МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об’єктно-орієнтоване програмування»

на тему:

**«Файловий менеджер»**

студента І курсу групи ІПЗ-20-1

спеціальності 121 «Інженерія програмного забезпечення»

Головецького Євгенія Володимировича

(прізвище, ім’я та по-батькові)

Керівник ст.викл. Чижмотря О.В.

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Л. Левківський \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Марчук .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Чижмотря \_

(підпис) (прізвище та ініціали)

Житомир – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о.зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ А.В.Морозов

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р.

ЗАВДАННЯ

НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Головецькому Євгенію Володимировичу

1. Тема роботи: Розробка програми "Файловий менеджер"

керівник роботи: О.В. Чижмотря

1. Строк подання студентом: “ 25 ” травня 2020р.
2. Вихідні дані до роботи: Розробити програму “ Файловий менеджер ”
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КП\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2.

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1,2,3 | Левківський В.Л., ст. викладач каф. КН |  |  |
| 1,2,3 | Чижмотря О.В., ст. викладач каф. ІПЗ |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 17” лютого 2021

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсового проекту | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі | 03.05.2020 | виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 05.05.2020 | виконано |
| 3 | Формулювання технічного завдання | 06.05.2020 | виконано |
| 4 | Опрацювання літературних джерел | 07.05.2020 | виконано |
| 5 | Проектування структури | 08.05.2020 | виконано |
| 6 | Написання програмного коду | 09.05.2020 | виконано |
| 7 | Відлагодження | 10.05.2020 | виконано |
| 8 | Написання пояснювальної записки | 02.06.2020 | виконано |
| 9 | Захист | 04.06.2020 |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головецький Є.В

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.В

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Завданням на курсовой проект (роботу) було створення файлового менеджеру.

Пояснювальна записка до курсового проекту (роботи) на тему «Файловий менеджер» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 19 сторінках друкованого тексту.

Пояснювальна записка має 16 сторінок додатків.

Список використаних джерел містить 7 найменувань і займає 1 сторінку. В роботі наведено 22 рисунка. Загальний обсяг роботи 36 сторінок.

Ключові слова: ФАЙЛОВИЙ, МЕНЕДЖЕР, ДОДАТОК, ПРОГРАМА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КЛАС

Зміст

[Вступ 2](#_Toc73557139)

[РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 3](#_Toc73557140)

[1.1 Аналіз задачі, засобів та методів її вирішення 3](#_Toc73557141)

[1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту 3](#_Toc73557142)

[Висновки до першого розділу 6](#_Toc73557143)

[РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 7](#_Toc73557144)

[2.1 Проектування загального алгоритму роботи програми 7](#_Toc73557145)

[2.2 Розробка функціональних алгоритмів роботи програми 8](#_Toc73557147)

[2.3 Розробка програмного забезпечення 9](#_Toc73557148)

[Діаграма класів 9](#_Toc73557149)

[Висновок до другого розділу: в другому розділі я описав код основних методів, склав діаграму класів та прецедентів.Описав загальний алгоритм роботи програми та додаткових класів 10](#_Toc73557150)

[Розділ 3. Опис роботи з програмним додатком та його тестування. 11](#_Toc73557151)

[3.1 Опис роботи з програмним додатком 11](#_Toc73557152)

[3.2 Тестування роботи програмного забезпечення 17](#_Toc73557153)

[Висновки 19](#_Toc73557154)

[Список використаних джерел 19](#_Toc73557155)

[Додатки 20](#_Toc73557156)

# Вступ

Файловий менеджер – тема, яка буде актуальною до тих пір, поки існуе поняття файлу. Сьогодні майже неможливо уявити будь-який пристрій який б не працював з файлами і рядовому користувачу потрібен швидкий та зручний інтерфейс для можливості вносити будь-які зміни у файловій структурі.Видалення, створення, переміщення, копіювання та багато інших дій повинні бути подані користувачеві в розумілій формі так щоб навіть людина, яка не знайома з комп’ютером змогла розібратися та зробити те що їй потрібно

Об’єктом дослідження данної курсової роботи є файлова структура

Предмет дослідження – пошук зручних способів для роботи з файлами та їх аналіз

Мета дослідження – поглибити навички у роботі з мовою програмування С# та Windows Forms, навчитися працювати з файлами та створювати зручний інтерфейс

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## Аналіз задачі, засобів та методів її вирішення

Я для себе поставив таку задачу– розробити функціональний додаток з зрозумілим мінімалістичним інтерфейсом для зручної роботи з файловою системою компютера. Для вирішення задачі я використовував середовище Visual Studio 2019

## Аналіз існуючого програмного забезпечення за тематикою курсового проекту

Шукаючи приклади файлових менеджерів, саме перше на що я зробив – натиснув WIN+E.Переді мною відкрився «Провідник».Є стандартною програмою для роботи з файлами на операційній системі Windows 10.Додаток містить велику кількість корисного и не дуже функціоналу, наприклад пчинаючи від стандартих операцій типу «Створити», «Видалити» і т.д, закінчуючи зміною вигляду плиток,їх сортуванням та ще купою функціоналу

Його перевагами як на мене є

* Простота у використанні
* Доступнісь для кожного користувача 10 вінди
* Зрозумілий інтерфейс
* Приємний оку дизайн

Основні недоліки

* Відсутність спаренного вікна для роботи з файлами
* Невелика кількість функціоналу в порівнянні з аналогами

Нижче продемонстровано загальний вигляд інтефейсу «Провідника»

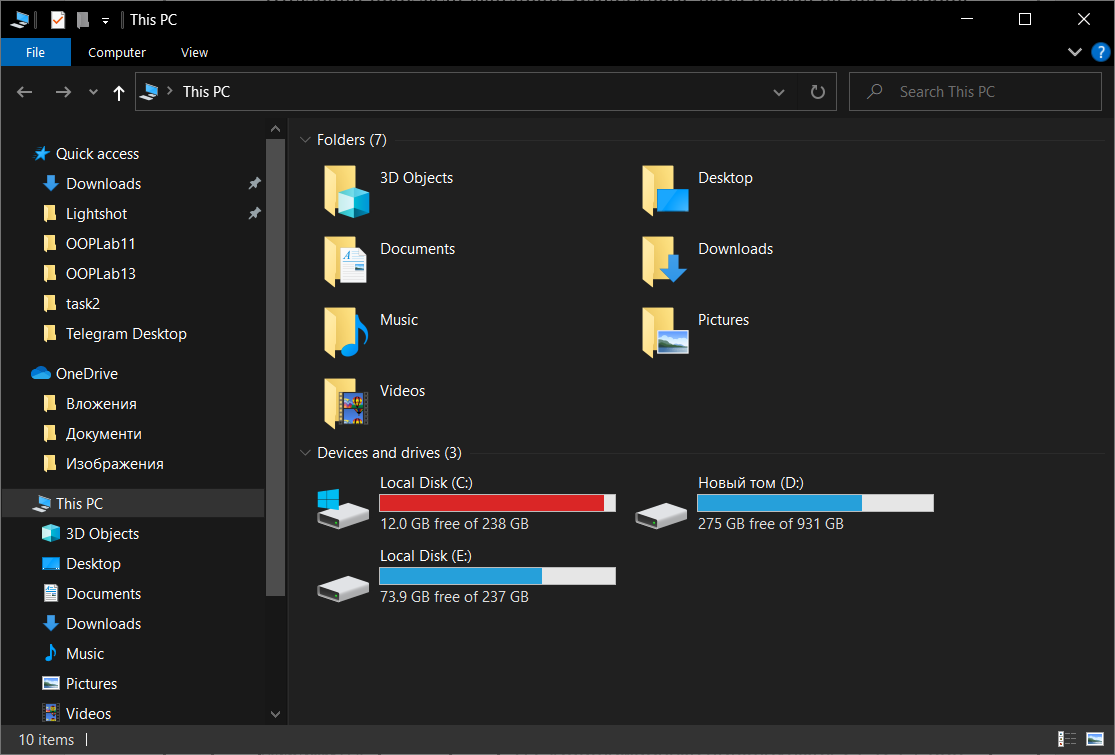


Рис 1.1. Загальний вигляд інтефейсу «Провідника»

