Звіт про виконання комп’ютерного практикуму з дисципліни

«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

| **Студент:** Рудяк Ганна Олександрівна | **Група:** ОН-21 | **ст. 2** |
| --- | --- | --- |
| **Комп’ютерний практикум № 10** | **Варіант:** 2 |

**Тема: РОЗРОБКА ГРАФІЧНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ WPF. ЧАСТИНА 4. ПОБУДОВА ГРАФІКІВ ЗА ДОПОМОГОЮ БІБЛІОТЕКИ КЛАСІВ OXYPLOT**

**Мета:** Познайомитися з особливостями використання сторонніх бібліотек класів у додатках написаних мовою програмування C#. Отримати досвід відображення графіків за допомогою бібліотеки класів OxyPlot.

**Індивідуальне завдання:** Відповідно до завдання практичної роботи No1.

**Текст програми Class1.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ккпп10

{

public class Class1

{

[Serializable]

public class Region : IComparable<Region> //специфікатор відкритого доступу

{

internal string Name;

internal double Land;

internal int People;

internal string Centr;

internal int CentrPeople;

public string name

{

get { return Name; }

set { Name = value; }

}

public double land

{

get { return Land; }

set { Land = value; }

}

public int people

{

get { return People; }

set { People = value; }

}

public string centr

{

get { return Centr; }

set { Centr = value; }

}

public void SetName(string st)//метод для зчитування значень закритих полів

{

Name = st;

}

public string GetName()//методи задання значень закритих полів

{

return Name;

}

public void SetLand(double db)//метод для зчитування значень закритих полів

{

if (db <= 0)

{

Land = 0;

}

else

{

Land = db;

}

}

public double GetLand()//методи задання значень закритих полів

{

return Land;

}

public double Data { get; set; }//автоматична властивість

public double Neightbours { get; set; }//автоматична властивість

public double PeopleObl//властивість доступна тільки для читання

{

get { return People - CentrPeople; }

}

public override string ToString()//метод ToString() що забезпечує отримання інформації про клас у текстовому вигляді

{

return string.Format("Область: {0}\nПлоща: {1} км^2\nНаселення: {2} осіб\n" +

"Обласний центр: {3}\nНасення обласного центру: {4} осіб\nДата створення:{5}\n" +

"Сусідні області:{6}\n", Name, Land, People, Centr, CentrPeople, Data, Neightbours, PeopleObl);

}

public Region(string name, double land, int People, string Centr, int CentrPeople, double data, double neightbours)

{

Name = name;

Land = land;

this.People = People;

this.Centr = Centr;

this.CentrPeople = CentrPeople;

Data = data;

Neightbours = neightbours;

}

public Region() : this(" ", 0, 0, " ", 0, 0, 0)

{

}

public int CompareTo(Region other)//визначаємо площа якої областы найбільша

{

if (Land == other.Land)

{ return 0; }

if (Land < other.Land)

{ return 1; }

else

{ return -1; }

}

}

}

}

**Текст програми MainWindow.xaml**

<Window x:Class="ккпп10.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:ккпп10"

xmlns:oxy="http://oxyplot.org/wpf"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<DockPanel LastChildFill="False" Background="#FFC6EBFA">

<Menu Width="795" DockPanel.Dock="Top" Height="26" Background="#FFC6EBFA">

<MenuItem Header="Файл" HorizontalAlignment="Center" VerticalAlignment="Center" Width="77" HorizontalContentAlignment="Center" Padding="15,0,6,0" FontSize="18" Background="#FFB3D6DF" Height="30" FontFamily="Bahnschrift" BorderBrush="#FF15093B">

<MenuItem Header="Відкрити" Click="Open"/>

<MenuItem Header="Зберегти" Click="Save"/>

<MenuItem Header="Зберегти опис" Click="Savedescr"/>

<MenuItem Header="Зберегти графік" Click="SavePlot"/>

<Separator/>

<MenuItem Header="Вихід" Click="Close"/>

</MenuItem>

<MenuItem Header=" Справка" Click="Help" Width="88" Height="30" HorizontalAlignment="Left" VerticalAlignment="Center" Background="#FFB3D6DF" FontSize="18" Padding="3,0,6,0" FontFamily="Bahnschrift" BorderBrush="#FF4B1212"/>

