Міністерство освіти і науки України

Національний авіаційний університет

Кафедра прикладної математики

**Лабораторна робота № 1**

**Виконав:**

Студент ПМ-351

Здота Д.В.

**Перевірив:**

Юрчук І.А.

Київ 2016

Зміст

[Тема: 2](#_Toc464687854)

[Мета: 3](#_Toc464687855)

[Завдання: 3](#_Toc464687856)

[Теоретичні відомості 4](#_Toc464687857)

[Виконання роботи 4](#_Toc464687858)

[Код програми: 4](#_Toc464687859)

[Обчислювальна ефективність алгоритму : 7](#_Toc464687860)

[Висновки: 8](#_Toc464687861)

Тема: Побудова двомірних геометричних об’єктів.

Мета: Вивчити афінні перетворення на площині та вміти застосовувати їх до геометричних конструкцій. Вміти реалізувати довільні рухи а площині як композицію повороту, масштабування, перенесення, та дзеркального відображення.

# Завдання:

1. Вивчити всі типи афінних перетворень на площині та їх матричне представлення в однорідних координатах.
2. Створити програмне забезпечення для реалізації повороту, зсуву, непропорційного розтягу (стиску) та дзеркального відображення відносно прямої Ах + Ву +С = 0геометричного об’єкту G, де G визначений для кожного студента окремо згідно його варіанту дотримуючись наступних вимог:

* Значення параметрів А, В та С задаються користувачем.
* Розмір та координати вузлів об’єкту G автор програми визначає на свій розсуд, виходячи з параметрів монітору та естетичних міркувань.

***ОБ’ЄКТ G -*** трапеція

# Теоретичні відомості

**Афінні перетворення**

Зазвичай [лінійна алгебра](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B0_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0" \o "Лінійна алгебра) використовує матриці для представлення [лінійних перетворень](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F" \o "Лінійне перетворення), і векторну суму для представлення [паралельних перенесень](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%BD%D0%B5%D1%81%D0%B5%D0%BD%D0%BD%D1%8F" \o "Паралельне перенесення). За допомогою[розширеної матриці](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B0_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D1%8F) можливо представити і те, і те як матричний добуток. Ця техніка вимагає розширити всі вектори додаванням «1» в кінці, всі матриці розширюються додаванням рядка нулів знизу, і колонки — вектора переноса — справа, а також одиниці в нижній правий кут. {\displaystyle {\vec {y}}=A{\vec {x}}+{\vec {b}}.}

Таке представлення показує набір оборотних афінних перетворень як [напівпрямий добуток](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%BF%D1%96%D0%B2%D0%BF%D1%80%D1%8F%D0%BC%D0%B8%D0%B9_%D0%B4%D0%BE%D0%B1%D1%83%D1%82%D0%BE%D0%BA" \o "Напівпрямий добуток) *Kn* і GL(*n*, *k*). Афінні перетворення утворюють [групу](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%BF%D0%B0_(%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0)" \o "Група (алгебра)) щодо операції [композиції відображень](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D1%96%D1%8F_%D0%B2%D1%96%D0%B4%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C" \o "Композиція відображень). Ця група називається афінною групою.

Зазвичай матрично-векторний добуток завжди відображає початок координат на початок координат, і, таким чином, не може представляти перенесення, яке обов'язково переносить початок координат в іншу точку. Додаванням «1» до кожного вектора, вважаємо простір відображенним на підмножину простору з одним додатковим виміром. В цьому просторі, початковий простір займає підмножину в якій останній індекс 1. Таким чином початок координат початкового простору буде знаходитися в (0,0, … 0, 1). Перенесення всередині початкового простору в термінах лінійного перетворення простору з більшою кількістю вимірів стає можливим. Це є приклад [однорідних координат](https://uk.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%80%D1%96%D0%B4%D0%BD%D1%96_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82%D0%B8).

Перевагою використання однорідних координат є те, що можливо комбінувати будь-яку кількість перетворень в одне шляхом перемноження матриць. Ця можливість використовується графічними програмами.

# Виконання роботи

## Код програми:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Text.RegularExpressions;

namespace Grafika.laba.\_1

{

public partial class Form1 : Form

{

Bitmap DrawArea;

Graphics g;

Pen mypen;

int[,] koor = { { 20, 70 }, { 120, 70 }, { 80, 20 }, { 60, 20 } };

public Form1()

{

InitializeComponent();

DrawArea = new Bitmap(pictureBox1.Size.Width, pictureBox1.Size.Height);

g = Graphics.FromImage(DrawArea);

mypen = new Pen(Brushes.Black);

}

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

{

koor =new int[4,2]{ { 20, 70 }, { 120, 70 }, { 80, 20 }, { 60, 20 } };

g.Clear(Color.White);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);/\*

g.Clear(Color.White);\*/

/\*

g.Dispose();\*/

pictureBox1.Image = DrawArea;

}

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

{

for (int i = 0; i < 4;i++ )

{

if ((koor[i, 0] += Convert.ToInt32(XzsuvtextBox.Text)) < 0 || koor[i, 0] > pictureBox1.Width ||

(koor[i, 1] += Convert.ToInt32(YzsuvtextBox.Text))<0 || koor[i,1]>pictureBox1.Height)

{

MessageBox.Show("Точка вийшла за екран");

koor[i, 0] -= Convert.ToInt32(XzsuvtextBox.Text);

koor[i, 1] -= Convert.ToInt32(YzsuvtextBox.Text);

return;