Основним конкурентов «Провідника» та моїм «натхненням» є файловий менеджер «Total commander».Умовно безкоштовна програма величезною кількістью найрізноманітнішого функціоналу починаючи від можливості використання команд терміналу прямо у додатку закінчуючи можливістю працювати з віддаленими директоріями та дисками

Його перевагами як на мене є

* Насамперед функціонал
* Мінімалістичний дизайн
* Наявність двох вікон

Основні недоліки

* Складний інтерфейс
* Програма платна

Нижче продемонстровано загальний вигляд інтефейсу «Total Сommander»

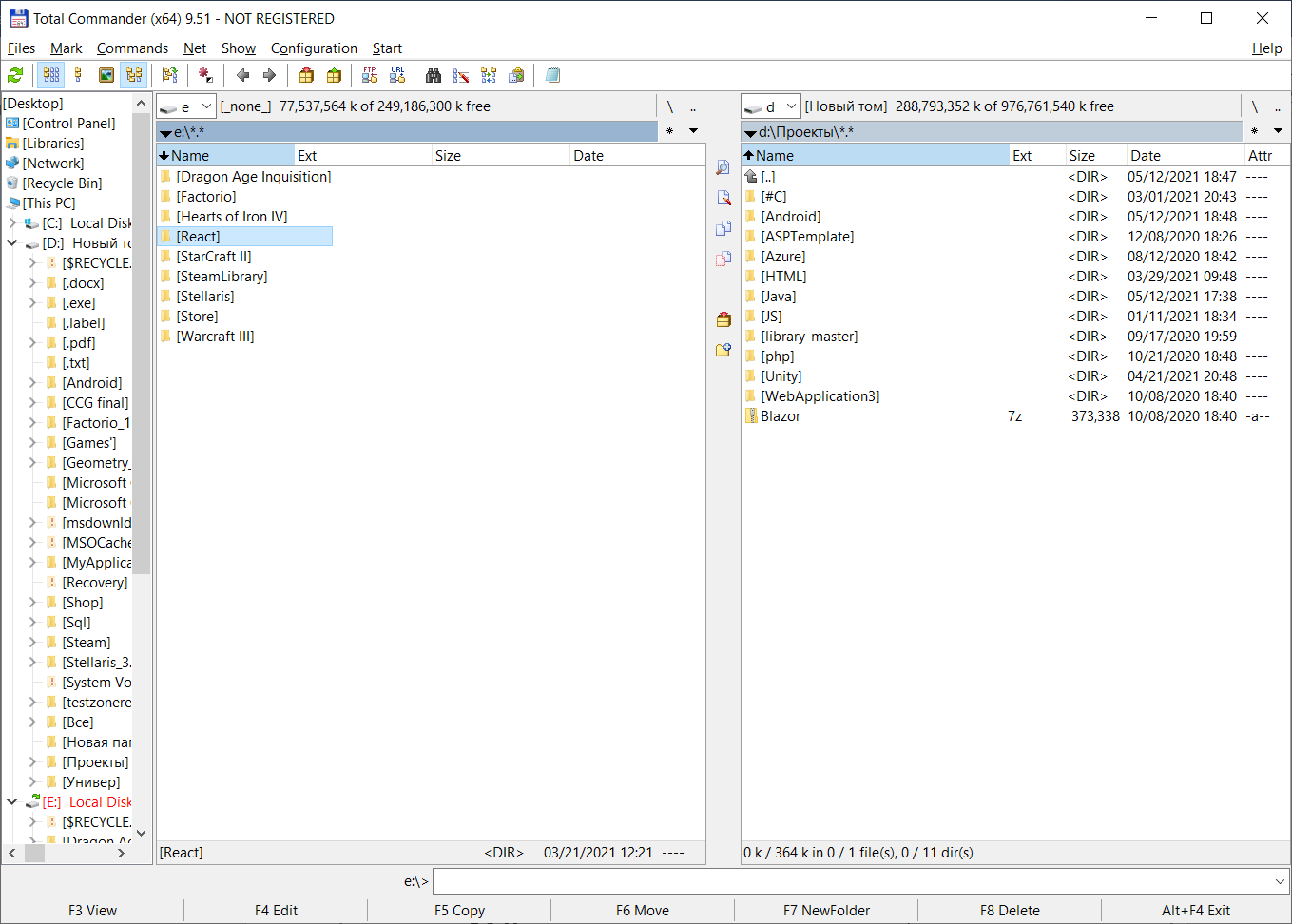


Рис 1.2. Загальний вигляд інтефейсу «Total Сommander»

FAR-manager – консольний файловий менеджер. Программа під Windows основним механізмом управління якої є консольні команди, також як і «Total Сommander» має двовіконний інтерфейс

Його перевагами як на мене є

* Опір на управління командами
* Наявність двох вікон
* Зручна робота з архівами

Осовні недоліки

* Опір на управління командами
* Дизайн програми

Нижче продемонстровано загальний вигляд інтефейсу «FAR-manager»

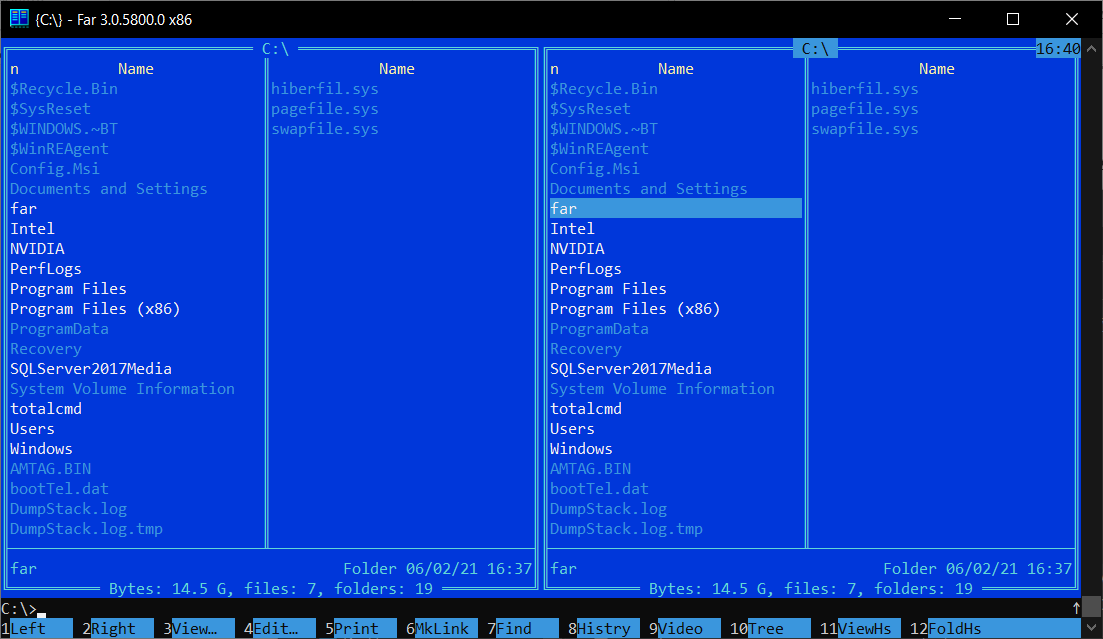


Рис 1.3. Загальний вигляд інтефейсу «FAR-manager»

