Міністерство освіти і науки, молоді та спорту України  
Національний технічний університет України

"Київський політехнічний інститут"

Кафедра обчислювальної техніки

# **ЗАВДАННЯ НА КУРСОВУ РОБОТУ**

з курсу

**"СИСТЕМНЕ ПРОГРАМУВАННЯ"**

Завдання видав Завдання прйняв

Пустоваров В.И. Глухенько К.А.

група № ІО-93

Київ – 2011 р.

Зміст

Вступ3

Блок-схеми алгоритмів виконання операцій з поясненням4

Інтерфейс програми8

Інструкція до програмного проекту8

Перевірка програмного проекту10

Контрольні приклади14

Висновки17

Додаток18

**Вступ**

Основною ідеєю програмного проекту є наведення ладу в теках або в файловій системі цілком. Тобто програмний проект допоможе налагодити розміщення тек або файлів в файловій системі за якоюсь ознакою (вид, дата, тематика), а саме відкрити в одному з трьох вікон теку з якої будуть переміщуватися файли. Створити (за потреби) теку в яку будуть переміщені файли або теки в двох інших вікнах та за допомогою графічного інтерфейсу виконати потрібну дію.

Файловий менеджер — комп'ютерна [програма](/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), що надає [інтерфейс](/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [користувача](/wiki/%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87) для роботи з [файловою системою](/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) та [файлами](/wiki/%D0%A4%D0%B0%D0%B9%D0%BB).

Він дозволяє виконувати найчастіші операції з файлами: створення, відкриття/програвання/запуск/перегляд, редагування, переміщення, перейменування, копіювання, вилучення, зміну [атрибутів](/wiki/%D0%90%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82) та властивостей, пошук файлів та призначення прав.

Існує два види файлових менеджерів — навігаційні та ортодоксальні. Основна їх відмінність одне від одного — у ортодоксальних є дві панелі, реалізовано відповідну модель роботи.

Розроблений файловий менеджер дозволяє зручно виконувати сортування файлів у теках. Наприклад, якщо потрібно з однієї теки файли вибірково переслати у дві інші теки, то можна зробити це в одному вікні програми.

Було обрано мову C#, оскільки вона розроблена корпорацією Microsoftспеціально для використання у операційних системах сімейства Windows для CLR - Common Language Runtime — «загального середовища виконання мов» — компоненту пакету [Microsoft](/wiki/Microsoft)[.NET Framework](/wiki/.NET_Framework), віртуальної машини, на якій виконуються всі [мови](/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) платформи .NET Framework. Тому ця мова має можливості для коректного виконання програм на Windows.

**Блок-схеми алгоритмів виконання операцій з поясненням**

Операція копіювання



Операція копіювання виконує копіювання вибраного елементу одного з вікон в інше вікно, тобто з одного місця файлової системи в інше. Спочатку виконується перевірка чи є вибраний елементи для копіювання. Далі виконується розпізнавання що є вибраним елементом, тека чи файл. Після чого виконується перевірка чи існує елемент з таким само ім’ям, якщо існує то виводиться діалогове вікно з запитом на заміну. Якщо користувач вибрав замінити то виконується копіювання, з відображення відповідного прогрес бару. Після завершення операції виконується вихід із обробника події копіювання.

Операція переміщення



Операція переміщення виконує переміщення вибраного елементу одного з вікон в інше вікно, тобто з одного місця файлової системи в інше. Спочатку виконується перевірка чи є вибраний елементи для переміщення. Далі виконується розпізнавання що є вибраним елементом, тека чи файл. Після чого виконується перевірка чи існує елемент з таким само ім’ям, якщо існує то виводиться діалогове вікно з запитом на заміну. Якщо користувач вибрав замінити то виконується переміщення, з відображення відповідного прогрес бару. Після завершення операції виконується вихід із обробника події переміщення.

Операція видалення

Операція видалення виконує видалення вибраного елементу з файлової системи.

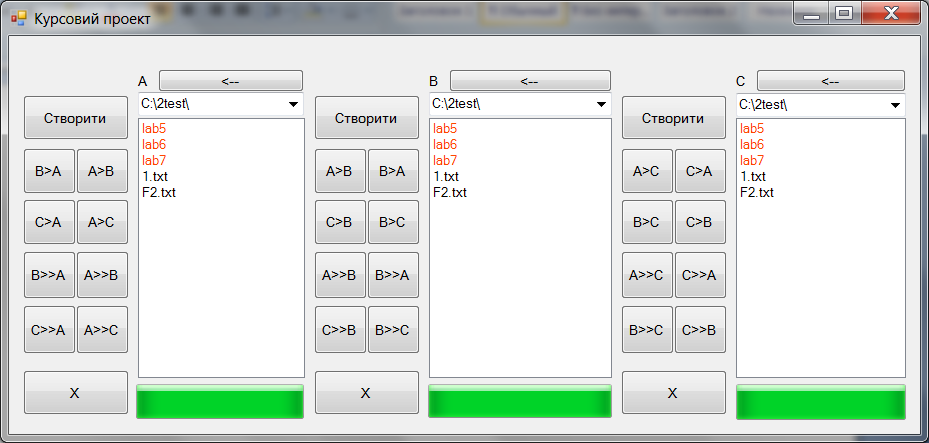
Операція видалення виконує видалення вибраного елементу з файлової системи. Спочатку виконується перевірка чи є вибраний елементи для видалення. Далі виводиться діалогове вікно для підтвердження дії. Якщо користувач підтвердив видалення то виконується розпізнавання що є вибраним елементом, тека чи файл. Після чого виконується видалення, з відображення відповідного прогрес бару. Після завершення операції виконується вихід із обробника події видалення.

Операція створення



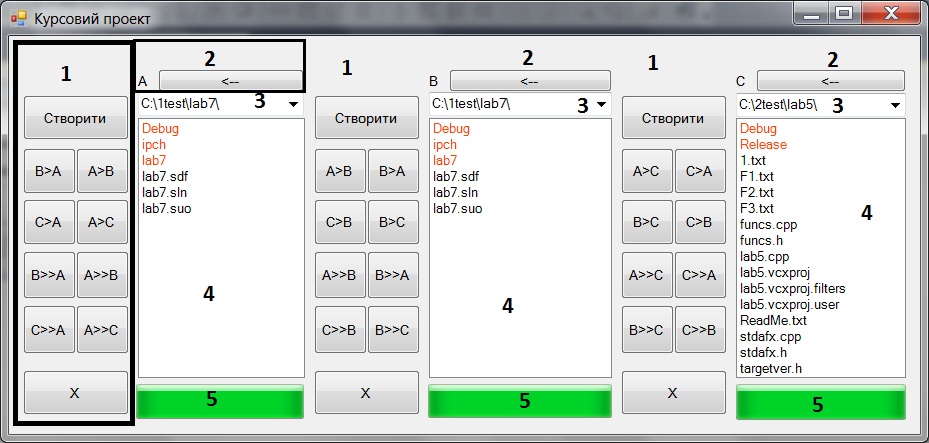
Операція створення виконує створення файлу або теки в файлової системи. Спочатку виводиться діалогове вікно яке дозволяє користувачу вибрати елемент який треба створити, ввести назву та за необхідності ввести розширення файлу. Якщо користувач підтвердив створення елементу, то виконується створення потрібного елементу.

