**Лабораторна робота № 7**

Тема: Робота з файлами

Мета: Дослідити файлове введення/виведення.

**Теоретичні відомості**

System.IO містить всі необхідні класи, методи і властивості для маніпуляцій з каталогами та файлами (У Таблиці 1 наведено основні класи).

Таблиця 7.1 – Класи в System.IO

|  |  |
| --- | --- |
| **Клас** | **Призначення** |
| BinaryReader та Writer | Читання і запис простих типів даних |
| Directory, File, DirectoryInfo та FileInfo | Створення, видалення та переміщення файлів і директорій. Отримання докладної інформації про файли, за допомогою властивостей, визначених у цих класах. |
| FileStream | Доступ до файлів потоковим способом |
| MemoryStream | Доступ до даних що зберігаються в пам'яті |
| StreamWriter та StreamReader | Читання і запис текстової інформації |
| StringReader та StringWriter | Читання і запис текстової інформації з рядкового буфера |

**Робота з класами DirectoryInfo і FileInfo**

Класи DirectoryInfo і FileInfo успадковані від FileSystemInfo, який є абстрактним. Це означає, що Ви не можете успадкувати від нього свій клас, але можете використовувати властивості, визначені у ньому. Таблиця 2 перераховує його властивості та методи.

Таблиця 7.2 – Властивості та методи класу FileSystemInfo

|  |  |
| --- | --- |
| **Властивість** | **Значення** |
| Attributes | Повертає атрибути файлу у вигляді значень перерахування FileAttributes |
| CreationTime | Повертає час створення файлу |
| Exists | Перевіряє чи є файл директорією чи ні |
| Extension | Повертає розширення файлу |
| LastAccessTime | Повертає час останнього доступу до файлу |
| FullName | Повертає повний шлях до файлу |
| LastWriteTime | Повертає час останньої зміни файлу |
| Name | Повертає ім'я даного файлу |
| Delete() | Видаляє файл. Будьте обережні при використанні цього методу. |

Клас DirectoryInfo містить методи для створення, переміщення і видалення каталогів. Щоб використовувати вищенаведені властивості, необхідно створити об'єкт класу DirectoryInfo як показано в прикладі:

DirectoryInfo dir1 = new DirectoryInfo(@"F:\WINNT");

Після цього вже можна переглянути властивості директорії за допомогою об'єкта dir1, як показано у фрагменті коду:

Console.WriteLine("Повне ім’я : {0}", dir1.FullName);

Console.WriteLine("Атрибути : {0}", dir1.Attributes.ToString());

Можна також використовувати значення перерахування FileAttributes. Вони наведені в таблиці 3.

Таблиця 7.3 – СписокFileAttributes

|  |  |
| --- | --- |
| **Властивість** | **Значення** |
| Archive | Повертає Архівний статус файлу |
| Compressed | Дозволяє дізнатися стиснутий файл чи ні |
| Directory | Показує чи є файл директорією чи ні |
| Encrypted | Показує закодований файл чи ні |
| Hidden | Показує прихований файл чи ні |
| Offline | Показує, що дані відсутні |
| ReadOnly | Показує чи є файл тільки для читання |
| System | Показує, чи є файл системним (можливо файл в директорії Windows) |

**Робота з файлами в директорії**

Припустимо, ви хочете отримати список всіх файлів з розширенням BMP в папці F: \ Pictures. Для цього можна використовувати наступний код:

DirectoryInfodir = new DirectoryInfo(@"F:\WINNT");

FileInfo[] bmpfiles = dir.GetFiles("\*.bmp);

Console.WriteLine("Загальна кількість Bmp файлів", bmpfiles.Length);

Foreach( FileInfo f inbmpfiles)

{

Console.WriteLine("Ім’я : {0}", f.Name);

Console.WriteLine("Довжина файлу : {0}", f.Length);

Console.WriteLine("Час творення : {0}", f.CreationTime);

Console.WriteLine("Атрибути файлу : {0}",

f.Attributes.ToString());

}

**Створення підкаталогів**

Наступний фрагмент коду описує як можна створити піддиректорію MySub вдиректорії Sub:

DirectoryInfodir = newDirectoryInfo(@"F:\WINNT");

try

{

dir.CreateSubdirectory("Sub");

dir.CreateSubdirectory(@"Sub\MySub");

}

catch(IOException e)

{

Console.WriteLine(e.Message);

}

Створення файлів за допомогою класу FileInfo

Клас FileInfo дозволяє створювати нові файли, одержувати інформацію, видаляти і переміщати їх. У цьому класі також є методи для відкриття, читання і запису в файл. У наступному прикладі показано, як можна створити текстовий файл і отримати доступ до його інформації (часу його створення, повне ім'я, і так далі):

FileInfofi = new FileInfo(@"F:\Myprogram.txt");

FileStreamfstr = fi.Create();

Console.WriteLine("Час створення: {0}",f.CreationTime);

Console.WriteLine("Повне ім’я: {0}",f.FullName);

Console.WriteLine("Атрибути файлу: {0}",f.Attributes.ToString());

//Видалення файлу Myprogram.txt.

Console.WriteLine("Натисніть будь-яку клавішу, щоб видалити файл");

Console.Read();

fstr.Close();

fi.Delete();

Опис методу Open()

У класі FileInfo є метод під назвою Open() за допомогою якого можна створювати файли, підставляючи в параметри значення перерахувань FileMode і FileAccess. Наступний фрагмент коду показує, як це робиться:

FileInfo f = new FileInfo("c:\myfile.txt");

FileStream s = f.Open(FileMode.OpenorWrite, FileAccess.Read);

Після цього, використовуючи об'єкт 's', можна читати і записувати в файл. У перевантаженому методі Open() можна тільки читати з файлу. Для запису в файл необхідно в параметрах відкриття використовувати значення FileAccess.ReadWrite. Таблиці 4 і 5 містять можливі значення FileMode і FileAccess.

Таблиця 7.4 – значення FilrMode

|  |  |
| --- | --- |
| **Значення** | **Призначення** |
| Append | Для відкриття файлу і додавання даних. Використовується спільно зі значенням FileAccess.Write. |
| Create | Для створення нового файлу. Якщо файл вже існує, то він затирається. |
| CreateNew | Для створення нового файлу. Якщо файл існує, то виникає виключення IOException. |
| Open | Для відкриття існуючого файлу |
| OpenOrCreate | Для відкриття існуючого або створення нового файлу. Якщо файл не існує, то буде створено новий. |
| Truncate | Для урізання існуючого файлу |

Таблиця 7.5 – значення FilrAccess

|  |  |
| --- | --- |
| **Значення** | **Призначення** |
| Read | Для читання (отримання) даних з файлу |
| ReadWrite | Для запису в або читання з файлу |
| Write | Для запису даних у файл |

Запис в текстовий файл за допомогою класу StreamWriter

Текстові дані або будь-яку іншу інформацію можна записати у файл використовуючи метод CreateText() в класі FileInfo. Однак попередньо необхідно отримати валідний StreamWriter. Саме StreamWrite забезпечує необхідну функціональність для запису у файл. Наступний приклад ілюструє це:

FileInfo f = new FileInfo("Mytext.txt")

StreamWriter w = f.CreateText();

w.WriteLine("Це з");

w.WriteLine("Глава 6");

w.WriteLine("C# Модуль");

w.Write(w.NewLine);

w.WriteLine("Дякуємо за очікування");

w.Close();

Читання з текстового файлу

Для читання з текстового файлу можна скористатися класом StreamReader. Для цього необхідно вказати ім'я файлу в статичному методі OpenText() класу File. Наступний приклад зчитує вміст файлу, яке було записано в попередньому прикладі:

Console.WriteLine("Читання вмісту з файлу");

StreamReader s = File.OpenText("Mytext.txt");

stringread = null;

while ((read = s.ReadLine()) != null)

{

Console.WriteLine(read);

}

s.Close();