## Висновки до першого розділу

В данному розділі я провів детальний аналіз аналогічних додатків, поставив задачу та описав необхідний до реалізаціїї функціонал

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Проектування загального алгоритму роботи програми

Для демострації функціоналу програми я створив діаграму прецендентів.

Діаграма продемонстрована нижче:

## 

Рис 2.1. Діаграмма прецендентів

Під час запуску програми користувач бачить перед собою два вікна з файлами та директоріями, і одразу ж може приступити до роботиз ними.Користувач мае можливість створювати папки,переміщати,копіювати,видаляти,перейменовувати файли.Також реалізована функція Drag&Drop;

## 2.2 Розробка функціональних алгоритмів роботи програми

Основне поле для роботи з програмою це 2 listView.Через них відбувається взаємодія з вайлами да функція Drag&Drop, оскільки стандартні listView мене не влаштовували, я прийняв рішення створити власний listView з деякими додатковими властивостями необхідність в яких виникала під час роботи програми.

Класс **MyListView**

|  |
| --- |
| class MyListView:ListView  public DirectoryInfo currentDirectory { get; set; }// обєкт якій містить поточну відкриту директорію на диску  public bool IsActive { get; set; }// флаг який показує активність listView  public TextBox StatmentBox { get; set; }// відповідний до listView TextBoxякий показуе шлях відкритої диркторії  } |

Данний клас внаслідуваний від стандартного listView

Також я додав до додатку можливість шифрувати та дешифрувати файли

Класс **Crypto**

|  |
| --- |
| public void Encrypt(string sourceFile, string destinationFile)  public void Decrypt(string sourceFile, string destinationFile)    private static void EncryptDecrypt(string sourceFile, string destinationFile, int[] key)  } //виконує битове шифрування файлів за значенням |

Також я додав можливість архівувати файли використовуючи сторонню бібліотеку ICSharpCode

Класс **WorkZip**

|  |
| --- |
| class WorkZip  public void CreateSample(string outPathname, string password, string folderName) // створюе архів  public void ExtractZipFile(string archivePath, string password, string outFolder)// витягує файли з архіву  private void CompressFolder(string path, ZipOutputStream zipStream, int folderOffset)// стискає файли |

Також я створив статичний класс який зберігає у собі інформацію яка потрібна при роботі з додатковими формами

|  |
| --- |
| public static class DataBank  public static DirectoryInfo CurrDir { get; set; }//поточна директорія  public static DirectoryInfo DestDir { get; set; }//кінцева директорія  public static List<string> SelectedFiles { get; set; } = new List<string>();//список вибраних файлів  public static void Clear()//метод для очистки обєкту |

## 2.3 Розробка програмного забезпечення

Основні реалізовані методи представлені в класі Form1.cs

Класс є основним у всій програмі оскільки там знаходится основна логіка роботи додатку.Також тут знаходится виклик додаткових форм

Представлено тільки назви методів та короткий опис

Клас **Form1.cs:**

|  |
| --- |
| private MyListView GetActiveList()//повертає активний список  private MyListView GetInActiveList()//повертає неактивний список  private void GetFilesAndDirs(MyListView listView)//отримуємо всі файли та директорії  private void listView\_MouseDoubleClick(object sender, MouseEventArgs e)//відкриває файл або робить перехід на вищий рівень якщо можна  private void comboBox\_SelectedValueChanged(object sender, EventArgs e)//оновлює лісти при зміні на інший диск  private void toolStripMenuCipher\_Click(object sender, EventArgs e)//шифрування  private void toolStripMenuDeСipher\_Click(object sender, EventArgs e)//дешифрування  private void toolStripAddToArchive\_Click(object sender, EventArgs e)//архівування  private void toolStripDeArchive\_Click(object sender, EventArgs e)//деархівування  private void listView\_DragEnter(object sender, DragEventArgs e)//метод на вхід курсору  private void listView\_DragDrop(object sender, DragEventArgs e)//метод на кидання файлів  private void listView\_ItemDrag(object sender, ItemDragEventArgs e)//метод на спробу перетягнути файл  private void toolStripButtonEdit\_Click(object sender, EventArgs e)//виконуемо файл  private void toolStripButtonMove\_Click(object sender, EventArgs e)//переміщаемо файл  private void toolStripButtonCopy\_Click(object sender, EventArgs e)//викликаемо форму копіювання  private void toolStripButtonDelete\_Click(object sender, EventArgs e)// викликаемо форму видалення  private void toolStripButtonRename\_Click(object sender, EventArgs e)//викликаемо форму перейменування |

Діаграма класів

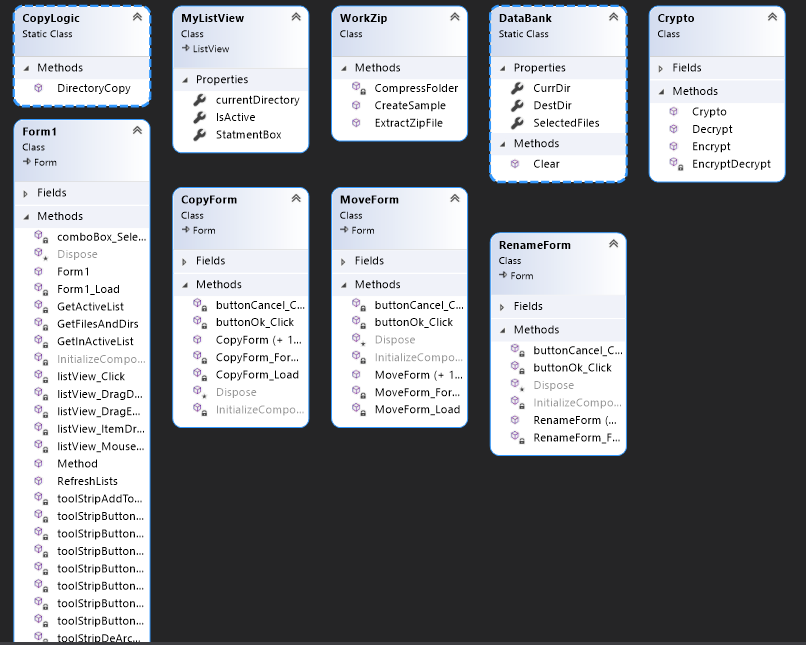


Рис. 2.4. Діаграма класів

## **Висновок до другого розділу**: в другому розділі я описав код основних методів, склав діаграму класів та прецедентів.Описав загальний алгоритм роботи програми та додаткових класів

# Розділ 3. Опис роботи з програмним додатком та його тестування.

# 3.1 Опис роботи з програмним додатком

Після запуску програми ми бачимо два вікна рисунок 3.1, на якому розміщені головні елементи управління.Інтуітивно зрозумілі кнопки говорять самі за себе.

