Министерство образования Республики Беларусь

Белорусский национальный технический университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем и технологий»

**Отчёт**

по лабораторной работе № 5

по дисциплине ***«Методы и алгоритмы компьютерной графики»***

Вариант 4

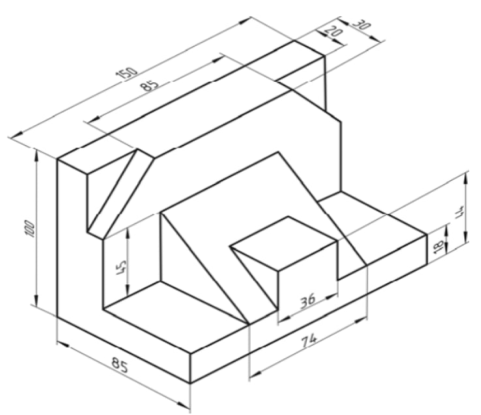
|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | ст.гр. Кишкурно М.В. |
| Преподаватель: |  | ст.пр. Тетерюкова И.О. |

2023

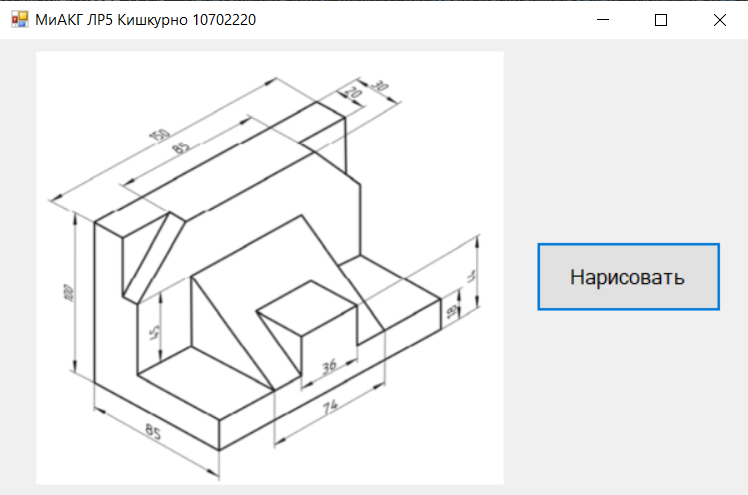
**Цель работы:** Приобретение приемов по построению объёмных моделей с помощью SolidWorks API.

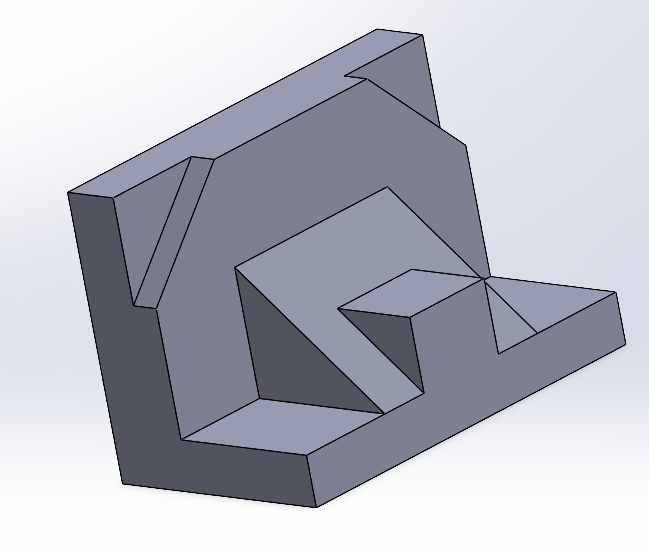
**Задание:**

Построить ЗD-модель из ЛР №3 с помощью SolidWorks API.

****

**Скриншот результата:**

****

****

**Листинг**

using SolidWorks.Interop.sldworks;

using SolidWorks.Interop.swconst;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SolidLab5

{

public partial class Form1 : Form

{

SldWorks app;

IModelDoc2 doc;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

double k = 1000.0;

double[] l = new double[] { 18 / k, 44 / k, 36 / k, 74 / k, 85 / k, 100 / k, 45 / k, 150 / k, 85 / k, 20 / k, 30 / k };

app = new SldWorks();

app.FrameState = (int)swWindowState\_e.swWindowMaximized;

app.Visible = true;

doc = (ModelDoc2)app.INewPart();

doc.SetUnits((short)swLengthUnit\_e.swMM, (short)swFractionDisplay\_e.swDECIMAL, 0, 0, false);

doc = app.IActiveDoc2;

var sm = doc.SketchManager;

var fm = doc.FeatureManager;

string front = "Front Plane";

string top = "Top Plane";

double size\_x = 0.01;

double size\_y = 0.01;

double size\_z = 0.01;

int pref\_toggle = (int)swUserPreferenceToggle\_e.swInputDimValOnCreate;

app.SetUserPreferenceToggle(pref\_toggle, false);

selectPlane(top);

sm.InsertSketch(false);

var rect = sm.CreateCenterRectangle(0, 0, 0, l[4] / 2, l[7] / 2, 0);

//очистить буфер выбранных элементов

doc.ClearSelection();

//выбрать линию с прямоугольника

var el = rect[0] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0, -size\_y, 0.2 + l[4] / 2 + size\_y);

el.DeSelect();

el = rect[1] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0 + l[7] / 2 + size\_x, -size\_y, 0);

el.DeSelect();