Робота з різними кодуваннями

**За замовчуванням в .NET всі текстові дані в кодуванні UTF8**, але часто потрібно зчитати текстовий файл, збережений в іншому кодуванні, допустимо в WIN1251. У такому випадку якщо у файлі був український чи російський текст в кодуванні WIN1251, при зчитуванні його в UTF8 ми отримаємо нечитаємі дані. Для того щоб переводити рядки з одного кодування в інше існує клас Encoding з простору імен System.Text, завдяки якому знаючи вихідну кодування ми можемо привести текстові дані до потрібного кодування. Розглянемо наступний приклад, наприклад, ми хочемо конвертувати зчитаний рядок в кодуванні WIN1251 в кодування DOS (866), для цього можна використовувати наступний код:

classProgram

{

privatestaticstring in1251;

privatestaticreadonlyEncoding enc1251 = Encoding.GetEncoding(1251);

privatestaticreadonlyEncoding enc866 = Encoding.GetEncoding(866);

staticvoidMain(string[] args)

{

//....

//тут якимось чином отримуємо дані в in1251

byte[] sourceBytes = enc1251.GetBytes(in1251);

stringoutputString = enc866.GetString(sourceBytes);

//даліробимоте, щонеобхіднозотриманимрядком

//....

}

}

У sourceBytes ми отримали вхідний рядок у вигляді масиву байт, які далі можна так само за допомогою Encoding зберегти в рядок з потрібним кодуванням. У випадку, коли ми читаємо текст з файлу з уже відомим кодуванням, справи йдуть ще простіше. При створенні екземпляра StreamReader можна явно вказати кодування джерела, код буде виглядати наступним чином:

using (varsr = new StreamReader("Mytext.txt", Encoding.GetEncoding(1251)))

{

stringread = null;

while ((read = sr.ReadLine()) != null)

Console.WriteLine(read);

}

При записі у файл із належним кодуванням для StreamWriter так само можна вказати кодування, в якому буде текст.

**Приклад роботи з файлами в додатках Windows Forms**

Розмістіть на форму програми 4 кнопки:

* button1, у її властивість Text запишіть «Створити файл».
* button2, у її властивість Text запишіть «Видалити файл».
* button3, у її властивість Text запишіть «Записати текст у файл».
* button4, у її властивість Text запишіть «Прочитати файл».

Додайте на форму текстове поле textBox, встановіть його властивість Multiline в true (ввімкнення багаторядкового режиму)

Приклад форми, яка повинна вийти, зображений на рисунку 1.

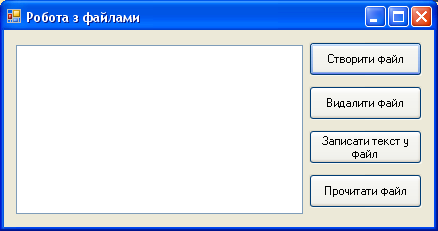


Рисунок 1 ­– Приклад програми для роботи з файлами

**Приклад коду програми:**

**using** System;

**using** System.Collections.Generic;

**using** System.ComponentModel;

**using** System.Data;

**using** System.Drawing;

**using** System.Linq;

**using** System.Text;

**using** System.Windows.Forms;

**using** System.IO; *//Підключаємо простір імен для роботи з файлами*

**namespace** Files

{

**publicpartialclass** Form1 : Form

{

**public** Form1()

{

InitializeComponent();

FileInfo file = **new** FileInfo(**"book.txt"**);

}

*// Створення файлів*

**privatevoid** button1\_Click(**object** sender, EventArgs e)

{

FileInfo file = **new** FileInfo(**"book.txt"**);

**if** (file.Exists == **false**) *//Якщо файл не існує*

{

file.Create(); *//Створюємо*

}

**else** MessageBox.Show(**"Файл уже існує!"**);

}

*//Видалення файлу*

**privatevoid** button2\_Click(**object** sender, EventArgs e)

{

FileInfo file = **new** FileInfo(**"book.txt"**);