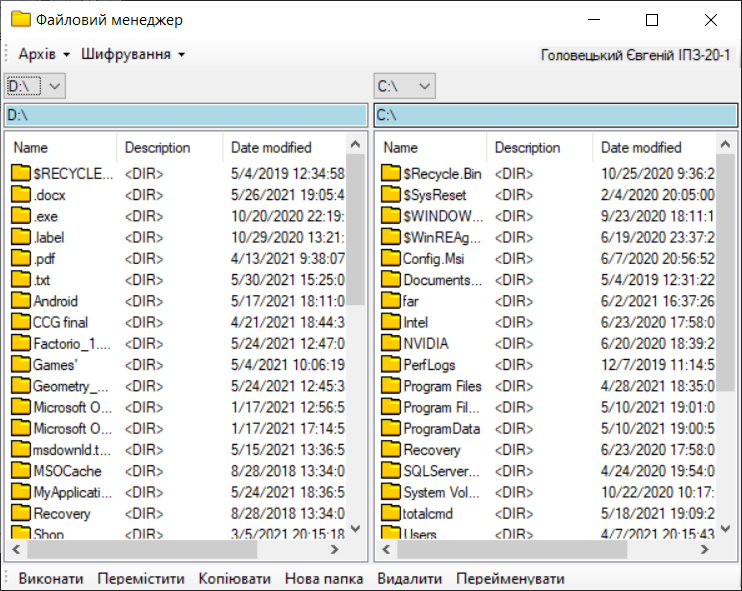


Рис. 3.1. Загальний вигляд програми

Робота з архівами

При натисканні на кнопку архів випадае меню з кнопками архівувати та деарзівуват.

Для коректного виконання потрібно спочатку вибрати файли в одному вікні та натиснути кновку в випадаючому меню додати до архіву далі зявиться форма для вказання назви нового архіву

Для деархіваціїї потрібно вибрати готовий архів і натиснути кнопку Роз архівувати

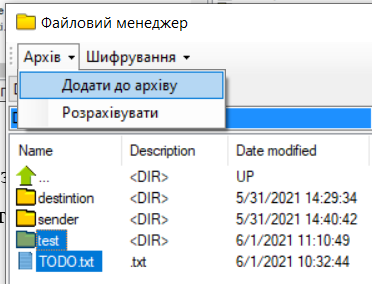


Рис. 3.2. Загальний вигляд меню «Архів»

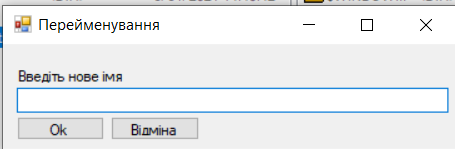


Рис. 3.3. Форма назви

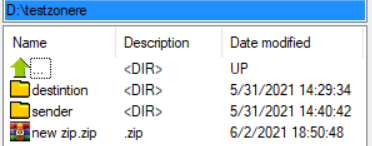


Рис. 3.4. Створений архів

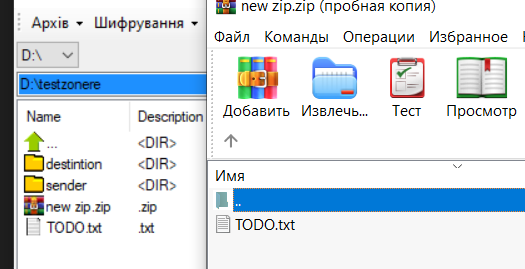


Рис. 3.5. Вміст архіву та демонстрація розархівування

Для шифрування потрібно зробити аналогічні дії що і архівом

Спочатку вибираемо файл і натискаємо шифрувати

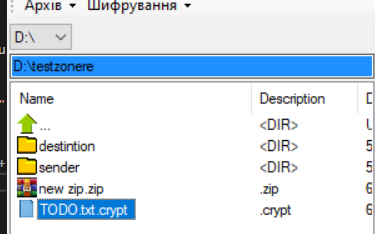


Рис. 3.6. Створений зашифрований файл

Далі розглянемо функціонал нижнього тулбару

Кнопка Виконати як не дивно виконує файл

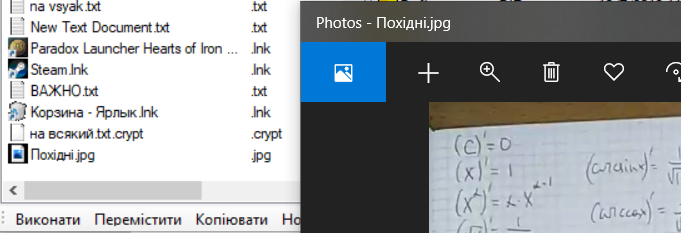


Рис. 3.7. Відкриття файлу Похідні

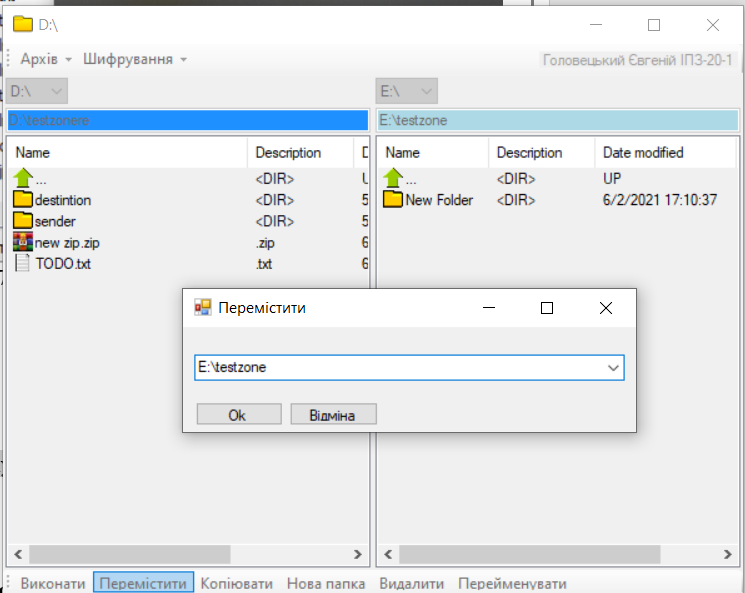
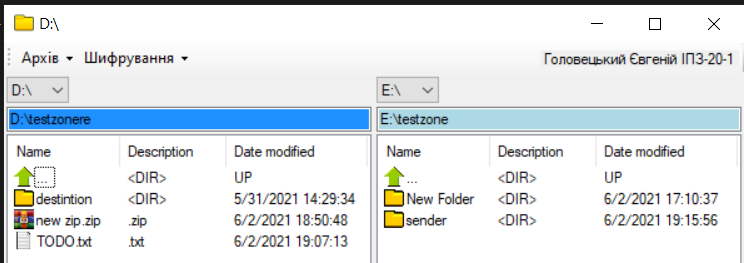


