МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Геометрическое моделирование»

**Пояснительная записка**

**к курсовой работе по теме**

**Построение аксонометрических проекций детали на языках высокого уровня**

Вариант 3

Выполнил:

студент группы ИВТАПбд-31

Вершинин Д. В.

Проверил:

Войт Н. Н.

Ульяновск, 2018

Оглавление

[Введение 3](#_Toc532329543)

[1. Техническое задание 4](#_Toc532329544)

[2. Теоретическая часть 5](#_Toc532329545)

[3. Проектирование 8](#_Toc532329546)

[4. Программная реализация 11](#_Toc532329547)

[Заключение 12](#_Toc532329548)

[Используемые ресурсы 13](#_Toc532329549)

[Приложение 14](#_Toc532329550)

# Введение

Геометрическое моделирование – раздел математического моделирования – позволяет решать разнообразные задачи в двумерном, трехмерном и, в общем случае, в многомерном пространстве. Геометрическая модель включает в себя системы уравнений и алгоритмы их реализации. Математической основой построения модели являются уравнения, описывающие форму и движение объектов. Все многообразие геометрических объектов является комбинацией различных примитивов – простейших фигур, которые в свою очередь состоят из графических элементов - точек, линий и поверхностей.

В настоящее время геометрическое моделирование успешно используется в управлении и других областях человеческой деятельности. Можно выделить две основные области применения геометрического моделирования: проектирование и научные исследования.

Геометрическое моделирование может использоваться при анализе числовых данных. В таких случаях исходным числовым данным ставится в соответствие некоторая геометрическая интерпретация, которая затем анализируется, а результаты анализа истолковываются в понятиях исходных данных.

# 1. Техническое задание

В ходе выполнения курсовой работы необходимо разработать программу на языке высокого уровня (С++, С#, Java и т. д.), в графическом режиме реализующую отображение в окне приложения объемного чертежа детали, заданной вариантом, в нескольких аксонометрических проекциях. Число применённых проекций определяет верхний уровень оценки по курсовой работе (проекции [1,2] – «удовлетворительно», проекции [1,2,3] – «хорошо», проекции [1,2,3,4] – «отлично»).

Список проекций:

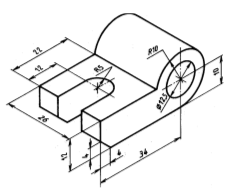
1. Аксонометрическая прямоугольная диаметрическая проекция

2. Аксонометрическая прямоугольная изометрическая проекция

3. Аксонометрическая косоугольная фронтальная изометрическая проекция

4. Аксонометрическая косоугольная горизонтальная изометрическая проекция.

Чертеж детали:



# 2. Теоретическая часть

При выполнении чертежей в ряде случаев оказывается необходимо наряду с изображением предметов в прямоугольных проекциях иметь и наглядные их изображения. Это нужно для обеспечения возможности более полно выявить конструктивные решения, заложенные в изображение предмета, правильно представить положение его в пространстве, оценить пропорции его частей и размеры.

Аксонометрическая проекция (или сокращенно аксонометрия) представляет собой один из видов наглядного изображения объектов, которое получается при параллельном проецировании. Суть этого способа проецирования заключается в том, что данный объект вместе с присоединенными к нему осями прямоугольных координат в пространстве проецируется параллельными лучами на некоторую произвольно выбранную плоскость, называемую плоскостью аксонометрических проекций.

Аксонометрические проекции более выразительны, чем прямоугольные на двух или трех плоскостях, так как передают одной проекцией на одной плоскости отображение сразу трех сторон объекта, что очень облегчает понимание его действительной формы и размеров. Таким образом, аксонометрия представляет собой такое изображение, в котором проекция любой точки в пространстве привязана к аксонометрическим осям, что делает изображение метрически определенным [2].

Аксонометрические проекции называют прямоугольными, если направление проецирования и проецирующие прямые перпендикулярны плоскости, на которую они проецируются, и косоугольными если направление проецирования не перпендикулярно плоскости аксонометрических проекций. Проекции аксонометрических осей на плоскость называют аксонометрическими осями, а проекции единицы измерения по осям – аксонометрическими единицами измерения. В зависимости от положения предмета и осей координат относительно плоскости проекций, а также в зависимости от направления проецирования единицы измерения проецируются в общем случае с искажением. Искажаются и размеры проецируемых предметов. Отношение длины аксонометрической единицы к ее истинной величине называют показателем или коэффициентом искажения для данной оси координат. Аксонометрические проекции называют изометрическими, если коэффициенты искажения по всем осям равны; диметрическими, если коэффициенты искажения по двум осям равны и триметрическими, если все коэффициенты различны.

В изометрических проекциях углы между аксонометрическими осями равны между собой и численно определяются как 3600 /3=1200. Коэффициенты искажения также равны между собой и равны 0,82. Это значит, что все линейные размеры геометрического образа, принимаемые параллельно осям декартовой системы координат, в изометрии искажаются в 0,82 раза (Рисунок 1).