**if** (file.Exists == **true**) *//Якщо файл існує*

{

file.Delete(); *//Видаляємо*

}

**else** MessageBox.Show(**"Такого файлу не існує!"**);

}

*// Запис тексту у файл*

**privatevoid** button3\_Click(**object** sender, EventArgs e)

{

StreamWriter write\_text; *//Клас для запису у файл*

FileInfo file = **new** FileInfo(**"book.txt"**);

write\_text = file.AppendText(); *//Дописуємо дані у файл, якщо файлу не існує він створиться*

write\_text.WriteLine(textBox1.Text); *//Записуємо у файл текст із текстового поля*

write\_text.Close(); *// Закриваємо файл*

}

*// Читання з файлу*

**privatevoid** button4\_Click(**object** sender, EventArgs e)

{

StreamReader streamReader = **new** StreamReader(**"book.txt"**); *//Відкриваємо файл для читання*

**string** str = **""**; *//Повідомляємо змінну, у яку будемо записувати текст із файлу*

**while** (!streamReader.EndOfStream) *//Цикл триватиме поки не буде досягнутий кінець файлу*

{

str += streamReader.ReadLine(); *//У змінну str по рядках записуємо вміст файлу*

}

textBox1.Text = str;

}

}

}

**Завдання до лабораторної роботи:**

**Варіант 1**

Створити файл, заповнити його випадковими числами. Вивести вміст файлу на екран.

**Варіант 2**

Створити файл, заповнити його випадковими цілими числами. Обчислити середнє арифметичне чисел, що перебувають у файлі, та дописати це значення в кінець файлу.

**Варіант 3**

Потрібно створити файл заповнений цілими числами, у якому значення кожного елемента хi+1 дорівнює (хi)2 і квадрати всіх чисел не перевершують n.

**Варіант 4**

Створити два файли заповнені відсортованими за зростанням цілими числами. Створити третій файл та записати в нього числа з першого та другого. Числа в третьому файлі також повинні розташовуватися за зростанням.

**Варіант 5**

Створити файл, заповнити його цілими числами. Знайти найбільше число у файлі вивести його на екран та записати у кінець файлу.

**Варіант 6**

Створити файл, записати в нього рядки, що містять прізвища власників і номери їхніх телефонів. Реалізувати можливість пошуку номера телефона у файлі за прізвищем власника (користувач вводить у перше текстове поле на формі прізвище, програма виводить в друге текстове поле знайдений у файлі номер телефону, якщо прізвище не знайдено, виводиться повідомлення про відсутність номеру телефону).

**Варіант 7**

Створити файл, заповнити його випадковими числами. Створити дві копії даного файлу. Одну з копій видалити. В іншій копії знайти мінімальне число та вивести його на екран.

**Варіант 8**

Створити програму для підрахунку кількості чисел у файлі, менших середнього арифметичного всіх елементів цього файлу.

**Варіант 9**

Створити програму, що міняє місцями перший і останній символи у вказаному файлі.

**Варіант 10**

Створити текстовий файл. Записати в нього текст, введений у текстове поле форми програми. Створити другий текстовий файл, у який записати кількість букв «А», «Б» та «В» у вказаному файлі. Вивести на екран максимальне число з другого файлу та до якої букви воно відноситься.

**Варіант 11**

Створити файл, заповнити його випадковими дійсними числами, обчислити суму елементів цього файлу та вивести її на екран.

**Варіант 12**

Створити файл, заповнити його випадковими буквами. Вказати в одному текстовому полі форми програми букву, а в трогуму цифру. Створити другий файл, у якому кожний елемент, що являється вказаною буквою замінити на вказану цифру.

**Варіант 13**

Створити текстовий файл. Заповнити його буквами та цифрами. Прочитати створений файл і знайти суму цифр, що зустрічаються в ньому.

**Варіант 14**

Створити файл заповнений випадковими позитивними та негативними числами. Прочитати вміст файлу та створити ще два файли, в перший записати всі позитивні числа, в другий – всі негативні.

**Варіант 15**

Створити текстовий файл. Заповнити його текстом. Прочитати створений файл і вивести його вміст на екран задом наперед.

***Контрольні запитання:***

1. В якому просторі імен зберігаються методи для роботи з файлами та каталогами?
2. Який метод необхідно використати для створення нового файлу та генерування виключення, якщо такий файл вже існує?
3. Для чого використовуються класи StringReader та StringWriter?
4. За допомогою якого класу можна отримати доступ до даних, що зберігаються в пам’яті?