Рис. 3.8. Виклик форми переміщення та вибір папки з другого вікна

 Рис. 3.9. Папку sender було переміщено на диск Е

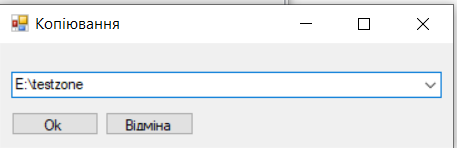


Рис. 3.10. Виклик форми копіювання та вибір папки з другого вікна

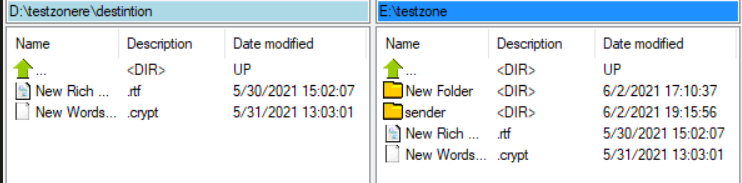


Рис. 3.11. Скопійовані файли

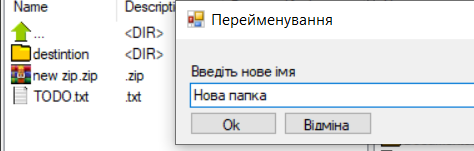


Рис. 3.12. Виклик форми переіменування та вибір введення імя

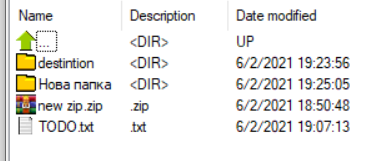


Рис. 3.13. Створена папка

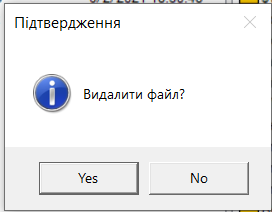


Рис. 3.14. Видалення створеної папки

Для виконання Drag&Drop порібно виділити потрібні файли та перетягнути їх в інше вікно

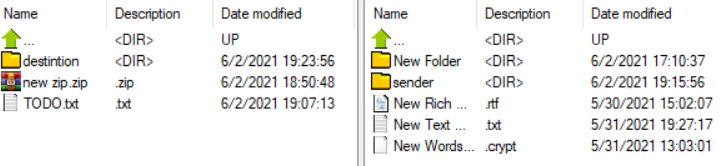


Рис. 3.15. Демонстрація Drag&Drop

# 3.2 Тестування роботи програмного забезпечення

Програма передбачає помилки при введенні користувачем, для зручного

користування програмою

При спробі натиснути на нижні кнопки без виділених файлів призведе до помилки

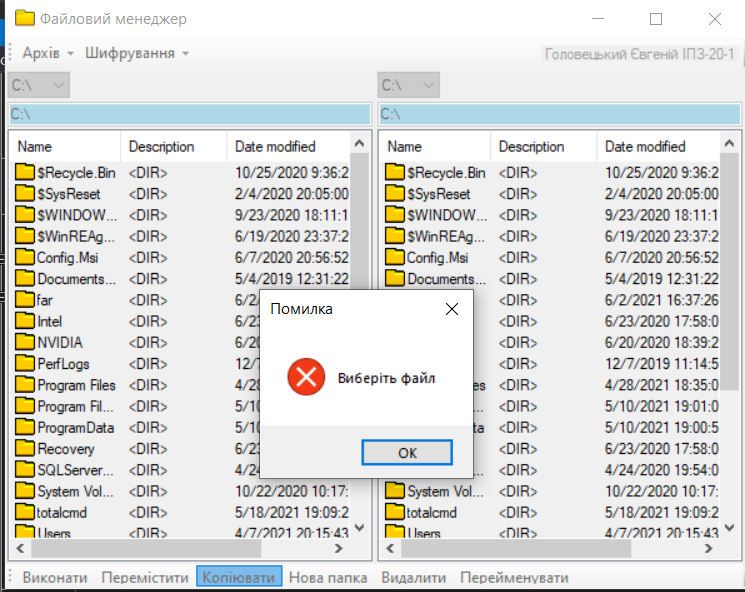


Рис. 3.16. Помилка при натисканні кнопку без виділених файлів

При спробі шифування папки також буде помилка про вибраний файл

При спробі розархівувати не zip файл також виникне помилка

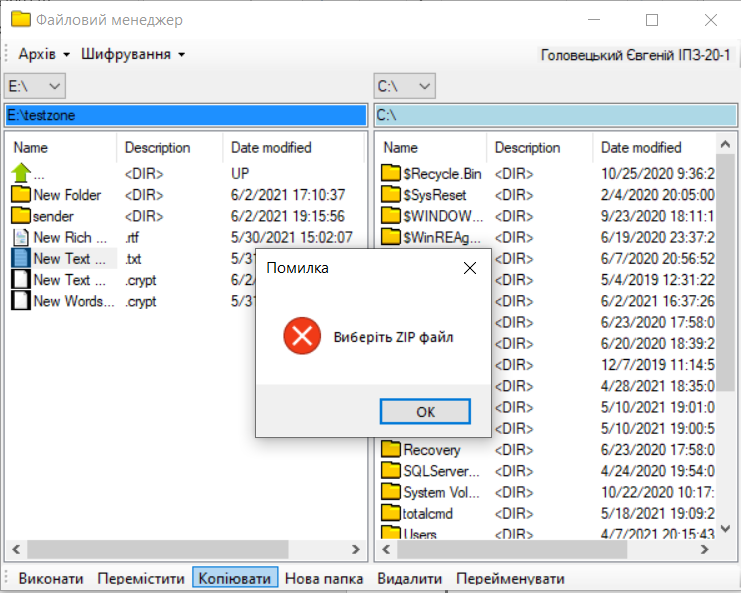


Рис. 3.17. Помилка при виділенню не zip файлу

**Висновок до третього розділу:**

В третьому розділі я показав основний функціонал програми, навів скріншоти, та короткий опис кроків

Також я описав декілька основних помилок я можуть виникнути при виконанні програми

# Висновки

## Список використаних джерел

1. Джеймс В. Ньюкирк, Роберт С. Косс, Роберт Мартин. Быстрая разработка программ. Принципы, примеры, практика/ Джеймс В. Ньюкирк, Роберт С. Косс, Роберт Мартин. - Вильямс, 2004 – 752 ст.
2. Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
3. Рихтер Дж. Р55 CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013. — 896 с.: ил. —
4. Move a folder from one Drive to another in c# [closed] [Електронний ресурс] Режим доступу до ресурсу: https://stackoverflow.com/questions/23927702/move-a-folder-from-one-drive-to-another-in-c-sharp
5. How to: Copy directories [Електронний ресурс] Режим доступу до ресурсу: https://docs.microsoft.com/en-us/dotnet/standard/io/how-to-copy-directories
6. Работа с файлами. Классы File и FileInfo [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/tutorial/5.3.php
7. admin. Добавление форм. Взаимодействие между формами [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/windowsforms/2.3.php