**Інтерфейс програми**

****

**Інструкція до програмного проекту**

Для запуску програмного продукту треба запустити виконавчий файл в середовищі Windows. Ключів для запуску немає. Після запуску програми з’явиться графічний інтерфейс програми, який має наступний вигляд:



**Елементи з однаковими індексами не відрізняються одне від одного.**

1)Панель інструментів для роботи з теками і файлами

2) Місце на якому вказана назва панелі та кнопка “<--”

3) Місце для введення шляху файлової системи

4) Список файлів та тек

5) Прогрес бар, який вказує на етап виконання операції

**Навігація в файловій системі**

Навігація в файловій системі може виконуватися двома шляхами:

--- За допомогою миші, для цього треба навести курсор на потрібну теку (в місці 4) та зробити подвійний клік

--- За допомогою прямого введення адреси в полі 3, для цього треба ввести адресу і натиснути клавішу ENTER

Повернення вверх по дереву файлової системи треба:

--- Натиснути кнопку “<--”

--- Або натиснути клавішу “Backspace”

**Виконання операцій над файлами**

Створення Файлу/Теки

Для створення Файлу/Теки треба

1. Натиснути на кнопку Створити

2) Вибрати що створити файл або теку та ввести назву

3) Підтвердити

Копіювання Файлу/Теки

Для копіювання Файлу/Теки треба

1) Вибрати в панелях звідки та куди копіювати, а також виділити файл або теку для копіювання

2)Натиснути відповідну кнопку <Ім’я плитки (2) з якого копіювати> > < Ім’я плитки (2) в яку копіювати >

3)За потреби підтвердити заміну

Переміщення Файлу/Теки

Для переміщення Файлу/Теки треба

1) Вибрати в панелях звідки та куди перемітити, а також виділити файл або теку для переміщення

2)Натиснути відповідну кнопку <Ім’я плитки (2) з якого перемістити> > < Ім’я плитки (2) в яку перемістити >

3)За потреби підтвердити заміну

Видалення Файлу/Теки

1)Вибрати потрібний файл або теку

2)Натиснути на кнопку “Х”

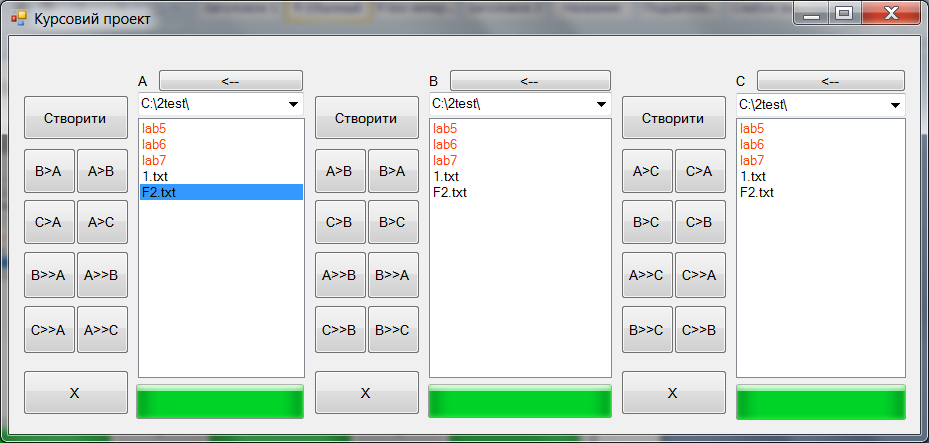
3) Підтвердити видалення

**Перевірка програмного проекту**

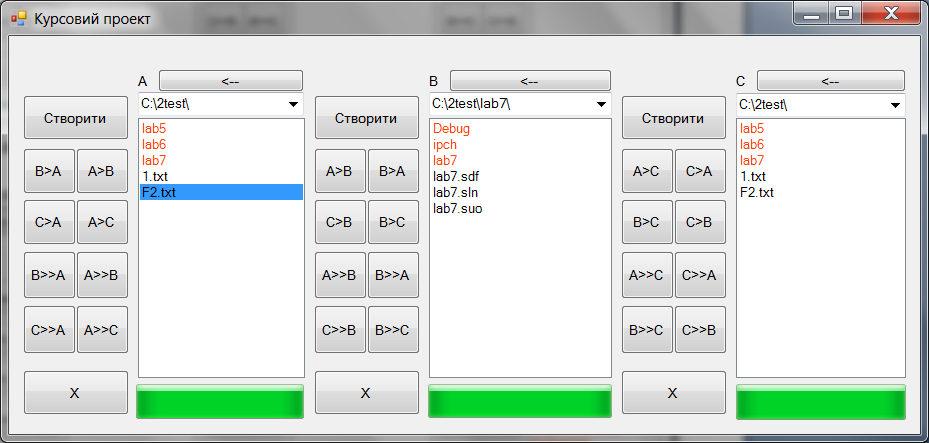
Для перевірки було виконано декілька тестів основних операцій. Критерієм успішного виконання операціє створення переміщення копіювання або видалення файлу/теки. Результати перевірялися за допомогою провідника Windows.