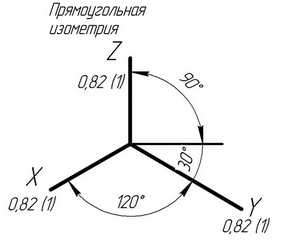


Рисунок 1 – угол между осями и коэффициент искажения в прямоугольной изометрии

В диметрических проекциях два угла между аксонометрическими осями равны по 131°25' и третий угол составляет 97°10', причем ориентация осей может быть левой и правой.

Коэффициенты искажения по аксонометрическим осям соответственно равны: два по 0,94 и третий – 0,47. Это значит, что линейные размеры геометрического образа, принимаемые параллельно декартовым осям координат, в диметрии соответственно искажаются в 0,94 и в 0,47 раза [1].

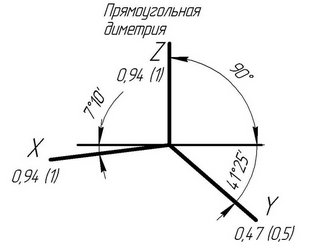


Рисунок 2 – угол между осями и коэффициент искажения в прямоугольной диметрии

Кроме указанных выше ортогональных изометрической и диметрической проекций, применяемых наиболее часто Государственный стандарт, предусматривает также применение следующих видов косоугольных аксонометрических проекций: косоугольная горизонтальная изометрия, косоугольная фронтальная изометрия (Рисунок 3).

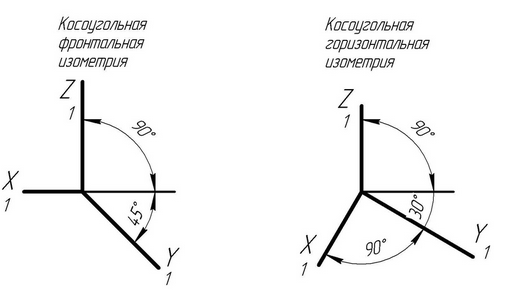


Рисунок 3 - угол между осями и коэффициент искажения в косоугольной фронтальной и горизонтальной изометрии

# 3. Проектирование

Для построения заданных проекций воспользуемся системой автоматизированного проектирования Компас 3D. Прямоугольная изометрия и прямоугольная диметрия строятся автоматически по трехмерной модели детали, поэтому сначала необходимо построить деталь. Для этого создадим новую деталь, на плоскости YOZ создадим эскиз и начертим в нем профильную проекцию заданной детали с соблюдением всех размеров. После этого с помощью функции элемент выдавливания создаем трехмерную деталь. Затем с помощью функции вырезать выдавливанием вырезаем отвертие в нижней части детали (Рисунок 4).

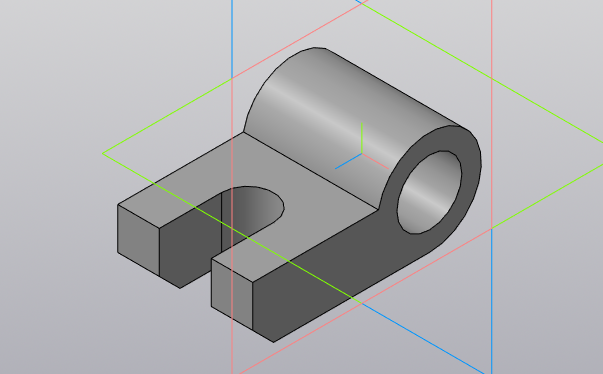


Рисунок 4 – Готовая деталь

После создания модели детали создадим чертеж и с помощью функции Стандартные виды с модели поместим прямоугольную изометрическую и прямоугольную диметрическую проекции на чертеж (Рисунок 5).

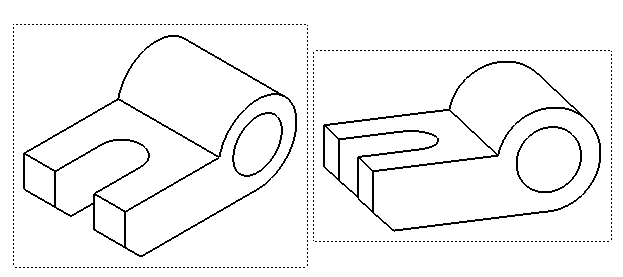


Рисунок 5 – Прямоугольная изометрия и диметрия

Косоугольную фронтальную и горизонтальную изометрию создадим вручную. Для этого с помощью вспомогательных линий зададим координатные оси с пересечениями под соответвующими углами и отложим на них действительные размеры детали (Рисунок 6, 7).