# Додатки

Додаток А

Form1.cs

|  |
| --- |
| using Microsoft.VisualBasic.FileIO;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Diagnostics;  using System.Drawing;  using System.IO;  using System.IO.Compression;  using System.Linq;  using System.Text;  using System.Text.RegularExpressions;  using System.Threading.Tasks;  using System.Windows.Forms;  using WindowsFormsApp1.Additional;  namespace WindowsFormsApp1  {  public partial class Form1 : Form  {  private ImageList imageList1;  private Crypto crypt = new Crypto("a");  private WorkZip workZip = new WorkZip();  public Form1()  {  InitializeComponent();  imageList1 = new ImageList();  listViewLeft.SmallImageList = imageList1;  listViewRight.SmallImageList = imageList1;  listViewLeft.StatmentBox = textBoxLeft;  listViewRight.StatmentBox = textBoxRight;  }  private MyListView GetActiveList()//повертає активний список  {  MyListView listView = null;  if (listViewLeft.IsActive == true)  listView = listViewLeft;  else if (listViewRight.IsActive == true)  listView = listViewRight;  return listView;  }  private MyListView GetInActiveList()//повертає неактивний список  {  MyListView listView = null;  if (listViewLeft.IsActive == false)  listView = listViewLeft;  else if (listViewRight.IsActive == false)  listView = listViewRight;  return listView;  }  private void GetFilesAndDirs(MyListView listView)//отримуємо всі файли та директорії  {  listView.BeginUpdate();  listView.Items.Clear();  try  {  ListViewItem item = new ListViewItem();  item.ImageKey = "PREVIOUSFOLDER";  item.SubItems.Add("<DIR>");  item.SubItems.Add("UP");  item.Text = "...";  if (listView.currentDirectory.Parent != null)  {  listView.Items.Add(item);  }  foreach (DirectoryInfo dir in listView.currentDirectory.GetDirectories())  {  item = new ListViewItem(dir.Name, 1);  item.ImageKey = "dir";  item.SubItems.Add("<DIR>");  item.SubItems.Add(dir.LastWriteTime.ToString());  listView.Items.Add(item);  }  foreach (FileInfo file in listView.currentDirectory.GetFiles())  {  Icon iconForFile = SystemIcons.WinLogo;  item = new ListViewItem(file.Name, 1);  if (!imageList1.Images.ContainsKey(file.Extension))  {  iconForFile = Icon.ExtractAssociatedIcon(file.FullName);  imageList1.Images.Add(file.FullName + file.Extension, iconForFile);  }  item.ImageKey = file.FullName + file.Extension;  item.SubItems.Add(file.Extension);  item.SubItems.Add(file.LastWriteTime.ToString());  item.SubItems.Add(file.Length.ToString());  listView.Items.Add(item);  }  }  catch (Exception) { }  listView.StatmentBox.Text = listView.currentDirectory.FullName;  listView.EndUpdate();  }  private void Form1\_Load(object sender, EventArgs e)  {  Icon dirIcon = new Icon("..\\..\\Assets\\folder.ico");  imageList1.Images.Add("dir", dirIcon);  dirIcon = new Icon("..\\..\\Assets\\up.ico");  imageList1.Images.Add("PREVIOUSFOLDER", dirIcon);  foreach (DriveInfo drive in DriveInfo.GetDrives())  {  comboBoxLeft.Items.Add(drive.Name);  comboBoxRight.Items.Add(drive.Name);  }  comboBoxLeft.SelectedIndex = 0;  comboBoxRight.SelectedIndex = 0;  }  private void listView\_MouseDoubleClick(object sender, MouseEventArgs e)//відкриває файл або робить перехід на вищий рівень якщо можна  {  MyListView listView = (MyListView)sender;  ListViewItem item = listView.SelectedItems[0];  if (item.ImageKey == "PREVIOUSFOLDER")  {  listView.currentDirectory = listView.currentDirectory.Parent;  this.Text = listView.currentDirectory.FullName;  GetFilesAndDirs(listView);  }  else if (new DirectoryInfo(listView.currentDirectory.FullName + "\\" + item.Text).Exists)  {  listView.currentDirectory = new DirectoryInfo(listView.currentDirectory.FullName + "\\" + item.Text);  GetFilesAndDirs(listView);  }  else  {  Process.Start(listView.currentDirectory.FullName + "\\" + item.Text);  }  }  private void comboBox\_SelectedValueChanged(object sender, EventArgs e)//оновлює лісти при зміні на інший диск  {  ComboBox comboBox = (ComboBox)sender;  if (comboBox == comboBoxLeft)  {  listViewLeft.Items.Clear();  string currDriver = comboBox.SelectedItem.ToString();  listViewLeft.currentDirectory = new DirectoryInfo(currDriver);  GetFilesAndDirs(listViewLeft);  }  else if (comboBox == comboBoxRight)  {  listViewRight.Items.Clear();  string currDriver = comboBox.SelectedItem.ToString();  listViewRight.currentDirectory = new DirectoryInfo(currDriver);  GetFilesAndDirs(listViewRight);  }  }  private void toolStripMenuCipher\_Click(object sender, EventArgs e)//шифрування  {  MyListView listView = GetActiveList();  ListViewItem item = new ListViewItem();  item = listViewLeft.SelectedItems[0];  crypt.Encrypt(listView.currentDirectory.FullName + "/" + item.Text, listView.currentDirectory.FullName + "/" + item.Text + ".crypt");  RefreshLists();  }//  private void toolStripMenuDeСipher\_Click(object sender, EventArgs e)//дешифрування  {  MyListView listView = GetActiveList();  ListViewItem item = new ListViewItem();  item = listViewLeft.SelectedItems[0];  if (item.Text.Contains(".crypt"))  {  crypt.Decrypt(listView.currentDirectory.FullName + "/" + item.Text, listView.currentDirectory.FullName + "/" + item.Text.Replace(".crypt", ""));  }  RefreshLists();  }  private void toolStripAddToArchive\_Click(object sender, EventArgs e)//архівування  {  MyListView listView = GetActiveList();  DirectoryInfo dir = Directory.CreateDirectory(listView.currentDirectory.FullName + "\\" + "TEMPDIRZIP");  foreach (ListViewItem item in listView.SelectedItems)  {  Directory.Move(listView.currentDirectory.FullName + "\\" + item.Text, dir.FullName + "\\" + item.Text);  }  workZip.CreateSample(listView.currentDirectory.FullName + "\\" + "NewZip.zip", null, dir.FullName);  if (dir.Exists)  {  dir.Delete(true);  }  this.Enabled = false;  DataBank.CurrDir = GetActiveList().currentDirectory;  DataBank.SelectedFiles.Add(GetActiveList().currentDirectory.FullName + "\\" + "NewZip.zip");  RenameForm renameForm = new RenameForm(this);  renameForm.Show();  }  private void