**Операція копіювання**

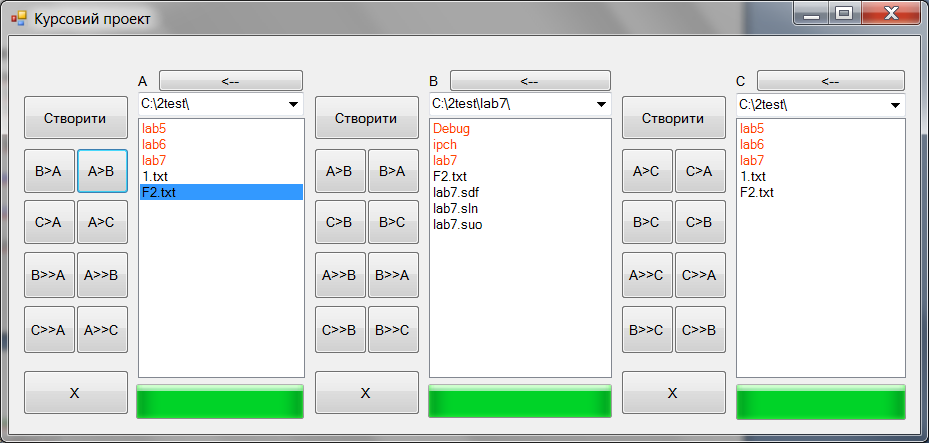
1. У бажаному віконці вибрано файл, який потрібно скопіювати.



2. Перейдено в іншому віконці до теки, в яку потрібно скопіювати обраний файл.

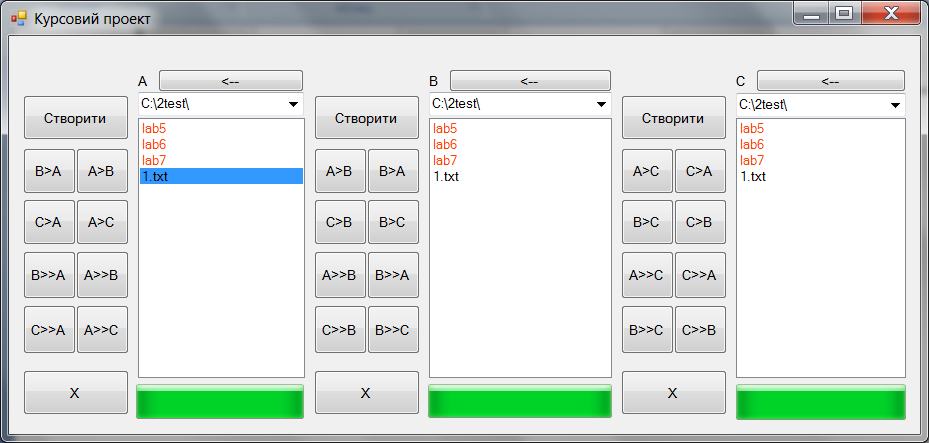


3. Для копіювання натиснуто A>B. Результат

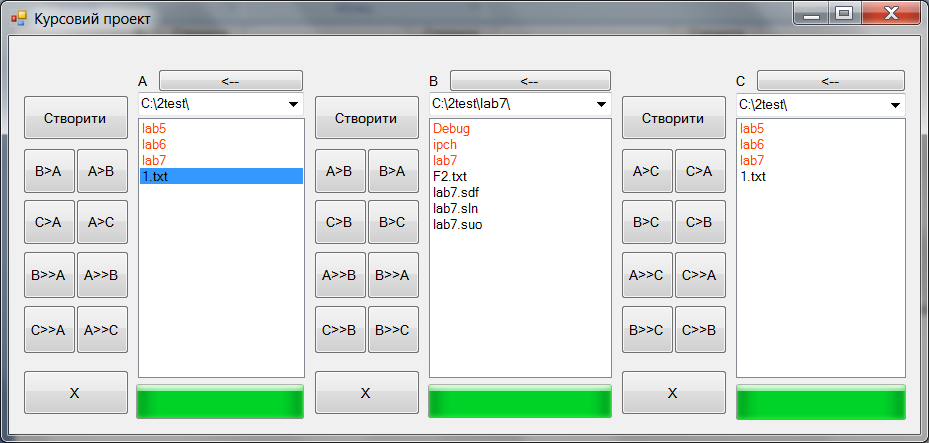


**Операція переміщення**

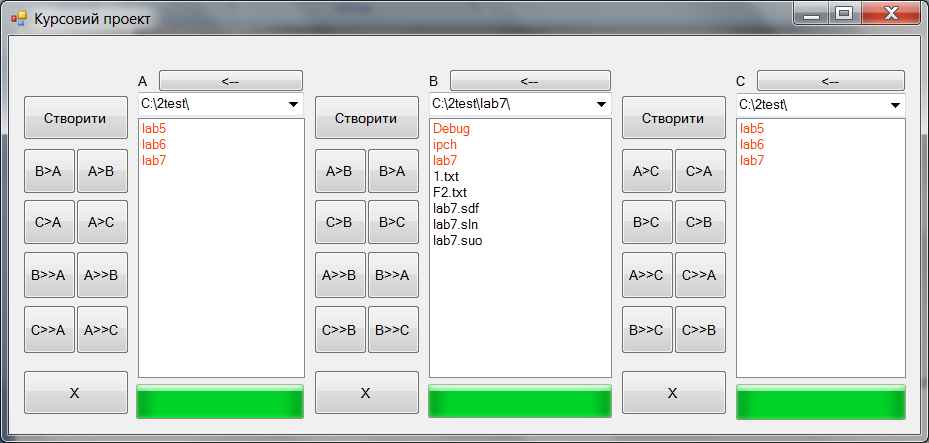
1. У бажаному віконці вибрано файл, який потрібно перемітити.



2. Перейдено в іншому віконці до теки, в яку потрібно скопіювати обраний файл.

