Министерство образования Республики Беларусь

Белорусский национальный технический университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем и технологий»

**Отчёт**

по лабораторной работе № 4

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

Вариант 4

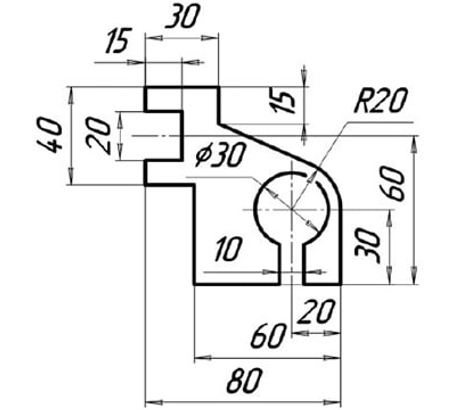
|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | ст.гр. Кишкурно М.В. |
| Преподаватель: |  | ст.пр. Тетерюкова И.О. |

2023

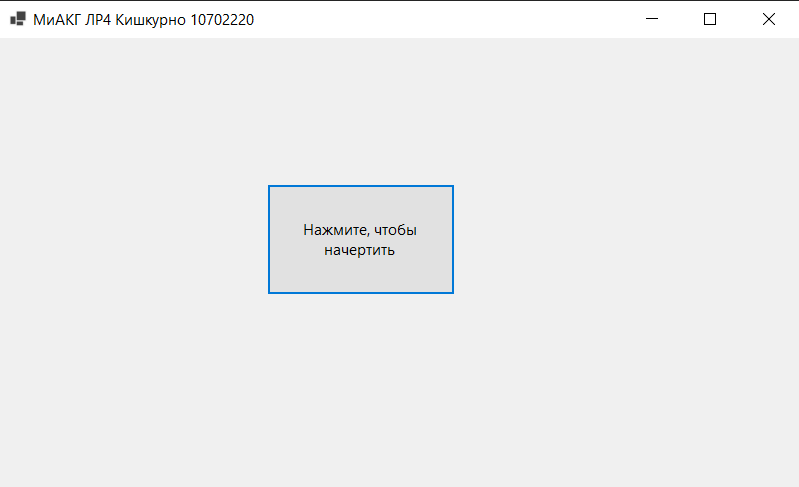
**Цель работы:** Приобретение приемов по построению чертежа с помощью SolidWorks API.

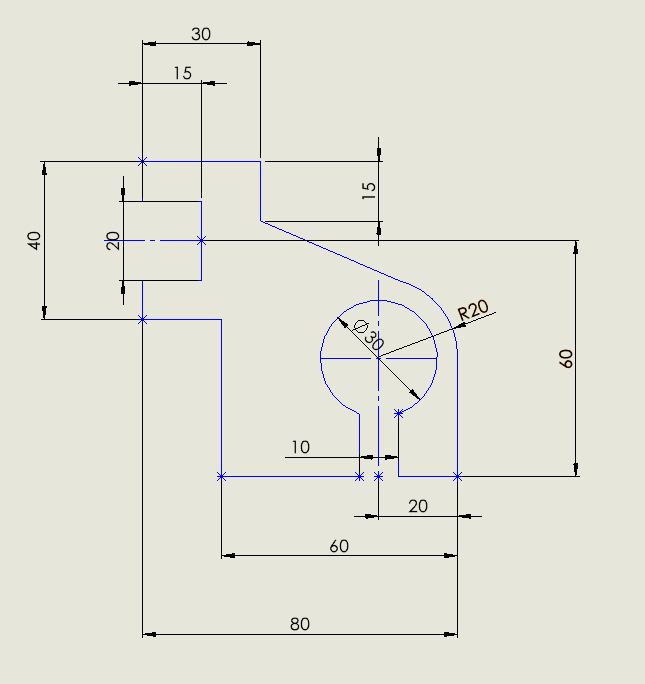
**Задание:**

Построить чертёж из ЛР №1 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.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace MIAKGlab4

{

public partial class Form1 : Form

{

private SldWorks sld;

private ModelDoc2 modelDoc;

private SketchManager sketchManager;

public Form1()