<MenuItem Header="Зберегти список" Click="SaveList" Width="152" Height="30" HorizontalAlignment="Left" VerticalAlignment="Center" Background="#FFB3D6DF" FontSize="18" Padding="3,0,6,0" FontFamily="Bahnschrift" BorderBrush="#FF4B1212"/>

</Menu>

<ToolBar Width="795" DockPanel.Dock="Top" Height="26" FontFamily="Bahnschrift" Background="#FF76ABDF">

<ToolBar.BorderBrush>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush>

</ToolBar.BorderBrush>

<Button Content="Зберегти" HorizontalAlignment="Center" VerticalAlignment="Center" Click="Save" FontSize="13"/>

<Button Content="Відкрити" HorizontalAlignment="Center" VerticalAlignment="Center" Click="Open" FontSize="13"/>

<Button Content="Зберегти опис" HorizontalAlignment="Center" VerticalAlignment="Center" Click="Savedescr" FontSize="13"/>

<Button Content="Зберегти cписок" HorizontalAlignment="Center" VerticalAlignment="Center" Click="SaveList" FontSize="13"/>

<Button Content="Зберегти графік" Click="SavePlot" FontSize="13"/>

</ToolBar>

<TabControl Width="788" DockPanel.Dock="Top" Height="378" >

<TabItem Header="Введення" FontFamily="Bahnschrift" Background="#FF8F9AF7" Margin="-2,-3,-2,0" FontSize="14">

<Grid Background="#FFF3F6F8">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="96\*"/>

<ColumnDefinition Width="26\*"/>

<ColumnDefinition Width="269\*"/>

<ColumnDefinition Width="391\*"/>

</Grid.ColumnDefinitions>

<Label Content="Назва області:" HorizontalAlignment="Left" VerticalAlignment="Center" FontSize="18" Width="360" FontFamily="Bahnschrift" Grid.ColumnSpan="3" Margin="16,0,0,0"/>

<Label Content="Площа, км^2:" HorizontalAlignment="Left" Grid.Row="1" VerticalAlignment="Center" FontSize="18" Width="360" FontFamily="Bahnschrift" Grid.ColumnSpan="3" Margin="16,0,0,0"/>

<Label Content="Населення, осіб:" HorizontalAlignment="Left" Grid.Row="2" VerticalAlignment="Center" FontSize="18" FontFamily="Bahnschrift" Width="360" Grid.ColumnSpan="3" Margin="16,0,0,0"/>

<Label Content="Обласний центр:" HorizontalAlignment="Left" Grid.Row="3" VerticalAlignment="Center" FontSize="18" Width="360" FontFamily="Bahnschrift" Grid.ColumnSpan="3" Margin="16,0,0,0"/>

<Label Content="Населення обласного центру, осіб:" HorizontalAlignment="Left" Grid.Row="4" VerticalAlignment="Top" FontSize="18" Width="360" FontFamily="Bahnschrift" Margin="16,2,0,0" Grid.ColumnSpan="3"/>

<Label Content="Дата створення:" HorizontalAlignment="Center" Grid.Row="5" VerticalAlignment="Center" FontSize="18" Width="360" FontFamily="Bahnschrift" Grid.ColumnSpan="3" RenderTransformOrigin="0.5,-1.224"/>

<Label Content="Кількість сусідніх обласей:" HorizontalAlignment="Center" Grid.Row="6" VerticalAlignment="Center" FontSize="18" Width="360" FontFamily="Bahnschrift" Grid.ColumnSpan="3"/>

<TextBox x:Name="name\_box" TextWrapping="Wrap" Text="Полтавська" Grid.Column="3" Margin="10,7,10,12" Height="25" Grid.Row="0" FontSize="16" FontFamily="Bahnschrift Light"/>

<TextBox x:Name="land\_box" TextWrapping="Wrap" Text="28748" Grid.Column="3" Margin="10,7,10,12" Height="25" Grid.Row="1" FontFamily="Bahnschrift Light" FontSize="16"/>

<TextBox x:Name="people\_box" TextWrapping="Wrap" Text="1413829" Grid.Column="3" Margin="10,6,10,13" Height="25" Grid.Row="2" FontFamily="Bahnschrift Light" FontSize="16"/>

<TextBox x:Name="centr\_box" TextWrapping="Wrap" Text="Полтава" Grid.Column="3" Margin="10,7,10,12" Height="25" Grid.Row="3" FontFamily="Bahnschrift Light" FontSize="16"/>