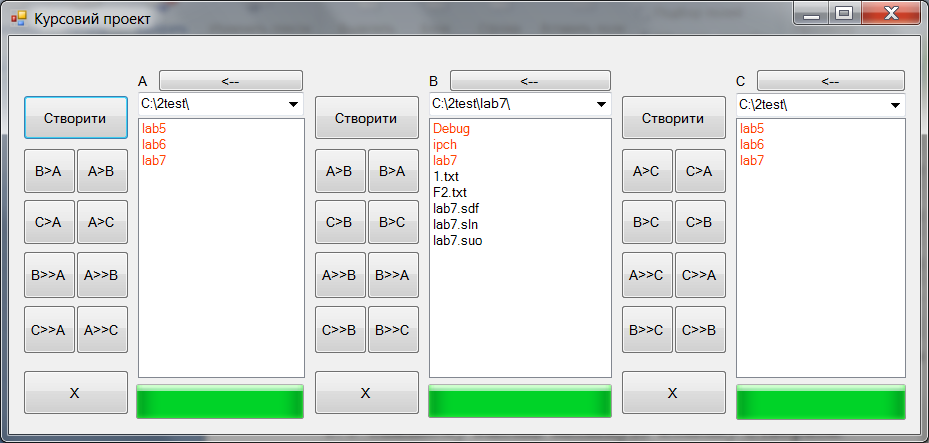


3. Для переміщення натиснуто A>>B. Результат

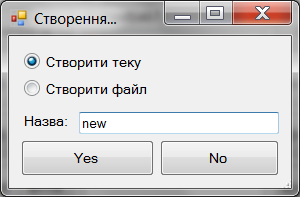


**Операція створення**

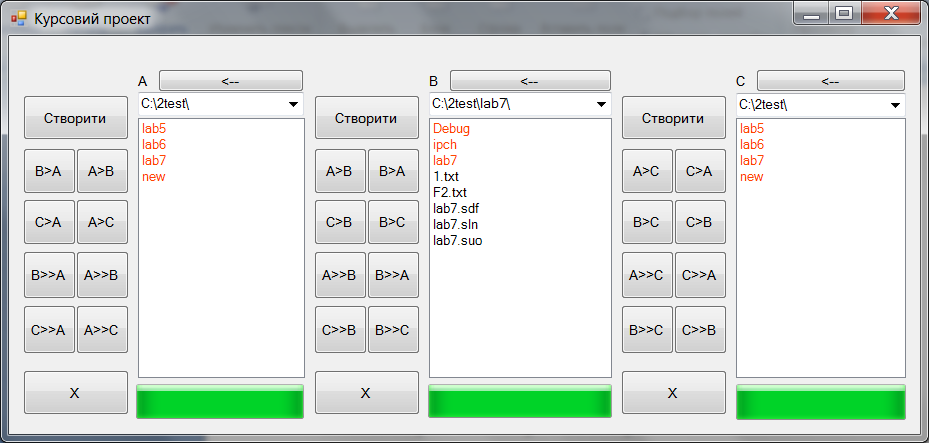
1. У бажаному віконці натиснуто клавішу Створити.



2. В діалоговому вікні введено назву нової теки

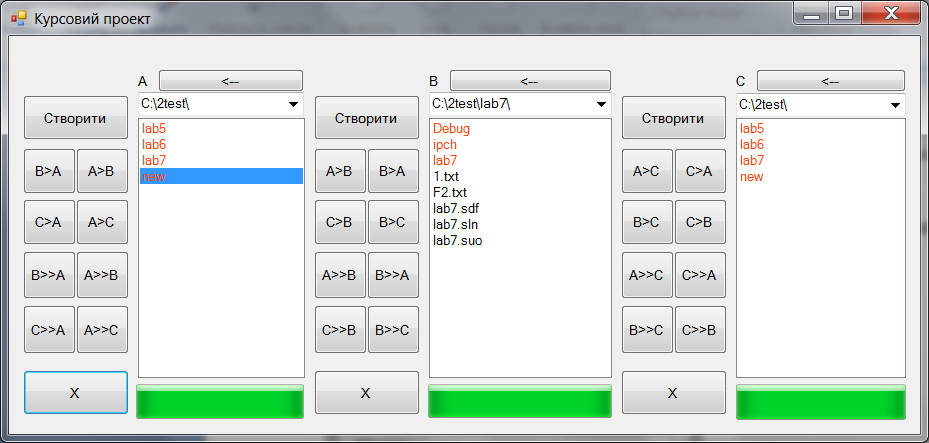


3. Натиснуто клавішу Yes. Результат

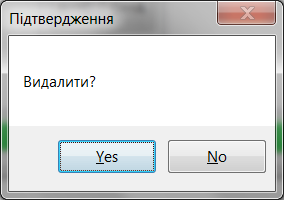


**Операція видалення**

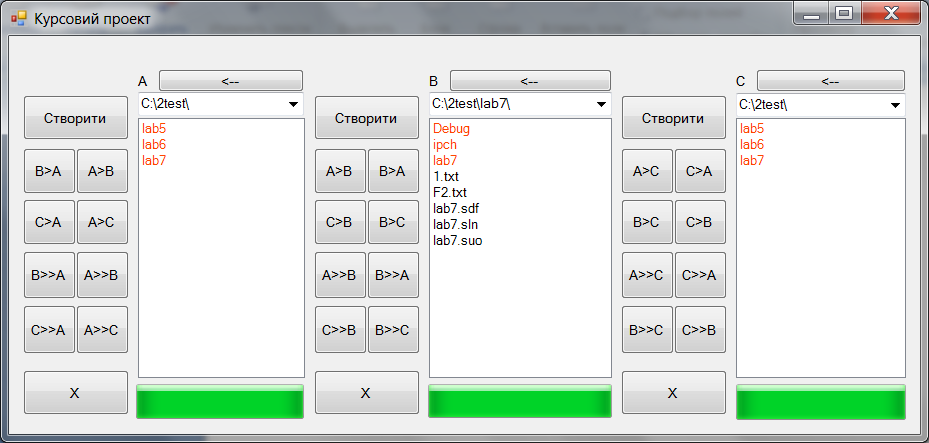
1. У бажаному віконці виділено потрібний елемент.