{

InitializeComponent();

}

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

{

double L0 = 0.03;

double L1 = 0.015;

double L2 = 0.02;

double L3 = 0.04;

double L4 = 0.08;

double L5 = 0.06;

double L6 = 0.02;

double L7 = 0.01;

double L8 = 0.03;

double L9 = 0.02;

double L12 = 0.016;

double L10 = 0.06;

double L11 = 0.015;

try

{

//попытка открыть либо получить открытый через SolidWorks Api

sld = new SldWorks();

}

catch

{

//если и это не помогло, тогда что-то не так

MessageBox.Show("Не удалось открыть SolidWorks либо найти открытое приложение");

return;

}

//задаем размер окна - на весь экран

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

//делаем окно видимым

sld.Visible = true;

//создать и открыть

modelDoc = (ModelDoc2)sld.INewDrawing((int)swDwgTemplates\_e.swDwgTemplateCustom);

//задать размеры в миллиметрах

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

//получаем открытый документ

modelDoc = (ModelDoc2)sld.ActiveDoc;

//получаем ссылку на интерфейс, ответственный за рисование

sketchManager = (SketchManager)modelDoc.SketchManager;

//выбираем какое всплывающее окно отключить

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

//отключаем всплывающее окно нанесения размеров

sld.SetUserPreferenceToggle(pref\_toggle, false);

double dx = 0.1;

double dy = 0.15;

double length = 0;

double tempLength = 0;

sketchManager.CreateLine(L0 + dx, - L11 + dy, 0, L0 + dx, 0 + dy, 0);

modelDoc.IAddVerticalDimension2(0.06 + dx, dy - L11/2, 0);

modelDoc.ClearSelection();

//рисуем линию

sketchManager.CreateLine(L0 + dx, 0 + dy, 0, 0 + dx, 0 + dy, 0);

//добавляем размер для линии

modelDoc.IAddHorizontalDimension2(L0 / 2 + dx, dy + 0.03, 0);

//после построения элемент выбран и выделение лучше снимать

//снимаем выделение со всех

modelDoc.ClearSelection();

length = (L3 - L2)/ 2;

sketchManager.CreateLine(0 + dx, 0 + dy, 0, 0 + dx, dy - length , 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(0 + dx, dy - length, 0, L1 + dx, dy - length, 0);

modelDoc.IAddHorizontalDimension2(L2 / 2 + dx, dy + 0.02, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L1 + dx, dy - length, 0, L1 + dx, dy - length - L2, 0);

modelDoc.IAddVerticalDimension2(-0.005+ dx, dy - L2 , 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L1 + dx, dy - length - L2, 0, 0 + dx, dy - length - L2, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(0 + dx, dy - length - L2, 0, 0 + dx, dy - 2\*length - L2, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(0 + dx, dy - 2 \* length - L2, 0, L4-L5 + dx, dy - 2 \* length - L2, 0);

modelDoc.ClearSelection();

tempLength = -L10 + L3 / 2 - 2 \* length - L2;

sketchManager.CreateLine(L4 - L5 + dx, dy - 2 \* length - L2, 0, L4 - L5 + dx, tempLength + dy, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L5 + dx, tempLength + dy, 0, L4 - L6 - L7 / 2 + dx, tempLength + dy, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L6 - L7 / 2 + dx, tempLength + dy, 0, L4 - L6 - L7 / 2 + dx, tempLength + L12 + dy, 0);

modelDoc.ClearSelection();

//строим дугу

sketchManager.CreateArc(L4 - L6 - L7 / 2 + L7/2 + dx, tempLength + L8 + dy, 0, L4 - L6 - L7 / 2 + dx, tempLength + L12 + dy, 0, L4 - L6 - L7 / 2 + L7+ dx, tempLength + L12 + dy, 0, -1);