toolStripDeArchive\_Click(object sender, EventArgs e)//деархівування  {  MyListView listView = GetActiveList();  ListViewItem item = new ListViewItem();  item = listView.SelectedItems[0];  if (item.Text.Contains(".zip"))  {  this.Text = listView.currentDirectory.FullName + item.Text;  workZip.ExtractZipFile(listView.currentDirectory.FullName + "\\" + item.Text, null, listView.currentDirectory.FullName);  }  else  {  MessageBox.Show("Виберіть ZIP файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  RefreshLists();  }  private void listView\_DragEnter(object sender, DragEventArgs e)//метод на вхід курсору  {  try  {  if (e.Data.GetDataPresent(typeof(string[])) || e.Data.GetDataPresent(DataFormats.FileDrop)) ;  e.Effect = DragDropEffects.Move;  }  catch (Exception)  {  throw;  }  }  private void listView\_DragDrop(object sender, DragEventArgs e)//метод на кидання файлів  {  MyListView listView = (MyListView)sender;  try  {  string[] data = (string[])e.Data.GetData(typeof(string[]));  if (e.Data.GetDataPresent(typeof(string[])))  {  data = (string[])e.Data.GetData(typeof(string[]));  }  else  {  data = (string[])e.Data.GetData(DataFormats.FileDrop);  }  foreach (string dat in data)  {  DirectoryInfo dir = new DirectoryInfo(dat);  if (dir.Exists)  {  FileSystem.MoveDirectory(dir.FullName, listView.currentDirectory.FullName + "\\" + dir.Name);  }  else  {  FileSystem.MoveFile(dir.FullName, listView.currentDirectory.FullName + "\\" + dir.Name);  }  }  }  catch (Exception)  {  }  RefreshLists();  }  public void RefreshLists()  {  GetFilesAndDirs(listViewLeft);  GetFilesAndDirs(listViewRight);  }  private void listView\_ItemDrag(object sender, ItemDragEventArgs e)//метод на спробу перетягнути файл  {  MyListView listView = (MyListView)sender;  try  {  List<string> data = new List<string>();  foreach (ListViewItem lvi in listView.SelectedItems)  {  data.Add(listView.currentDirectory.FullName + "\\" + lvi.Text);  }  string[] dat = data.ToArray();  listView.DoDragDrop(dat, DragDropEffects.Move);  }  catch (Exception)  {  }  }  private void listView\_Click(object sender, EventArgs e)  {  MyListView listView = (MyListView)sender;  try  {  if (listView == listViewLeft)  {  listViewRight.IsActive = false;  listViewRight.StatmentBox.BackColor = Color.LightBlue;  }  else  {  listViewLeft.IsActive = false;  listViewLeft.StatmentBox.BackColor = Color.LightBlue;  }  listView.IsActive = true;  listView.StatmentBox.BackColor = Color.DodgerBlue;  }  catch (Exception)  {  throw;  }  }//метод на клік по елементу списку  private void toolStripButtonEdit\_Click(object sender, EventArgs e)//виконуемо файл  {  try  {  MyListView listView = GetActiveList();  if (Directory.Exists(listView.currentDirectory.FullName + "\\" + listView.SelectedItems[0].Text))  {  MessageBox.Show("Виберіть файл");  }  else  {  Process.Start(listView.currentDirectory.FullName + "\\" + listView.SelectedItems[0].Text);  }  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  RefreshLists();  }  private void toolStripButtonMove\_Click(object sender, EventArgs e)//переміщаемо файл  {  try  {  this.Enabled = false;  DataBank.DestDir = GetInActiveList().currentDirectory;  DataBank.CurrDir = GetActiveList().currentDirectory;  foreach (ListViewItem item in GetActiveList().SelectedItems)  DataBank.SelectedFiles.Add(GetActiveList().currentDirectory.FullName + "\\" + item.Text);  MoveForm moveForm = new MoveForm(this);  moveForm.Show();  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  RefreshLists();  }  private void toolStripButtonCopy\_Click(object sender, EventArgs e)//викликаемо форму копіювання  {  try  {  this.Enabled = false;  DataBank.DestDir = GetInActiveList().currentDirectory;  DataBank.CurrDir = GetActiveList().currentDirectory;  foreach (ListViewItem item in GetActiveList().SelectedItems)  DataBank.SelectedFiles.Add(GetActiveList().currentDirectory.FullName + "\\" + item.Text);  CopyForm copyForm = new CopyForm(this);  copyForm.Show();  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  RefreshLists();  }  private void toolStripButtonDelete\_Click(object sender, EventArgs e)//видалення  {  try  {  DialogResult result = MessageBox.Show(  "Видалити файл?",  "Підтвердження ",  MessageBoxButtons.YesNo,  MessageBoxIcon.Information,  MessageBoxDefaultButton.Button1,  MessageBoxOptions.DefaultDesktopOnly);  if (result == DialogResult.Yes)  {  foreach (ListViewItem item in GetActiveList().SelectedItems)  {  if (Directory.Exists(GetActiveList().currentDirectory.FullName + "\\" + item.Text))  FileSystem.DeleteDirectory(GetActiveList().currentDirectory.FullName + "\\" + item.Text, DeleteDirectoryOption.DeleteAllContents);  else  File.Delete(GetActiveList().currentDirectory.FullName + "\\" + item.Text);  }  }  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  RefreshLists();  }  private void toolStripButtonFolder\_Click(object sender, EventArgs e)//створюемо папку  {  try  {  if (GetActiveList() == null)  {  MessageBox.Show("Виберіть вікно", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  Directory.CreateDirectory(GetActiveList().currentDirectory.FullName + "\\" + "New Folder");  DataBank.CurrDir = GetActiveList().currentDirectory;  DataBank.SelectedFiles.Add(GetActiveList().currentDirectory.FullName + "\\" + "New Folder");  RenameForm renameForm = new RenameForm(this);  renameForm.Show();  RefreshLists();  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true;  }  }  private void toolStripButtonRename\_Click(object sender, EventArgs e)//викликаемо форму перейменування  {  try  {  this.Enabled = false;  DataBank.CurrDir = GetActiveList().currentDirectory;  DataBank.SelectedFiles.Add(GetActiveList().currentDirectory.FullName + "\\" + GetActiveList().SelectedItems[0].Text);  RenameForm renameForm = new RenameForm(this);  renameForm.Show();  }  catch (Exception)  {  MessageBox.Show("Виберіть файл", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);  this.Enabled = true; RefreshLists();  }  }  }  } |