2. Натиснуто клавішу Х

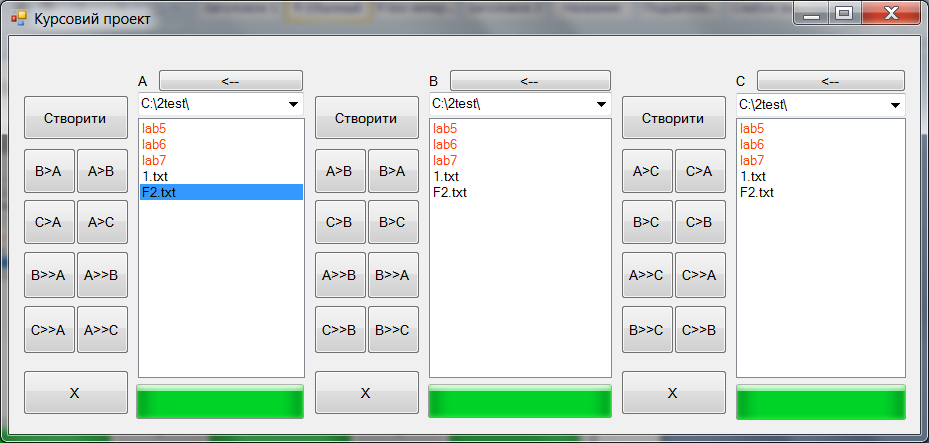


3. Натиснуто клавішу Yes. Результат

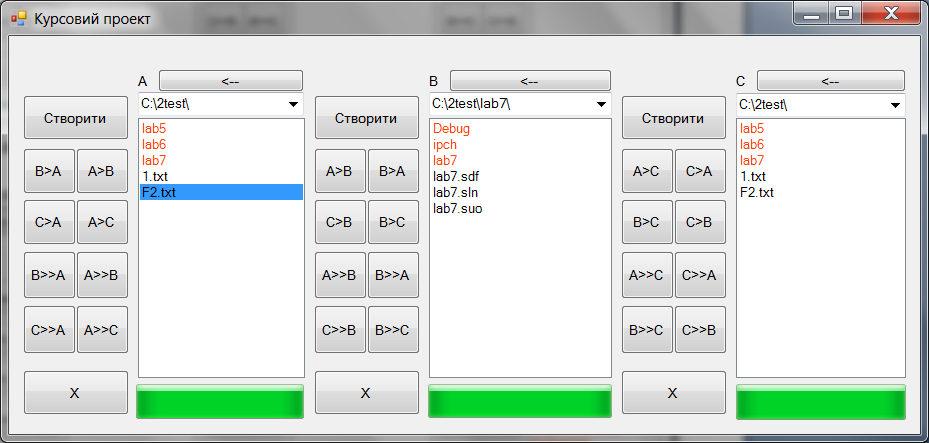


**Контрольні приклади**

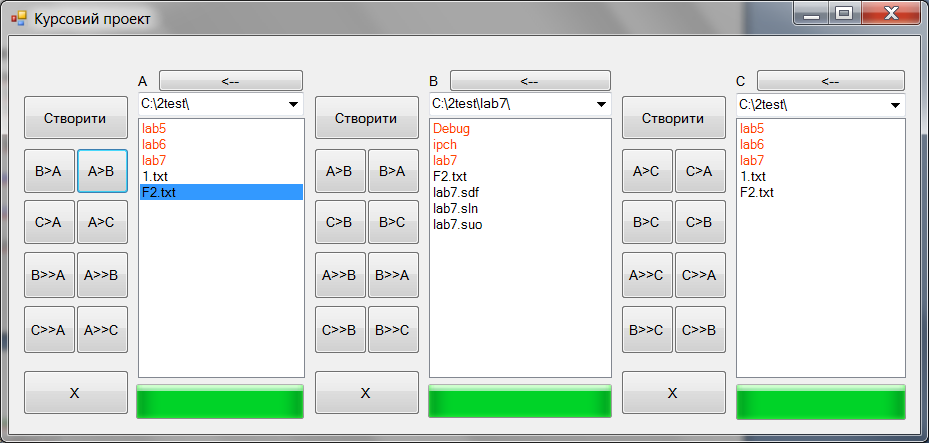
1. Потрібно у бажаному віконці вибрати файл, який потрібно перемістити або скопіювати.

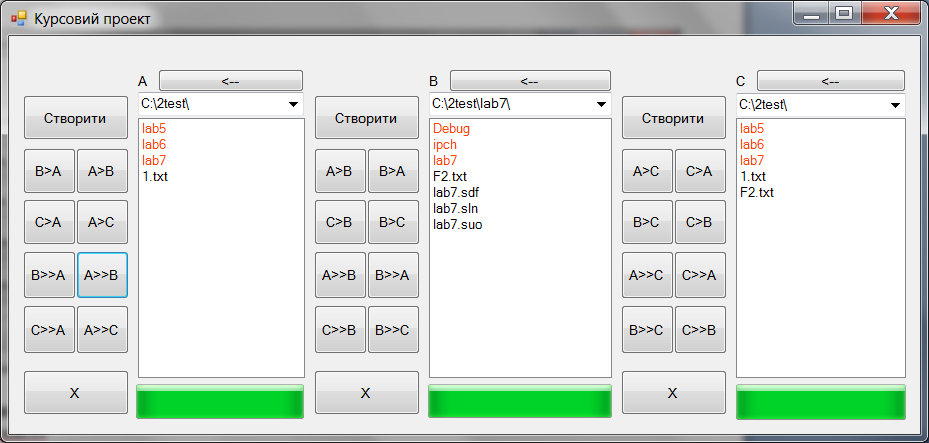


2. Перейти в іншому віконці до теки, в яку потрібно перемістити або скопіювати обраний файл.



3. Для копіювання або пересилки файлу натиснути A>Bабо A>>B відповідно.





**Висновки:**

При розробці даної програми було отримано навички в створюванні форм та динамічній зміні елементів GUI на мові С#, а також досліджено внутрішні команди для роботи з файлами та директоріями. Також я більш детально ознайомився з виключеннями у Windows. Ще я отримав навички в роботі з передачею за посиланням змінних.

**Додаток**

**Лістинг програми (C#)**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace SP

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**bodies.cs**

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;

namespace SP

{

class bodies

{

/\*\*

\* Метод збільшення прогресу виконання завдання

\*\*/

private static void IncProgress(ProgressBar pb, double k, ref double t)

{

t += k;

if (t > 1)

{

pb.Increment((int)t);

t -= (int)t;

}

}

/\*\*

\* Тіло обробника події зміни теки

\*\*/

public static void ComboBox\_SelectedIndexChanged(ComboBox comboBox, ListBox listBox, ref int last, ProgressBar progressBar)

{

listBox.DrawMode = DrawMode.Normal;

listBox.Items.Clear();

if (System.IO.Directory.Exists(comboBox.SelectedItem.ToString()))

{

progressBar.Value = 0;

String[] s = System.IO.Directory.GetDirectories(comboBox.SelectedItem.ToString());

string[] s1 = System.IO.Directory.GetFiles(comboBox.SelectedItem.ToString());

double k = (double)100 / (double)(s.Length + s1.Length - 2);

double t = 0;

for (int i = 0; i < s.Length; i++)

{

listBox.Items.Add(System.IO.Path.GetFileNameWithoutExtension(s[i]));

IncProgress(progressBar, k, ref t);

}

for (int i = 0; i < s1.Length; i++)

{

listBox.Items.Add(System.IO.Path.GetFileName(s1[i]));

IncProgress(progressBar, k, ref t);

}

last = comboBox.SelectedIndex;

progressBar.Value = 100;

}

else

listBox.Items.Add("Тека не знайдена");

listBox.DrawMode = DrawMode.OwnerDrawFixed;

}

/\*\*

\* Тіло обробника подій натиснення клавіш в елементах comboBox

\*\*/

public static void comboBox\_KeyPress(ComboBox comboBox, ListBox listBox, ref int last)

{

listBox.Items.Clear();

if (System.IO.Directory.Exists(comboBox.Text))