<TextBox x:Name="centpeople\_box" TextWrapping="Wrap" Text="219593" Grid.Column="3" Margin="10,8,10,10" Height="25" Grid.Row="4" FontFamily="Bahnschrift Light" FontSize="16"/>

<TextBox x:Name="data\_box" TextWrapping="Wrap" Text="10,01" Grid.Column="3" Margin="10,6,10,13" Height="25" Grid.Row="5" FontFamily="Bahnschrift Light" FontSize="16"/>

<TextBox x:Name="neight\_box" TextWrapping="Wrap" Text="5" Grid.Column="3" Margin="10,6,10,13" Height="25" Grid.Row="6" FontFamily="Bahnschrift Light" FontSize="16"/>

<Button x:Name="start" Content="Cтарт" HorizontalAlignment="Left" Grid.Row="7" VerticalAlignment="Center" Grid.ColumnSpan="4" RenderTransformOrigin="0.543,-0.659" Padding="5,1,1,1" Grid.IsSharedSizeScope="True" MinHeight="25" Width="790" Height="43" Foreground="#FF01020D" BorderBrush="#FF070A23" Background="#FF8F9AF7" FontFamily="Bahnschrift SemiBold" FontSize="24" FontWeight="Bold" Click="startclick" Margin="-4,0,0,0"/>

</Grid>

</TabItem>

<TabItem Header="Результат" FontFamily="Bahnschrift" BorderBrush="#FF6B7DF6" Background="#FF8F9AF7" Margin="-2,-4,-2,0" FontSize="14">

<TextBlock x:Name="result" HorizontalAlignment="Left" Grid.Row="8" TextWrapping="Wrap" Text="TextBlock" VerticalAlignment="Top" Grid.ColumnSpan="3" Width="790" Height="130" Grid.RowSpan="4" Padding="5,0,0,0" Margin="5,8,0,0"/>

</TabItem>

<TabControl>

<TabItem Header="TabItem">

<Grid Background="#FFE5E5E5"/>

</TabItem>

<TabItem Header="TabItem">

<Grid Background="#FFE5E5E5"/>

</TabItem>

</TabControl>

<TabItem Header="Список" FontFamily="Bahnschrift" BorderBrush="#FF6B7DF6" Background="#FF8F9AF7" Margin="-11,-4,6,0" FontSize="14">

<DataGrid x:Name="list" />

</TabItem>

<TabItem Header="Графік" FontFamily="Bahnschrift" BorderBrush="#FF6B7DF6" Background="#FF8F9AF7" Margin="-11,-4,6,0" FontSize="14">

<oxy:PlotView x:Name="RegionPlot" />

</TabItem>

</TabControl>

<TabControl Width="100">

<TabItem Header="TabItem">

<Grid Background="#FFE5E5E5"/>

</TabItem>

<TabItem Header="TabItem">

<Grid Background="#FFE5E5E5"/>

</TabItem>

</TabControl>

</DockPanel>

</Window>

**Текст програми MainWindow.xaml.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Formatters.Binary;

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;

using System.Xml.Linq;

using System.Xml.Serialization;

using Microsoft.Win32;

using OxyPlot;

using OxyPlot.Axes;

using OxyPlot.Series;

using static ккпп10.Class1;

namespace ккпп10

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

List<Region> transList = new List<Region>();

transList.Add(new Region("Київська", 28131, 1754284, "Київ", 2952301, 10.01, 4));

transList.Add(new Region("Полтавська", 28748, 1413829, "Полтава", 219593, 10.01, 5));

transList.Add(new Region("Харківська", 31415, 2694007, "Харків", 1421125, 10.01, 3));

transList.Add(new Region("Одеська", 33314, 2343749, "Одеса", 1010537, 27.02, 3));

transList.Add(new Region("Сумська", 23832, 1068247, "Суми", 256474, 10.01, 3));

list.ItemsSource = transList;

PlotModel MyPlot = new PlotModel();

RegionPlot.Model = MyPlot;

MyPlot.Title = "Графік залежності населення від площі";

transList.Sort();

LineSeries transSeries = new LineSeries();

foreach (Region transfr in transList)

{

transSeries.Points.Add(new DataPoint(transfr.Land, transfr.People/10));

}

MyPlot.Series.Add(transSeries);

LinearAxis x = new LinearAxis();