CopyForm.cs

|  |
| --- |
| using Microsoft.VisualBasic.FileIO;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using WindowsFormsApp1.Additional;  namespace WindowsFormsApp1  {  public partial class CopyForm : Form  {  public CopyForm()  {  InitializeComponent();  }  Form1 F;  public CopyForm(Form1 f)  {    InitializeComponent();  F = f;  }  private void buttonOk\_Click(object sender, EventArgs e)  {    foreach (string item in DataBank.SelectedFiles)  {  if (new DirectoryInfo(item).Exists)  FileSystem.CopyDirectory(item, DataBank.DestDir.FullName,true);  else  FileSystem.CopyFile(item, DataBank.DestDir.FullName + "\\" + (new DirectoryInfo(item).Name));    }  this.Close();  }  private void buttonCancel\_Click(object sender, EventArgs e)  {  this.Close();  }  private void CopyForm\_Load(object sender, EventArgs e)  {  comboBox1.Items.Add(DataBank.DestDir.FullName);  foreach (var dir in DataBank.DestDir.GetDirectories())  {  comboBox1.Items.Add(dir.FullName);  }  }    private void CopyForm\_FormClosed(object sender, FormClosedEventArgs e)  {  F.Enabled = true;    }  }  } |

RenameForm.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using WindowsFormsApp1.Additional;  namespace WindowsFormsApp1  {  public partial class RenameForm : Form  {  public RenameForm()  {  InitializeComponent();  }  Form1 F;  public RenameForm(Form1 f)  {  InitializeComponent();  F = f;  }  private void buttonOk\_Click(object sender, EventArgs e)  {      if (Directory.Exists(DataBank.SelectedFiles[0]))  {  new DirectoryInfo(DataBank.SelectedFiles[0]).MoveTo(DataBank.CurrDir + "\\" + textBox1.Text);  }else  {  string ext = new FileInfo(DataBank.SelectedFiles[0]).Extension;  File.Move(DataBank.SelectedFiles[0], DataBank.CurrDir + "\\"+textBox1.Text+ext);  }    this.Close();  }  private void buttonCancel\_Click(object sender, EventArgs e)  {  this.Close();  }  private void RenameForm\_FormClosed(object sender, FormClosedEventArgs e)  {  F.Enabled = true;  F.RefreshLists();  }  }  } |

MoveForm.cs

|  |
| --- |
| using Microsoft.VisualBasic.FileIO;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using WindowsFormsApp1.Additional;  namespace WindowsFormsApp1  {  public partial class MoveForm : Form  {  public MoveForm()  {  InitializeComponent();  }  Form1 F;  public MoveForm(Form1 f)  {  InitializeComponent();  F = f;  }  private void MoveForm\_Load(object sender, EventArgs e)  {  comboBox1.Items.Add(DataBank.DestDir.FullName);  foreach (var dir in DataBank.DestDir.GetDirectories())  {  comboBox1.Items.Add(dir.FullName);  }  }  private void buttonOk\_Click(object sender, EventArgs e)  {  foreach (string item in DataBank.SelectedFiles )  {    if (Directory.Exists(new DirectoryInfo(item).FullName))  {  FileSystem.MoveDirectory(new DirectoryInfo(item).FullName, comboBox1.SelectedItem.ToString() + "\\" + new DirectoryInfo(item).Name);  }  else  {  FileSystem.MoveFile(new DirectoryInfo(item).FullName, comboBox1.SelectedItem.ToString() + "\\" + new DirectoryInfo(item).Name);  }  }    this.Close();  }  private void buttonCancel\_Click(object sender, EventArgs e)  {  DataBank.Clear();  this.Close();  }  private void MoveForm\_FormClosed(object sender, FormClosedEventArgs e)  {  F.RefreshLists();  F.Enabled = true;  }  }  } |

Crypto.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace WindowsFormsApp1.Additional  {  class Crypto  {  private readonly int[] \_key;  private readonly int[] \_inversedKey;  public Crypto(string key)  {  var indexPairs = key  .Select((chr, idx1) => new { chr, idx1 })  .OrderBy(arg => arg.chr)  .Select((arg, idx2) =>  new  {  arg.idx1,  idx2  })  .ToArray();  \_key = indexPairs  .OrderBy(arg => arg.idx1)  .Select(arg => arg.idx2)  .ToArray();  \_inversedKey = indexPairs  .OrderBy(arg => arg.idx2)  .Select(arg => arg.idx1)  .ToArray();  }  public void Encrypt(string sourceFile, string destinationFile)  {  EncryptDecrypt(sourceFile, destinationFile, \_key);  }  public void Decrypt(string sourceFile, string destinationFile)  {  EncryptDecrypt(sourceFile, destinationFile, \_inversedKey);  }  private static void EncryptDecrypt(string sourceFile, string destinationFile, int[] key)  {  var keyLength = key.Length;  var buffer1 = new byte[keyLength];  var buffer2 = new byte[keyLength];  using (var source = new FileStream(sourceFile, FileMode.Open))  using (var destination = new FileStream(destinationFile, FileMode.OpenOrCreate))  {  while (true)  {  var read = source.Read(buffer1, 0, keyLength);  if (read == 0)  {  return;  }  else if (read < keyLength)  {  for (int i = read; i < keyLength; i++)  {  buffer1[i] = (byte)' ';  }  }  for (var i = 0; i < keyLength; i++)  {  var idx = i / keyLength \* keyLength + key[i % keyLength];  buffer2[idx] = buffer1[i];  }  destination.Write(buffer2, 0, keyLength);  }  }  }//виконує битове шифрування файлів за значенням  }  } |

WorkZip.cs

|  |
| --- |
| using ICSharpCode.SharpZipLib.Core;  using ICSharpCode.SharpZipLib.Zip;  using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace WindowsFormsApp1.Additional  {  class WorkZip  {  public void CreateSample(string outPathname, string password, string folderName) // створюе архів  {  using (FileStream fsOut = File.Create(outPathname))  using (var zipStream = new ZipOutputStream(fsOut))  {  zipStream.SetLevel(3);  zipStream.Password = password;  int folderOffset = folderName.Length + (folderName.EndsWith("\\") ? 0 : 1);  CompressFolder(folderName, zipStream, folderOffset);  }  }  public void ExtractZipFile(string archivePath, string password, string outFolder)// витягує файли з архіву  {  using (var fsInput = File.OpenRead(archivePath))  using (var zf = new ZipFile(fsInput))  {  if (!String.IsNullOrEmpty(password))  {  zf.Password = password;  }  foreach (ZipEntry zipEntry in zf)  {  if (!zipEntry.IsFile)  {  continue;  }  String entryFileName = zipEntry.Name;  var fullZipToPath = Path.Combine(outFolder, entryFileName);  var directoryName = Path.GetDirectoryName(fullZipToPath);  if (directoryName.Length > 0)  {  Directory.CreateDirectory(directoryName);  }  var buffer = new byte[4096];  using (var zipStream = zf.GetInputStream(zipEntry))  using (Stream fsOutput = File.Create(fullZipToPath))  {  StreamUtils.Copy(zipStream, fsOutput, buffer);  }  }  }  }  private void CompressFolder(string path, ZipOutputStream zipStream, int folderOffset)// стискає файли  {  var files = Directory.GetFiles(path);  foreach (var filename in files)  {  var fi = new FileInfo(filename);  var entryName = filename.Substring(folderOffset);  entryName = ZipEntry.CleanName(entryName);  var newEntry = new ZipEntry(entryName);  newEntry.DateTime = fi.LastWriteTime;  newEntry.Size = fi.Length;  zipStream.PutNextEntry(newEntry);  var buffer = new byte[4096];  using (FileStream fsInput = File.OpenRead(filename))  {  StreamUtils.Copy(fsInput, zipStream, buffer);  }  zipStream.CloseEntry();  }  var folders = Directory.GetDirectories(path);  foreach (var folder in folders)  {  CompressFolder(folder, zipStream, folderOffset);  }  }  }  } |

DataBank.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Diagnostics;  using System.Drawing;  using System.IO;  using System.IO.Compression;  using System.Linq;  using System.Text;  using System.Text.RegularExpressions;  using System.Threading.Tasks;  using System.Windows.Forms;  using WindowsFormsApp1.Additional;  namespace WindowsFormsApp1.Additional  {  public static class DataBank  {  public static DirectoryInfo CurrDir { get; set; }//поточна директорія  public static DirectoryInfo DestDir { get; set; }//кінцева директорія  public static List<string> SelectedFiles { get; set; } = new List<string>();//список вибраних файлів  public static void Clear()//метод для очистки обєкту  {  CurrDir = null;  DestDir = null;  SelectedFiles = null;  }  }  } |