Звіт про виконання комп’ютерного практикуму з дисципліни

«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

|  |  |  |
| --- | --- | --- |
| **Студент:** Рудяк Ганна Олександрівна | **Група:** ОН-21 | **ст. 2** |
| **Комп’ютерний практикум № 6** | **Варіант:** 2 |

**Тема: ФАЙЛОВЕ ВВЕДЕННЯ ТА ВИВЕДЕННЯ. СЕРІАЛІЗАЦІЯ ОБ’ЄКТІВ**

**Мета:** Вивчити основні принципи роботи з файлами та серіалізації об’єктів в мові програмування С#. Отримати досвід збереження окремих екземплярів класу та колекцій і з допомогою бінарного форматера та у файлах XML.

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

**Текст програми**

using System;

using System.Collections.Generic;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

using System.Xml.Serialization;

namespace KP6

{

[Serializable]

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

{

private string Name;

double Land;

int People;

string Centr;

private int CentrPeople;

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, Neightbours, Data, 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; }

}

}

internal class Program

{

static void Main(string[] args)

{

//Створити список

var transList = new List<Region>();

//Додати елементи в список

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));

Region region1 = new Region("Київська", 28131, 1754284, "Київ", 2952301, 10.01, 4);

using (StreamWriter sw = new StreamWriter("C:\\kp6\\region1.txt"))

{

sw.WriteLine(region1);

}

BinaryFormatter bf = new BinaryFormatter();

using (FileStream fs = new FileStream("C:\\kp6\\region1.dat", FileMode.Create))

{

bf.Serialize(fs, region1);

}

Region regionz;

using (FileStream fs = new FileStream("C:\\kp6\\region1.dat", FileMode.Open))

{

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

}

Console.WriteLine(regionz);

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

using (FileStream fs = new FileStream("C:\\kp6\\region1.xml", FileMode.Create))

{

xmls.Serialize(fs, transList);

}

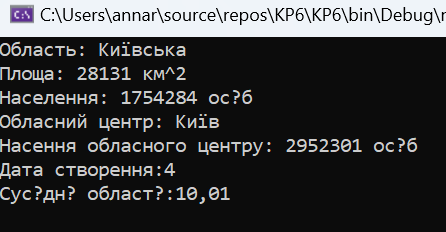
Console.ReadLine();