//добавляем диаметральный размер

modelDoc.IAddDiameterDimension2 (dx + 0.055, dy - 0.045, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L6 - L7 / 2 + L7 + dx, tempLength + L12 + dy, 0, L4 - L6 - L7 / 2 + L7 + dx, tempLength + dy, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L6 - L7 / 2 + L7 + dx, tempLength + dy, 0, L4 - L6 + L6 + dx, tempLength + dy, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L6 + L6 + dx, tempLength + dy, 0, L4 - L6 + L6 + dx, tempLength + L10/2 + 0.00055+ dy, 0);

modelDoc.ClearSelection();

//строим дугу

sketchManager.CreateArc(L4 - L6 - L7 / 2 + L7 / 2 + dx, tempLength + L8 + dy, 0, L4 + dx, tempLength + L10 / 2 + 0.00055 + dy, 0, L4 - L6 + L8 / 2 - L7 + dx, dy - length + 0.0002 - L2, 0, 1);

//добавляем радиальный размер

modelDoc.IAddRadialDimension2(dx + 0.085, dy - 0.04, 0);

modelDoc.ClearSelection();

sketchManager.CreateLine(L4 - L6 + L6 - L6 + L8 / 2 - L7 + dx, dy - length - L2 +0.0002, 0, L0 + dx, -L11 + dy, 0);

modelDoc.ClearSelection();

//осевая на верхнюю

sketchManager.CreateCenterLine(L1 + dx, dy - L3 / 2, 0, L1 - 0.025+ dx, dy - L3 / 2, 0);

modelDoc.ClearSelection();

//осевые на окружность

sketchManager.CreateCenterLine(L4 - L6 + dx, tempLength -0.01+ dy, 0, L4 - L6 + dx, tempLength + 0.05 + dy, 0);

modelDoc.ClearSelection();

sketchManager.CreateCenterLine(L4 - L6 - L8/2 - 0.001+ dx, tempLength + L8 + dy, 0, L4 - L6 + L8/2 +0.001+ dx, tempLength + L8 + dy, 0);

modelDoc.ClearSelection();

//размеры по точкам

SketchPoint firstPoint;

SketchPoint secondPoint;

firstPoint = sketchManager.CreatePoint(0 + dx, 0 + dy,0);

secondPoint = sketchManager.CreatePoint(0 + dx, dy - 2 \* length - L2, 0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddVerticalDimension2(dx - 0.025, dy - L3/2, 0);

modelDoc.ClearSelection();

firstPoint = sketchManager.CreatePoint(0 + dx, dy - 2 \* length - L2, 0);

secondPoint = sketchManager.CreatePoint(L4 - L6 + L6 + dx, tempLength + dy,0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddHorizontalDimension2(dx +L4/2, dy - 0.12, 0);

modelDoc.ClearSelection();

firstPoint = sketchManager.CreatePoint(L4 - L5 + dx, tempLength + dy, 0);

secondPoint = sketchManager.CreatePoint(L4 + dx, tempLength + dy, 0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddHorizontalDimension2(dx + L4/2+0.01, dy - 0.1, 0);

modelDoc.ClearSelection();

firstPoint = sketchManager.CreatePoint(L4 - L6 + dx, tempLength + dy, 0);

secondPoint = sketchManager.CreatePoint(L4 + dx, tempLength + dy, 0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddHorizontalDimension2(dx +L4/2+L5/2 , dy - 0.09, 0);

modelDoc.ClearSelection();

firstPoint = sketchManager.CreatePoint(L4 - L6 - L7 / 2 + dx, tempLength + dy, 0);

secondPoint = sketchManager.CreatePoint(L4 - L6 - L7 / 2 + L7 + dx, tempLength + L12 + dy, 0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddHorizontalDimension2(dx + L4/2, dy - 0.075, 0);

modelDoc.ClearSelection();

firstPoint = sketchManager.CreatePoint(L1 + dx, dy - L3/2, 0);

secondPoint = sketchManager.CreatePoint(L4 + dx, tempLength + dy, 0);

firstPoint.Select(true);

secondPoint.Select(true);

modelDoc.IAddVerticalDimension2(dx + 0.11, dy - 0.05, 0);

modelDoc.ClearSelection();

}

}

}