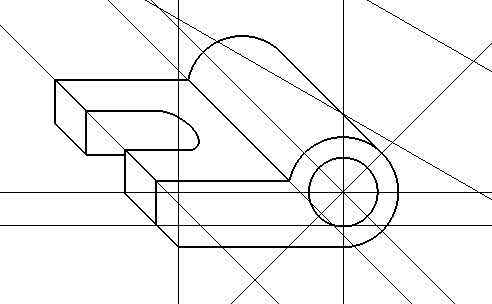


Рисунок 6 – Косоугольная фронтальная изометрия

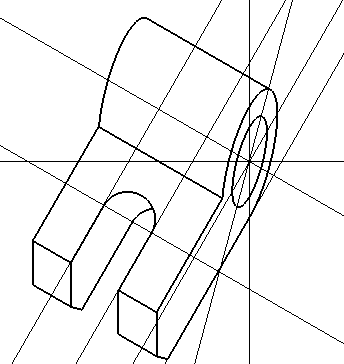


Рисунок 7 – Косоугольная горизонтальная изометрия

После создания проекций сохраним каждую из них в формате PNG для дальнейшего использования в программе.

# 4. Программная реализация

Для реализации программы, реализующей программный вывод созданных проекций выоспользуемся языком C# и WPF. Сначала создадим приложение WPF. Приложение будет содержать два окна. В главном окне будут располагаться 4 кнопки при нажатии на которые будет вызываться второе окно с изображением проекции. Картинки с проекциями поместим в папку ресурсов внутри проекта. Для каждой из кнопок напишем обработчик, который будет передавать во второе окно ссылку на соответствующий рисунок и вызывать метод для его отображения (Рисунок 8, 9).

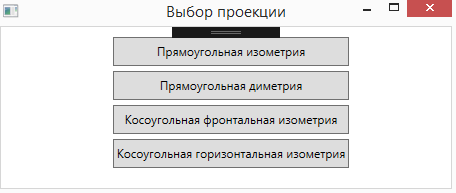


Рисунок 8 – Главное окно приложения

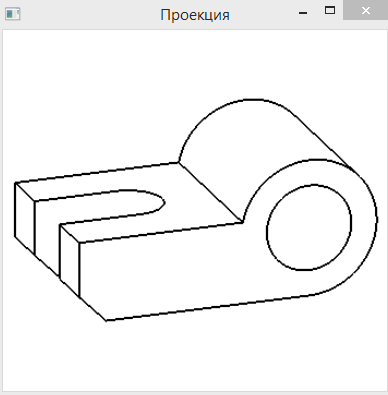


Рисунок 9 – Вызываемое окно приложения

# Заключение

В ходе выполнения курсовой работы мы познакомились с построением аксонометрических проекций, построили различные виды проекции для заданной детали. Разработали приложение, выводящее проекции на экран по нажатию на кнопку.

# Используемые ресурсы

1. Н.В. Васина Методические указания для самостоятельной работы студентов по дисциплине «Основы черчения и технический рисунок»/ Н.В. Васина. - Тула, 2013.. – 32 с.
2. К.С. Ившин Аксонометрические проекции учебно-методическое пособие/ К.С. Ившин. - ФГБОУ ВПО «Удмуртский государственный университет », 2012.. – 51 с.
3. https://metanit.com/sharp/wpf/ - Руководство по WPF (Дата обращения: 11.12.2018)
4. https://professorweb.ru/my/wpf/ - элементы управления WPF (Дата обращения: 11.12.2018)

# Приложение

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace GeomMod

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

TaskWindow tw = new TaskWindow();

tw.image.Source = new BitmapImage(new Uri("/Resources/1.png", UriKind.Relative)) {

CreateOptions = BitmapCreateOptions.IgnoreImageCache

};

tw.Show();

tw.Owner = this;

}

private void button2\_Click(object sender, RoutedEventArgs e)

{

TaskWindow tw = new TaskWindow();

tw.image.Source = new BitmapImage(new Uri("/Resources/2.png", UriKind.Relative))

{

CreateOptions = BitmapCreateOptions.IgnoreImageCache

};

tw.Show();

tw.Owner = this;

}

private void button3\_Click(object sender, RoutedEventArgs e)

{

TaskWindow tw = new TaskWindow();

tw.image.Source = new BitmapImage(new Uri("/Resources/3.png", UriKind.Relative))

{

CreateOptions = BitmapCreateOptions.IgnoreImageCache

};

tw.Show();

tw.Owner = this;

}

private void button4\_Click(object sender, RoutedEventArgs e)

{

TaskWindow tw = new TaskWindow();

tw.image.Source = new BitmapImage(new Uri("/Resources/4.png", UriKind.Relative))

{

CreateOptions = BitmapCreateOptions.IgnoreImageCache

};

tw.Show();

tw.Owner = this;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace GeomMod

{

/// <summary>

/// Логика взаимодействия для TaskWindow.xaml

/// </summary>

public partial class TaskWindow : Window

{

//public string imageUri { get; set; }

public TaskWindow()

{

InitializeComponent();

}

}

}