{

String[] s = System.IO.Directory.GetDirectories(comboBox.Text); ;

for (int i = 0; i < s.GetLength(0); i++)

listBox.Items.Add(System.IO.Path.GetFileNameWithoutExtension(s[i]));

s = System.IO.Directory.GetFiles(comboBox.Text); ;

for (int i = 0; i < s.GetLength(0); i++)

listBox.Items.Add(System.IO.Path.GetFileName(s[i]));

if (comboBox.Items[last].ToString() != comboBox.Text.ToString())

{

comboBox.Items.Insert(last + 1, comboBox.Text);

last++;

comboBox.SelectedIndex = last;

comboBox.Update();

}

}

else

listBox.Items.Add("Тека не знайдена");

}

/\*\*

\* Тіло обробника події подвійного натиснення на клавішу миші

\*\*/

public static void listBox\_MouseDoubleClick(ComboBox comboBox, ListBox listBox, ref int last)

{

if (System.IO.Directory.Exists(comboBox.SelectedItem.ToString() + listBox.SelectedItem.ToString()))

{

comboBox.Items.Insert(comboBox.SelectedIndex + 1, comboBox.SelectedItem.ToString() + listBox.SelectedItem.ToString() + "\\");

comboBox.SelectedIndex++;

}

else if (System.IO.File.Exists(comboBox.SelectedItem.ToString() + listBox.SelectedItem.ToString()))

{

System.Diagnostics.Process p = new System.Diagnostics.Process();

p.StartInfo = new System.Diagnostics.ProcessStartInfo(comboBox.Items[last].ToString() + listBox.SelectedItem.ToString());

p.Start();

}

}

/\*\*

\* Тіло обробника подій натиснення клавіш в елементах listBox

\*\*/

public static void listBox\_KeyPress(ComboBox comboBox, ListBox listBox, ref int last)

{

if (System.IO.Directory.GetDirectoryRoot(comboBox.Items[last].ToString()) != comboBox.Items[last].ToString())

{

last--;

comboBox.SelectedIndex = last;

comboBox.Update();

comboBox.Items.RemoveAt(last + 1);

}

}

/\*\*

\* Копіювання файлу

\*\*/

private static void CopyFile(String s1, String s2)

{

try

{

System.IO.File.Copy(s1, s2, replace);

}

catch (System.IO.IOException ex)

{

MessageBoxButtons buttons = MessageBoxButtons.YesNoCancel;

if (!yes)

{

MessageBoxButtons button = MessageBoxButtons.YesNo;

DialogResult res = MessageBox.Show("Замінити всі файли?", "Підтвердження", button);

switch (res)

{

case DialogResult.Yes:

{

replace = true;

System.IO.File.Copy(s1, s2, replace);

break;

}

case DialogResult.No:

{

yes = true;

try

{

System.IO.File.Copy(s1, s2, replace);

}

catch (System.IO.IOException ex1)

{

DialogResult res1 = MessageBox.Show("Замінити файл " + System.IO.Path.GetFileName(s1) + " ?", "Підтвердження", buttons);

switch (res1)

{

case DialogResult.Yes:

{

System.IO.File.Copy(s1, s2, true);

break;

}

case DialogResult.Cancel:

{

cansel = true;

break;

}

}

}

break;

}

}

}

else

{

if (!cansel)

{

DialogResult res = MessageBox.Show("Замінити файл " + System.IO.Path.GetFileName(s1) + " ?", "Підтвердження", buttons);

switch (res)

{

case DialogResult.Yes:

{

System.IO.File.Copy(s1, s2, true);

break;

}

case DialogResult.Cancel:

{

cansel = true;

break;

}

}

}

}

}

}

/\*\*

\* Переміщення файлу

\*\*/

private static void MoveFile(String s1, String s2)

{

try

{

if (replace)

System.IO.File.Delete(s2);

System.IO.File.Move(s1, s2);

}

catch (System.IO.IOException ex)

{

MessageBoxButtons buttons = MessageBoxButtons.YesNoCancel;

if (!yes)

{

MessageBoxButtons button = MessageBoxButtons.YesNo;

DialogResult res = MessageBox.Show("Замінити всі файли?", "Підтвердження", button);

switch (res)

{

case DialogResult.Yes:

{

replace = true;

System.IO.File.Delete(s2);

System.IO.File.Move(s1, s2);

break;

}

case DialogResult.No:

{

yes = true;

try

{

System.IO.File.Move(s1, s2);

}

catch (System.IO.IOException ex1)

{

DialogResult res1 = MessageBox.Show("Замінити файл " + System.IO.Path.GetFileName(s1) + " ?", "Підтвердження", buttons);

switch (res1)

{

case DialogResult.Yes:

{

System.IO.File.Delete(s2);

System.IO.File.Move(s1, s2);

break;

}

case DialogResult.Cancel:

{

cansel = true;

break;

}

}

}

break;

}

}

}

else

{

if (!cansel)

{

DialogResult res = MessageBox.Show("Замінити файл " + System.IO.Path.GetFileName(s1) + " ?", "Підтвердження", buttons);

switch (res)

{

case DialogResult.Yes:

{

System.IO.File.Delete(s2);

System.IO.File.Move(s1, s2);

break;

}

case DialogResult.Cancel:

{

cansel = true;

break;

}

}

}

}

}

}

/\*\*

\* Копіювання теки

\*\*/

private static void CopyDir(string FromDir, string ToDir)

{

if (!System.IO.Directory.Exists(ToDir))

System.IO.Directory.CreateDirectory(ToDir);

foreach (string s1 in System.IO.Directory.GetFiles(FromDir))

{

string s2 = ToDir + "\\" + System.IO.Path.GetFileName(s1);

CopyFile(s1, s2);

}

foreach (string s in System.IO.Directory.GetDirectories(FromDir))

{

if (ToDir[ToDir.Length - 1] == '\\')

CopyDir(s, ToDir + System.IO.Path.GetFileName(s));

else

CopyDir(s, ToDir + "\\" + System.IO.Path.GetFileName(s));

}

}

/\*\*

\* Переміщення теки

\*\*/

private static void MoveDir(string FromDir, string ToDir)

{

if (!System.IO.Directory.Exists(ToDir))

System.IO.Directory.CreateDirectory(ToDir);

foreach (string s1 in System.IO.Directory.GetFiles(FromDir))

{

string s2 = ToDir + "\\" + System.IO.Path.GetFileName(s1);

MoveFile(s1, s2);

}

foreach (string s in System.IO.Directory.GetDirectories(FromDir))

{

if (ToDir[ToDir.Length - 1] == '\\')

MoveDir(s, ToDir + System.IO.Path.GetFileName(s));

else

MoveDir(s, ToDir + "\\" + System.IO.Path.GetFileName(s));

