleEtaAuto = (this->angleEtaAuto->Value.intVal == 1);
    this->model->angleEta = this->angleEta->Value;
    this->model->cutRadius = this->cutRadius->Value;
```

```
49
50      return this->model->updateParameters();
51  }
52
53  void MenuConditions::initParameterS(size_t id) {
54      this->parameterS = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
55      this->setFieldName(this->parameterS, _T("Подача S"));
56      this->parameterS->Id = id;
57      this->parameterS->Value = this->model->parameterS;
58  }
59
60  void MenuConditions::initParameterT(size_t id) {
61      this->parameterT = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
62      this->setFieldName(this->parameterT, _T("Глубина t"));
63      this->parameterT->Id = id;
64      this->parameterT->Value = this->model->parameterT;
65  }
66
67  void MenuConditions::initAngleLambdaD(size_t id) {
68      this->angleLambdaD = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
69      this->setFieldName(this->angleLambdaD, _T("Угол λд"));
70      this->angleLambdaD->Id = id;
71      this->angleLambdaD->Value = this->model->angleLambdaD;
72  }
73
74  void MenuConditions::initAngleEtaAuto(size_t id) {
75      this->angleEtaAuto = this->collection->Add
          (ControlTypeEnum::ksControlTwinSwitcher);
76      this->setFieldName(this->angleEtaAuto, _T("Угол η.
          Автоматически"));
77      this->angleEtaAuto->Label2 = _T("Вручную");
78      this->angleEtaAuto->Id = id;
79      this->angleEtaAuto->Value = (this->model->angleEtaAuto ? 1 : 2);
80  }
81
82  void MenuConditions::initAngleEta(size_t id) {
83      this->angleEta = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
84      this->setFieldName(this->angleLambdaD, _T("Угол η"));
85      this->angleEta->Id = id;
86      this->angleEta->Value = this->model->angleEta;
87  }
88
89  void MenuConditions::initCutRadius(size_t id) {
90      this->cutRadius = this->collection->Add
          (ControlTypeEnum::ksControlEditLength);
91      this->setFieldName(this->cutRadius, _T("Радиус режущей кромки"));
92      this->cutRadius->Id = id;
93      this->cutRadius->Value = this->model->cutRadius;
94  }
95
96  void MenuConditions::checkAngleEtaAuto() {
97      if (this->angleEtaAuto->Value.intVal == 1) {
```

```
 98          this->setFieldInactive(field: this->angleEta);
 99      }
100      else {
101          this->setFieldActive(field: this->angleEta);
102      }
103  }
104
105  bool MenuConditions::findAngleEta() {
106      bool check = false;
107      if (this->angleEtaAuto->Value.intVal == 1) {
108          this->angleEta->Value = this->model->angleEta;
109          check = true;
110      }
111      return check;
112  }
```