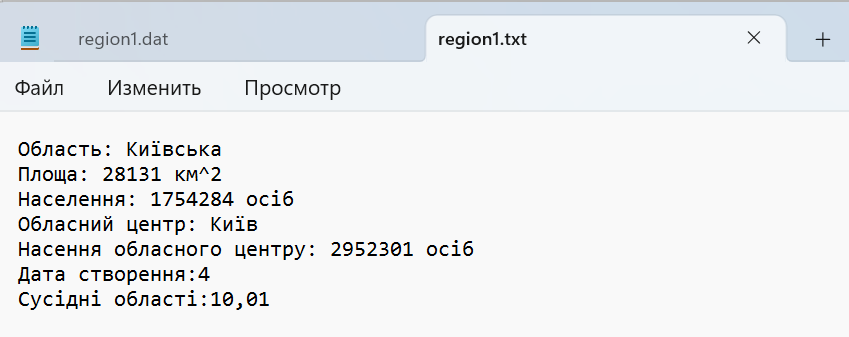
}

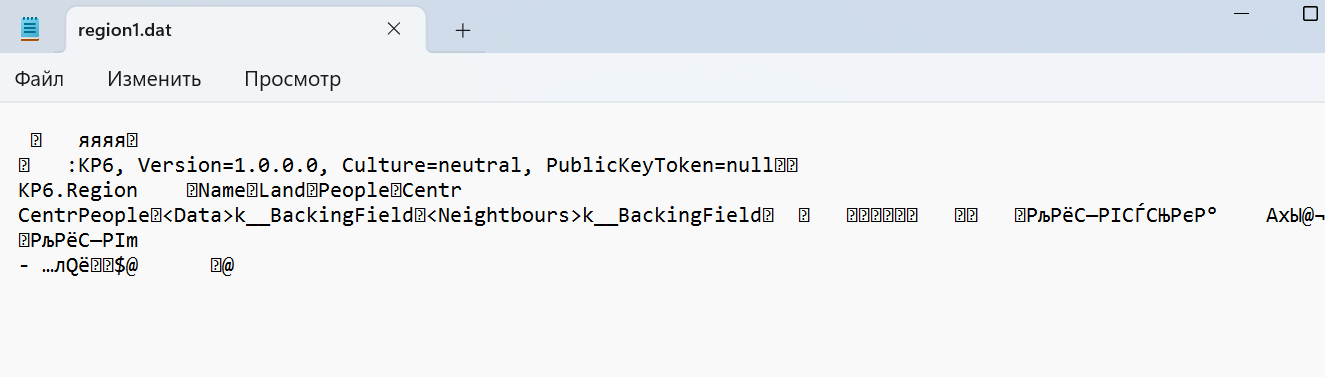
}

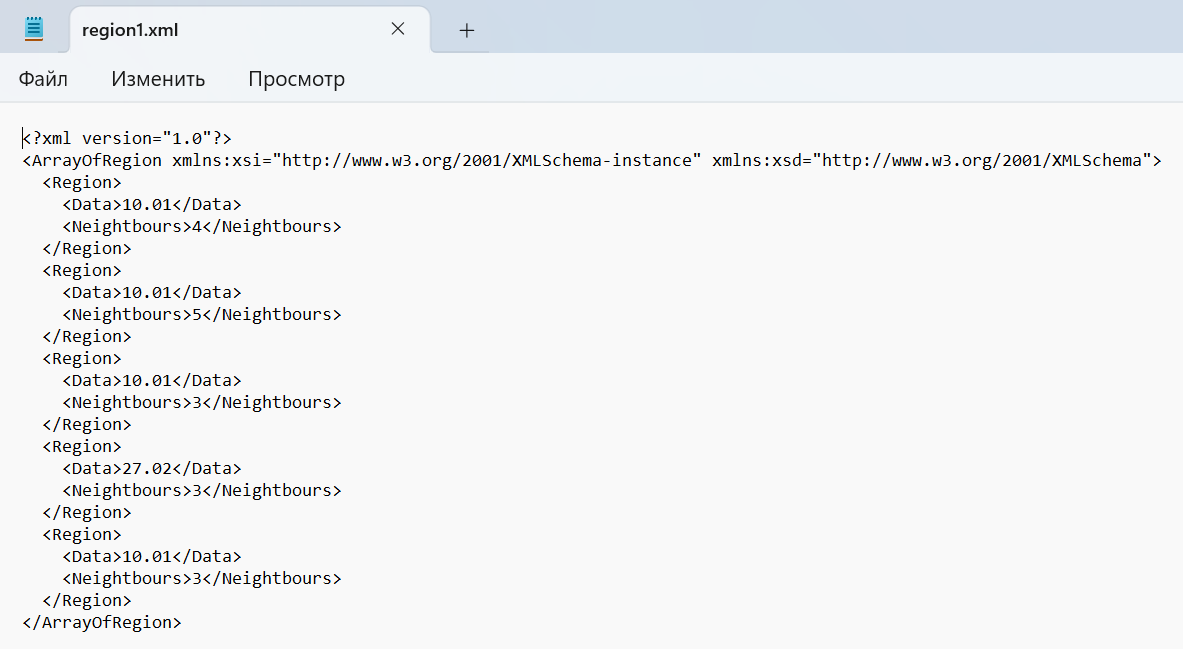
}

**Результати виконання програми**









|  |  |
| --- | --- |
| **Дата виконання:** 20.04.2022 | **Дата захисту:** |
| **Оцінка:** | **Підпис:** |
| **Викладач:** |  |