}

if (!cansel)

System.IO.Directory.Delete(FromDir);

}

/\*\*

\* Обробник події копіювання

\*\*/

public static void Copy(ComboBox comboBoxI, ListBox listBoxI, ref int lastI, ComboBox comboBoxO, ListBox listBoxO, ref int lastO, ProgressBar progressBarO)

{

if (listBoxI.SelectedItems.Count != 0)

{

yes = false;

replace = false;

cansel = false;

progressBarO.Value = 0;

double k = (double)100 / (double)(listBoxI.SelectedItems.Count);

double z = 0;

for (int i = 0; i < listBoxI.SelectedItems.Count; i++)

{

if (System.IO.Directory.Exists(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString()))

{

if (!System.IO.Directory.Exists(comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString()))

System.IO.Directory.CreateDirectory(comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString());

CopyDir(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString(), comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString());

IncProgress(progressBarO, k, ref z);

}

else if (System.IO.File.Exists(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString()))

{

CopyFile(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString(), comboBoxO.Items[lastO].ToString() + "\\" + listBoxI.SelectedItems[i].ToString());

IncProgress(progressBarO, k, ref z);

}

}

}

}

/\*\*

\* Обробник події переміщення

\*\*/

public static void Move(ComboBox comboBoxI, ListBox listBoxI, ref int lastI, ComboBox comboBoxO, ListBox listBoxO, ref int lastO, ProgressBar progressBarO)

{

if (listBoxI.SelectedItems.Count != 0)

{

yes = false;

replace = false;

cansel = false;

progressBarO.Value = 0;

double k = (double)100 / (double)(listBoxI.SelectedItems.Count);

double z = 0;

for (int i = 0; i < listBoxI.SelectedItems.Count; i++)

{

if (System.IO.Directory.Exists(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString()))

{

if (!System.IO.Directory.Exists(comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString()))

System.IO.Directory.CreateDirectory(comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString());

MoveDir(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString(), comboBoxO.Items[lastO].ToString() + listBoxI.SelectedItems[i].ToString());

IncProgress(progressBarO, k, ref z);

}

else if (System.IO.File.Exists(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString()))

{

MoveFile(comboBoxI.Items[lastI].ToString() + listBoxI.SelectedItems[i].ToString(), comboBoxO.Items[lastO].ToString() + "\\" + listBoxI.SelectedItems[i].ToString());

IncProgress(progressBarO, k, ref z);

}

}

}

}

/\*\*

\* Обробник події видалення

\*\*/

public static void Delete(ComboBox comboBox, ListBox listBox, ref int last, ProgressBar progressBar)

{

if (listBox.SelectedItems.Count != 0)

{

DialogResult res = MessageBox.Show("Видалити?", "Підтвердження", MessageBoxButtons.YesNo);

switch (res)

{

case DialogResult.Yes:

{

progressBar.Value = 0;

double k = (double)100 / (double)(listBox.SelectedItems.Count);

double z = 0;

for (int i = 0; i < listBox.SelectedItems.Count; i++)

{

if (System.IO.Directory.Exists(comboBox.Items[last].ToString() + listBox.SelectedItems[i].ToString()))

{

System.IO.Directory.Delete(comboBox.Items[last].ToString() + listBox.SelectedItems[i].ToString(), true);

IncProgress(progressBar, k, ref z);

}

else if (System.IO.File.Exists(comboBox.Items[last].ToString() + listBox.SelectedItems[i].ToString()))

{

System.IO.File.Delete(comboBox.Items[last].ToString() + listBox.SelectedItems[i].ToString());

IncProgress(progressBar, k, ref z);

}

}

ComboBox\_SelectedIndexChanged(comboBox, listBox, ref last, progressBar);

break;

}

case DialogResult.No:

{

break;

}

}

}

}

/\*\*

\* Обробник події створення

\*\*/

public static void Create(ComboBox comboBox, ListBox listBox, ref int last, bool Dir, string name)

{

if (Dir)

{

if (!System.IO.Directory.Exists(comboBox.Items[last].ToString() + "\\" + name))

System.IO.Directory.CreateDirectory(comboBox.Items[last].ToString() + "\\" + name);

}

else

if (!System.IO.File.Exists(comboBox.Items[last].ToString() + "\\" + name))

System.IO.File.Create(comboBox.Items[last].ToString() + "\\" + name);

}

public static Boolean replace = false;

public static Boolean yes = false;

public static Boolean cansel = false;

}

}

**Form1.cs**

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;

namespace SP

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

String[] s = System.IO.Directory.GetLogicalDrives();

for (int i = 0; i < s.GetLength(0); i++)

{

comboBox1.Items.Add(s[i]);

comboBox2.Items.Add(s[i]);

comboBox3.Items.Add(s[i]);

}

comboBox1.SelectedIndex = 0;

comboBox1.Update();

comboBox2.SelectedIndex = 0;

comboBox2.Update();

comboBox3.SelectedIndex = 0;

comboBox3.Update();

}

/\*\*

\* Обробник події зміни теки

\*\*/

private void ComboBox1\_SelectedIndexChanged(object sender, System.EventArgs e)

{

ComboBox c = (ComboBox)sender;

switch (c.Name)

{

case "comboBox1":

{

bodies.ComboBox\_SelectedIndexChanged(c, listBox1, ref last1, progressBar1);

break;

}

case "comboBox2":

{

bodies.ComboBox\_SelectedIndexChanged(c, listBox2, ref last2, progressBar2);

break;

}

case "comboBox3":

{

bodies.ComboBox\_SelectedIndexChanged(c, listBox3, ref last3, progressBar3);

break;

}

}

}

/\*\*

\* Обробник подій натиснення клавіш в елементах comboBox

\*\*/

private void comboBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

ComboBox c = (ComboBox)sender;

switch (c.Name)

{

case "comboBox1":

{

if ((int)e.KeyChar == 13)

bodies.comboBox\_KeyPress(c, listBox1, ref last1);

break;

}

case "comboBox2":

{

if ((int)e.KeyChar == 13)

bodies.comboBox\_KeyPress(c, listBox2, ref last2);

break;

}

case "comboBox3":

{

if ((int)e.KeyChar == 13)

bodies.comboBox\_KeyPress(c, listBox3, ref last3);

break;

}

}

}

/\*\*

\* Обробник події подвійного натиснення на клавішу миші

\*\*/

private void listBox1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

ListBox l = (ListBox) sender;

switch (l.Name)

{

case "listBox1":