LinearAxis y = new LinearAxis();

x.Position = AxisPosition.Bottom;

x.Minimum = 0;

x.Title = "Площа області, км^2";

y.Position = AxisPosition.Left;

y.Minimum = 0;

y.Title = "Населення області, осіб";

MyPlot.Axes.Add(x);

MyPlot.Axes.Add(y);

}

Region region;

private void Open(object sender, RoutedEventArgs e)

{

var openDlg = new OpenFileDialog();

openDlg.Filter = "Файли трансформаторів | \*.rgn";

if (openDlg.ShowDialog() == true)

{

BinaryFormatter bf = new BinaryFormatter();

using (FileStream fs = new FileStream(openDlg.FileName, FileMode.Open))

{

Region region1 = (Region)bf.Deserialize(fs);

name\_box.Text = region1.Name;

land\_box.Text = region1.Land.ToString();

people\_box.Text = region1.People.ToString();

centr\_box.Text = region1.Centr;

centpeople\_box.Text = region1.CentrPeople.ToString();

data\_box.Text = region1.Data.ToString();

neight\_box.Text = region1.Neightbours.ToString();

}

}

}

private void Save(object sender, RoutedEventArgs e)

{

var saveDlg = new SaveFileDialog();

saveDlg.Filter = "Файли регіонів | \*.rgn";

if (saveDlg.ShowDialog() == true)

{

Region region = new Region();

region.Name = name\_box.Text;

region.Land = double.Parse(land\_box.Text);

region.People = int.Parse(people\_box.Text);

region.Centr = centr\_box.Text;

region.CentrPeople = int.Parse(centpeople\_box.Text);

region.Data = double.Parse(data\_box.Text);

region.Neightbours = double.Parse(neight\_box.Text);

BinaryFormatter bf = new BinaryFormatter();

using (FileStream fs = new FileStream(saveDlg.FileName, FileMode.Create))

{

bf.Serialize(fs, region);

}

}

}

private void Savedescr(object sender, RoutedEventArgs e)

{

var saveDlg = new SaveFileDialog();

saveDlg.Filter = "Текстові файли | \*.txt";

if (saveDlg.ShowDialog() == true)

{

using (StreamWriter sw = new StreamWriter(saveDlg.FileName))

{

sw.WriteLine(result.Text);

}

}

}

private void Close(object sender, RoutedEventArgs e)

{

Close();

}

private void Help(object sender, RoutedEventArgs e)

{

MessageBox.Show("Перевірка перевірки");

}

private void startclick(object sender, RoutedEventArgs e)

{

string name = name\_box.Text;

double land = double.Parse(land\_box.Text);

int People = int.Parse(people\_box.Text);

string Centr = centr\_box.Text;

int CentrPeople = int.Parse(centpeople\_box.Text);

double data = double.Parse(data\_box.Text);

double neightbours = double.Parse(neight\_box.Text);

Region region = new Region(name, land, People, Centr, CentrPeople, data, neightbours);

result.Text = region.ToString();

}

private void SaveList(object sender, RoutedEventArgs e)

{

var saveDlg = new SaveFileDialog();

saveDlg.Filter = "Файли XML | \*.xml";

if (saveDlg.ShowDialog() == true)

{

XmlSerializer xmls = new XmlSerializer(typeof(List<Region>));

using (FileStream fs = new FileStream(saveDlg.FileName, FileMode.Create))

{

xmls.Serialize(fs, list.ItemsSource);

}

}

}

private void SavePlot(object sender, RoutedEventArgs e)

{

var saveDlg = new SaveFileDialog();

saveDlg.Filter = "Файли PNG | \*.png";

if (saveDlg.ShowDialog() == true)

{

RegionPlot.SaveBitmap(saveDlg.FileName, (int)RegionPlot.Width, (int)RegionPlot.Height, OxyColors.Transparent);