//вытянуть бобышку

var feature = featureExtrusion(l[0]);

doc.ClearSelection();

//получить массив грани бобышки

var faces = feature.GetFaces();

//выбрать низ бобышки

var ent = faces[0] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

rect = sm.CreateCenterRectangle(-l[4] / 2 + l[10] / 2, 0, 0, -l[4] / 2 + l[10], l[7] / 2, 0);

el = rect[0] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0, -size\_y, 0.2 + l[4] / 2 + size\_y);

el.DeSelect();

el = rect[1] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0 + l[7] / 2 + size\_x, -size\_y, 0);

el.DeSelect();

var feature1 = featureExtrusion(l[5], true);

doc.ClearSelection();

//получить массив грани бобышки

faces = feature.GetFaces();

//выбрать вторую (вверх бобышки)

ent = faces[1] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

rect = sm.CreateCenterRectangle((l[10] / 2), 0, 0, l[4] / 2, l[3] / 2, 0);

el = rect[0] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0, -size\_y, 0.2 + l[4] / 2 + size\_y);

el.DeSelect();

el = rect[1] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0 + l[7] / 2 + size\_x, -size\_y, 0);

el.DeSelect();

//вытянуть бобышку

var feature2 = featureExtrusion(l[6]);

doc.ClearSelection();

//получить массив грани бобышки

faces = feature2.GetFaces();

//выбрать вторую (вверх бобышки)

ent = faces[1] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

sm.CreateLine(-l[4] / 2 + l[10], l[0] + l[6], 0, l[4] / 2, l[0], 0);

doc.ClearSelection();

sm.CreateLine(l[4] / 2, l[0] + l[6], 0, l[4] / 2, l[0], 0);

doc.ClearSelection();

sm.CreateLine(-l[4] / 2 + l[10], l[0] + l[6], 0, l[4] / 2, l[0] + l[6], 0);

doc.ClearSelection();

featureCut(l[3], false, swEndConditions\_e.swEndCondThroughAll);

doc.ClearSelection();

//получить массив грани бобышки

faces = feature.GetFaces();

//выбрать вторую (вверх бобышки)

ent = faces[0] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

//получить массив грани бобышки

faces = feature.GetFaces();

//выбрать низ бобышки

ent = faces[2] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

rect = sm.CreateCenterRectangle(l[10] / 2, 0, 0, l[4] / 2, l[2] / 2, 0);

el = rect[0] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0, -size\_y, 0.2 + l[4] / 2 + size\_y);

el.DeSelect();

el = rect[1] as SketchSegment;

el.Select(true);

doc.IAddDimension2(0 + l[7] / 2 + size\_x, -size\_y, 0);

el.DeSelect();

//вытянуть бобышку

var feature4 = featureExtrusion(l[1],true);

doc.ClearSelection();

//получить массив грани бобышки

faces = feature1.GetFaces();

//выбрать вторую (вверх бобышки)

ent = faces[0] as Entity;

//выбрать верхнюю грань

ent.Select(true);

//добавить на нее эскиз

sm.InsertSketch(false);

sm.CreateLine(l[3]/2 , l[5], 0, l[7] / 2 , l[5], 0);

doc.ClearSelection();

sm.CreateLine(l[7] / 2, l[0] + l[6], 0, l[7] / 2, l[5], 0);

doc.ClearSelection();

sm.CreateLine(l[3] / 2, l[5], 0, l[7] / 2, l[0] + l[6], 0);

doc.ClearSelection();

sm.CreateLine(-l[3] / 2, l[5], 0, -l[7] / 2, l[5], 0);

doc.ClearSelection();

sm.CreateLine(-l[7] / 2, l[0] + l[6], 0,- l[7] / 2, l[5], 0);

doc.ClearSelection();

sm.CreateLine(-l[3] / 2, l[5], 0, -l[7] / 2, l[0] + l[6], 0);

doc.ClearSelection();

featureCut(l[10] - l[9], false, swEndConditions\_e.swEndCondBlind);

doc.ClearSelection();

}

//выбрать плоскость

private void selectPlane(string name, string obj = "PLANE")

{

doc.Extension.SelectByID2(name, obj, 0, 0, 0, false, 0, null, 0);

}

private Feature featureExtrusion(double deepth, bool dir = false)

{

return doc.FeatureManager.FeatureExtrusion2(true, false, dir,

(int)swEndConditions\_e.swEndCondBlind, (int)swEndConditions\_e.swEndCondBlind,

deepth, 0, false, false, false, false, 0, 0, false, false, false, false, true,

true, true, 0, 0, false);

}

private Feature featureCut(double deepth, bool flip = false, swEndConditions\_e mode = swEndConditions\_e.swEndCondBlind)

{

return doc.FeatureManager.FeatureCut2(true, flip, false, (int)mode, (int)mode,

deepth, 0, false, false, false, false, 0, 0, false, false, false, false, false,

false, false, false, false, false);

}

private Feature featureExtrusionFromSerface(double deepth, bool dir = false)

{

return doc.FeatureManager.FeatureExtrusion2(true, false, dir,

(int)swEndConditions\_e.swEndCondOffsetFromSurface, (int)swEndConditions\_e.swEndCondOffsetFromSurface,

deepth, 0, false, false, false, false, 0, 0, false, false, false, false, true,

true, true, 0, 0, false);

}

}

}