{

if (l.SelectedItem != null)

bodies.listBox\_MouseDoubleClick(comboBox1, l, ref last1);

break;

}

case "listBox2":

{

if (l.SelectedItem != null)

bodies.listBox\_MouseDoubleClick(comboBox2, l, ref last2);

break;

}

case "listBox3":

{

if (l.SelectedItem != null)

bodies.listBox\_MouseDoubleClick(comboBox3, l, ref last3);

break;

}

}

}

/\*\*

\* Обробник подій натиснення клавіш в елементах listBox

\*\*/

private void listBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

ListBox l = (ListBox)sender;

switch (l.Name)

{

case "listBox1":

{

if ((int)e.KeyChar == 8)

bodies.listBox\_KeyPress(comboBox1, l, ref last1);

break;

}

case "listBox2":

{

if ((int)e.KeyChar == 8)

bodies.listBox\_KeyPress(comboBox2, l, ref last2);

break;

}

case "listBox3":

{

if ((int)e.KeyChar == 8)

bodies.listBox\_KeyPress(comboBox3, l, ref last3);

break;

}

}

}

// B>A

private void button9\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox2, listBox2, ref last2, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

private int last1 = 0;

private int last2 = 0;

private int last3 = 0;

//A>B

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

{

bodies.Copy(comboBox1, listBox1, ref last1, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//C>A

private void button8\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox3, listBox3, ref last3, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

//A>C

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

{

bodies.Copy(comboBox1, listBox1, ref last1, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//A>B

private void button17\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox1, listBox1, ref last1, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

// B>A

private void button20\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox2, listBox2, ref last2, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

//C>B

private void button16\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox3, listBox3, ref last3, comboBox2, listBox2, ref last2, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//B>C

private void button19\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox2, listBox2, ref last2, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//A>C

private void button27\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox1, listBox1, ref last1, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//C>A

private void button30\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox3, listBox3, ref last3, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

//B>C

private void button26\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox2, listBox2, ref last2, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//C>B

private void button29\_Click(object sender, EventArgs e)

{

bodies.Copy(comboBox3, listBox3, ref last3, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//A>>B

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

{

bodies.Move(comboBox1, listBox1, ref last1, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

//B>>A

private void button7\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox2, listBox2, ref last2, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

private void button31\_Click(object sender, EventArgs e)

{

bodies.listBox\_KeyPress(comboBox1, listBox1, ref last1);

}

private void button32\_Click(object sender, EventArgs e)

{

bodies.listBox\_KeyPress(comboBox2, listBox2, ref last2);

}

private void button33\_Click(object sender, EventArgs e)

{

bodies.listBox\_KeyPress(comboBox3, listBox3, ref last3);

}

//C>>A

private void button6\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox3, listBox3, ref last3, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//A>>C

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

{

bodies.Move(comboBox1, listBox1, ref last1, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//A>>B

private void button14\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox1, listBox1, ref last1, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//B>>A

private void button18\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox2, listBox2, ref last2, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//C>>B

private void button12\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox3, listBox3, ref last3, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//B>>C

private void button15\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox2, listBox2, ref last2, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//A>>C

private void button24\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox1, listBox1, ref last1, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//C>>A

private void button28\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox3, listBox3, ref last3, comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

}

//B>>C

private void button22\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox2, listBox2, ref last2, comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

//C>>B

private void button25\_Click(object sender, EventArgs e)

{

bodies.Move(comboBox3, listBox3, ref last3, comboBox2, listBox2, ref last2, progressBar2);

bodies.ComboBox\_SelectedIndexChanged(comboBox3, listBox3, ref last3, progressBar3);

bodies.ComboBox\_SelectedIndexChanged(comboBox2, listBox2, ref last2, progressBar2);

}

private void button5\_Click(object sender, EventArgs e)

{

bodies.Delete(comboBox1, listBox1, ref last1, progressBar1);

}

private void button13\_Click(object sender, EventArgs e)

{

bodies.Delete(comboBox2, listBox2, ref last2, progressBar2);

}

private void button23\_Click(object sender, EventArgs e)

{

bodies.Delete(comboBox3, listBox3, ref last3, progressBar3);

}

private void button10\_Click(object sender, EventArgs e)

{

Form2 dialog = new Form2();

String name = "";

Boolean Dir = false;

Boolean no = false;

dialog.setControl(ref name, ref Dir, ref no);

if (!no)

{

bodies.Create(comboBox1, listBox1, ref last1, Dir, name);

}

bodies.ComboBox\_SelectedIndexChanged(comboBox1, listBox1, ref last1, progressBar1);

}

private void listBox1\_DrawItem(object sender, DrawItemEventArgs e)

{

e.DrawBackground();

Brush myBrush = Brushes.Black;

if (System.IO.Directory.Exists(comboBox1.Items[last1].ToString() + listBox1.Items[e.Index].ToString()))

myBrush = Brushes.OrangeRed;

e.Graphics.DrawString(listBox1.Items[e.Index].ToString(), e.Font, myBrush, e.Bounds, StringFormat.GenericDefault);

e.DrawFocusRectangle();

}

private void listBox2\_DrawItem(object sender, DrawItemEventArgs e)

{

e.DrawBackground();

Brush myBrush = Brushes.Black;

if (System.IO.Directory.Exists(comboBox2.Items[last2].ToString() + listBox2.Items[e.Index].ToString()))

myBrush = Brushes.OrangeRed;

e.Graphics.DrawString(listBox2.Items[e.Index].ToString(), e.Font, myBrush, e.Bounds, StringFormat.GenericDefault);

e.DrawFocusRectangle();

}

private void listBox3\_DrawItem(object sender, DrawItemEventArgs e)

{

e.DrawBackground();

Brush myBrush = Brushes.Black;

if (System.IO.Directory.Exists(comboBox3.Items[last3].ToString() + listBox3.Items[e.Index].ToString()))

myBrush = Brushes.OrangeRed;

e.Graphics.DrawString(listBox3.Items[e.Index].ToString(), e.Font, myBrush, e.Bounds, StringFormat.GenericDefault);

e.DrawFocusRectangle();

}

}

}

**Form2.cs**

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;

namespace SP

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public void setControl(ref String name1, ref Boolean Dir1, ref Boolean no1)

{

this.ShowDialog();

name1 = name;

Dir1 = Dir;

no1 = no;

this.Close();

}

private String name;

private Boolean Dir;

private Boolean no;

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

{

no = true;

this.Hide();

}

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

{

name = textBox1.Text;

Dir = radioButton1.Checked;

this.Hide();

}

}

}