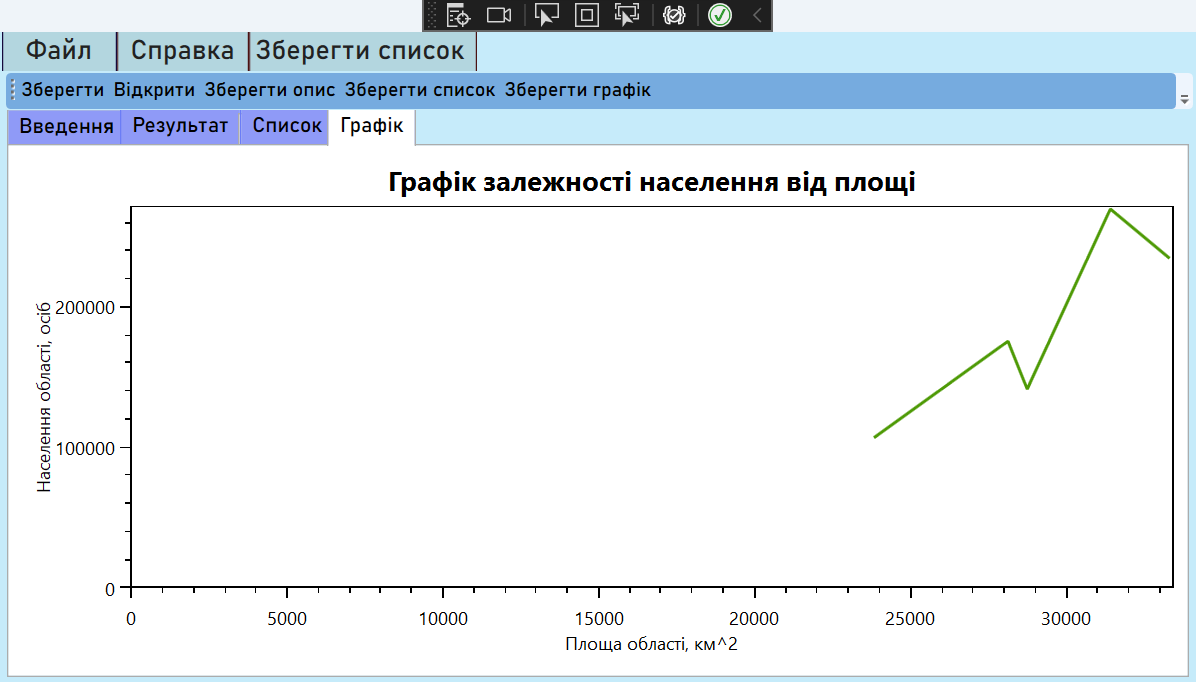
}

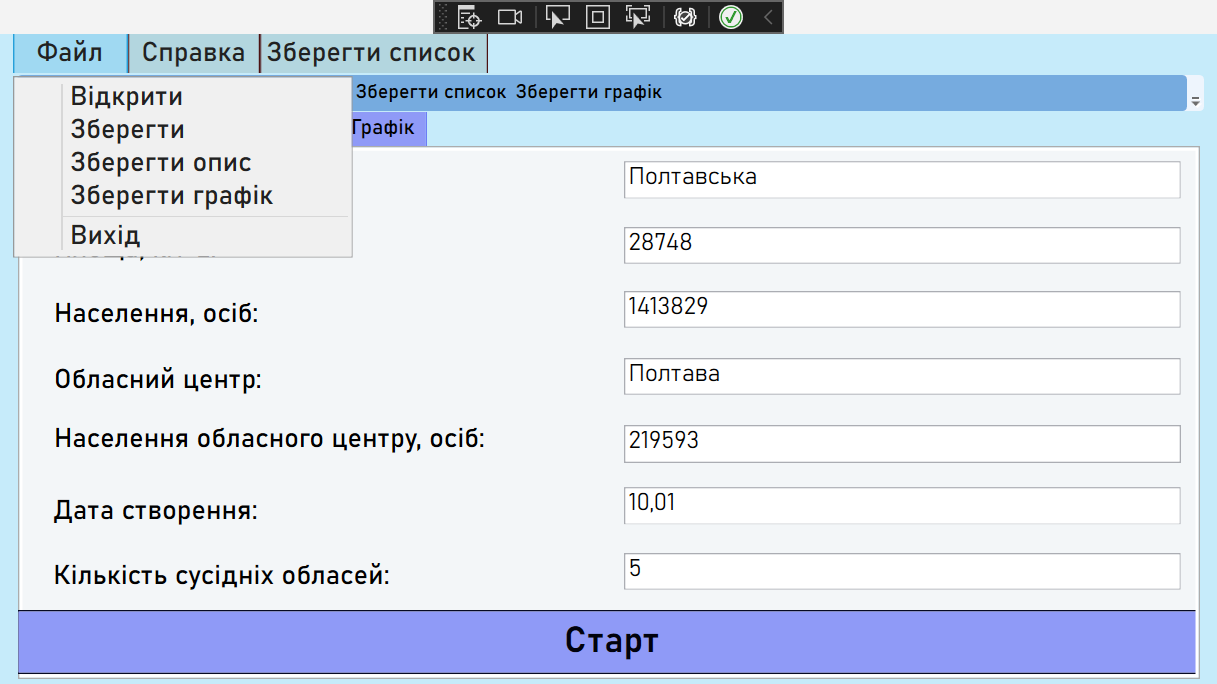
}

}

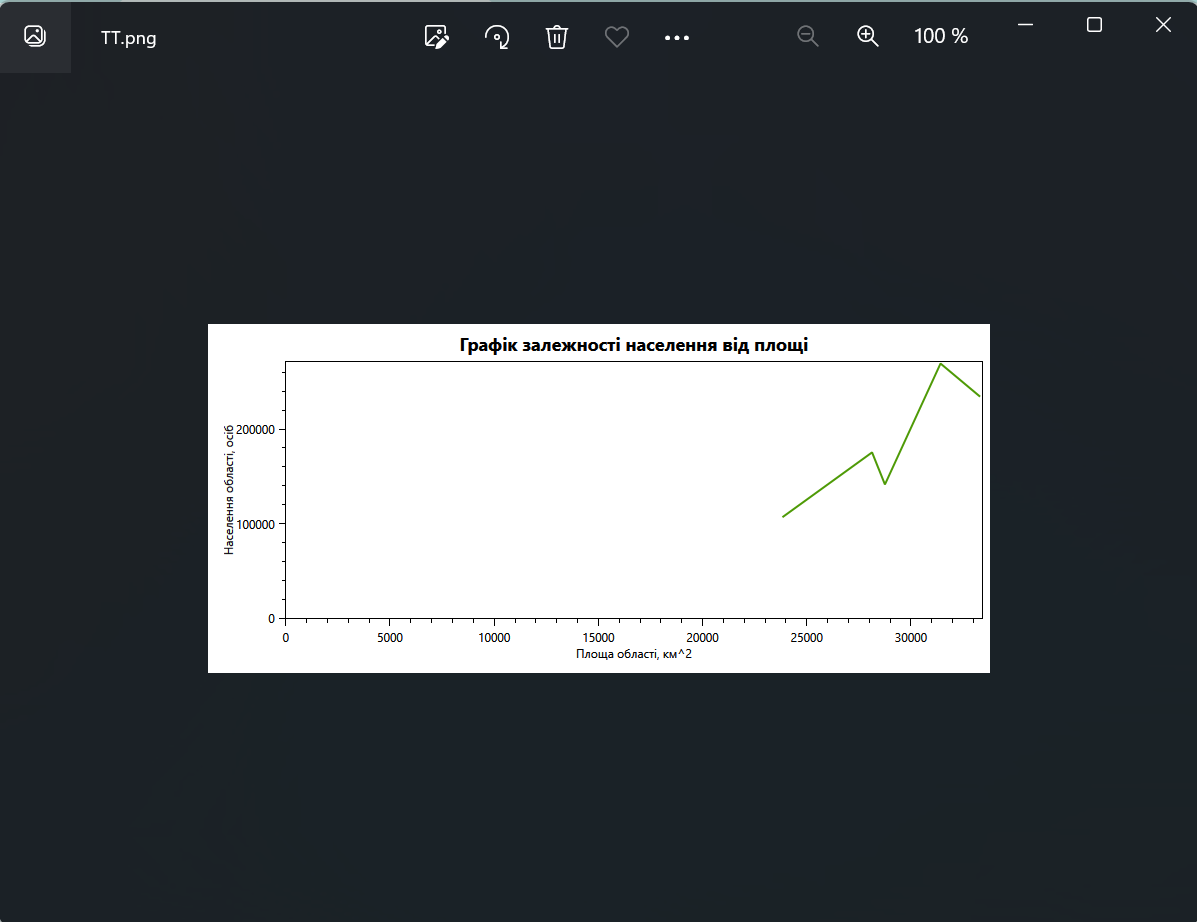
}

**Результати виконання програми**

****

****

****

****

| **Дата виконання:** 24.05.2022 | **Дата захисту:** |
| --- | --- |
| **Оцінка:** |  |
| **Викладач:** | **Підпис:** |