}

}

g.Clear(Color.White);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);

pictureBox1.Image = DrawArea;

}

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

{

for (int i = 0; i < 4; i++)

{

koor[i, 0] = (int)((koor[i, 0] - Math.Min(koor[0, 0], koor[3, 0])) \* Convert.ToDouble(textBox1.Text) + Math.Min(koor[0, 0], koor[3, 0]));

koor[i, 1] = (int)((koor[i, 1] - Math.Min(koor[0, 1],koor[3, 1])) \* Convert.ToDouble(textBox2.Text) + Math.Min(koor[0, 1], koor[3, 1]));

/\*if (koor[i, 0] < 0 || koor[i, 0] > pictureBox1.Width ||

koor[i, 1] < 0 || koor[i, 1] > pictureBox1.Height)

{

MessageBox.Show("Точка вийшла за екран");

koor = new int[4, 2] { { 20, 70 }, { 120, 70 }, { 80, 20 }, { 60, 20 } };

return;

}\*/

}

g.Clear(Color.White);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);

pictureBox1.Image = DrawArea;

}

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

{

g.Clear(Color.White);

g.DrawEllipse(new Pen(Brushes.Red), Convert.ToInt32(textBox3.Text)-1, Convert.ToInt32(textBox4.Text)-1, 2, 2);

for (int i = 0; i < 4; i++)

{

int X = koor[i, 0]-Convert.ToInt32(textBox3.Text),

Y = koor[i, 1] - Convert.ToInt32(textBox4.Text);

koor[i, 0] = (int)(Math.Cos(Convert.ToDouble(textBox5.Text) \* Math.PI / 180.0) \* X - Math.Sin(Convert.ToInt32(textBox5.Text) \* Math.PI / 180.0) \* Y + Convert.ToDouble(textBox3.Text));

koor[i, 1] = (int)(Math.Cos(Convert.ToDouble(textBox5.Text) \* Math.PI / 180.0) \* Y + Math.Sin(Convert.ToInt32(textBox5.Text) \* Math.PI / 180.0) \* X + Convert.ToDouble(textBox4.Text));

/\*if (koor[i, 0] < 0 || koor[i, 0] > pictureBox1.Width ||

koor[i, 1] < 0 || koor[i, 1] > pictureBox1.Height)

{

MessageBox.Show("Точка вийшла за екран");

koor = new int[4, 2] { { 20, 70 }, { 120, 70 }, { 80, 20 }, { 60, 20 } };

return;

}\*/

}

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);

pictureBox1.Image = DrawArea;

}

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

{

double a = Convert.ToDouble(textBox6.Text);

double b = Convert.ToDouble(textBox7.Text);

double C = Convert.ToDouble(textBox8.Text);

g.Clear(Color.White);

if(Math.Abs(b)>0.05)

g.DrawLine(new Pen(Brushes.Red), 0, (int)(-C / b), pictureBox1.Width, (int)(-(C + a \* pictureBox1.Width) / b));

else

g.DrawLine(new Pen(Brushes.Red), (int)(-(C / a)), 0, (int)(-(C + b \* pictureBox1.Height) / a), pictureBox1.Height);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);

double a2=100, b2;

if (Math.Abs(b) > 0.05)

b2 = -a \* a2 / b;

else

{

b2 = 100;

a2 = -b \* b2 / a;

}

for (int i = 0; i < 4; i++)

{

double C2 = (-b2 \* koor[i, 1] - a2 \* koor[i, 0]);

int Y3;

int X3;

if (Math.Abs(b) < 0.02)

{

X3 = (int)(-C / a);

Y3 = koor[i, 1];

}

else if (Math.Abs(a) < 0.02)

{

Y3 = (int)(-C / b);

X3 = koor[i, 0];

}

else

{

Y3 = (int)((C / a - C2 / a2) / (b2 / a2 - b / a));

X3 = (int)(-(C2 + b2 \* Y3) / a2);

g.FillEllipse(Brushes.Blue, X3 - 2, Y3 - 2, 4, 4);

g.DrawLine(new Pen(Brushes.Green),2 \* X3 - koor[i, 0], 2 \* Y3 - koor[i, 1], koor[i, 0], koor[i, 1]);

koor[i, 0] = 2 \* X3 - koor[i, 0];

koor[i, 1] = 2 \* Y3 - koor[i, 1];

}

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[1, 0], koor[1, 1]);

g.DrawLine(mypen, koor[2, 0], koor[2, 1], koor[3, 0], koor[3, 1]);

g.DrawLine(mypen, koor[0, 0], koor[0, 1], koor[3, 0], koor[3, 1]);

pictureBox1.Image = DrawArea;

}

}

}

## Обчислювальна ефективність алгоритму :

***O(n)*.**

# Висновки:

1. Під час виконання лабораторної роботи я розглянув та вивчив афінні перетворення . В своїй роботі я реалізував її алгоритми.
2. В моєму програмному забезпечені алгоритм афінні перетворення використовуються для дій над заданим точками об’єктом.
3. Після реалізації програмного забезпечення я провів його тестування та прийшов до висновків що воно є досить швидким та використовує дуже мало ресурсів комп’ютера.
4. Розглянувши та проаналізувавши результати тестування а також код програмного забезпечення, я встановив обчислювальну складність алгоритму яка дорівнює.