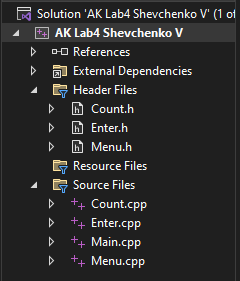
*Шевченко Владислав Володимирович, ПЗ-22-1*

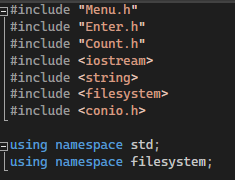
**Лабораторна робота №4**

Запустимо IDE Microsoft Visual Studio 2022 Community, створимо проект .sln зі стандартом С++20 Standard.

Створимо 3 додаткові класи, напишемо в кожний методи, що відповідають за певну з трьох основних функцій роботи програми – виведення, введення і рахування, дамо відповідним файлам .h і .cpp назви “Menu”, “Enter” і “Count” відповідно.



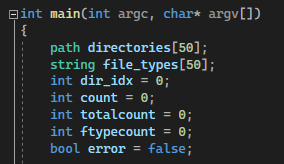
Під час написання коду використовуватимемо бібліотеки “iostream” як базову, “string” для детальної роботи з об’єктами типу string, “conio.h” для підтримки функції “\_getch()” і, врешті-решт, бібліотеку “filesystem” для роботи з файловою системою Windows.



--------------------------------------------------------------------------------------------------------------------------

Напишемо код для всіх трьох класів, код для функції main.

В функції main зазначимо наступні параметри та ініціалізуємо такі змінні:

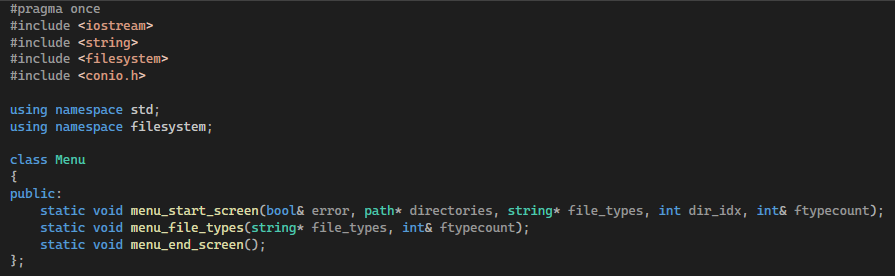


Надалі, кожна зі змінних матиме в коді наступне значення:

* масив directories міститиме всі введені користувачем папки (якщо такі існують);
* масив file\_types міститиме всі текстові записи введених користувачем певних форматів файлів (якщо такі існують);
* dir\_idx використовуватиметься для роботи з кількістю папок для підрахунку файлів;
* count використовуватиметься для розрахунку індивідуальної кількості файлів в кожній з зазначених папок;
* totalcount позначатиме загальну кількість файлів в усіх зазначених папках (якщо їх більше за одну);
* ftypecount використовуватиметься для розрахунку кількості введених користувачем форматів файлів (якщо такі існують);
* error використовуватиметься для генерації і виведення кодів завершення роботи програми.

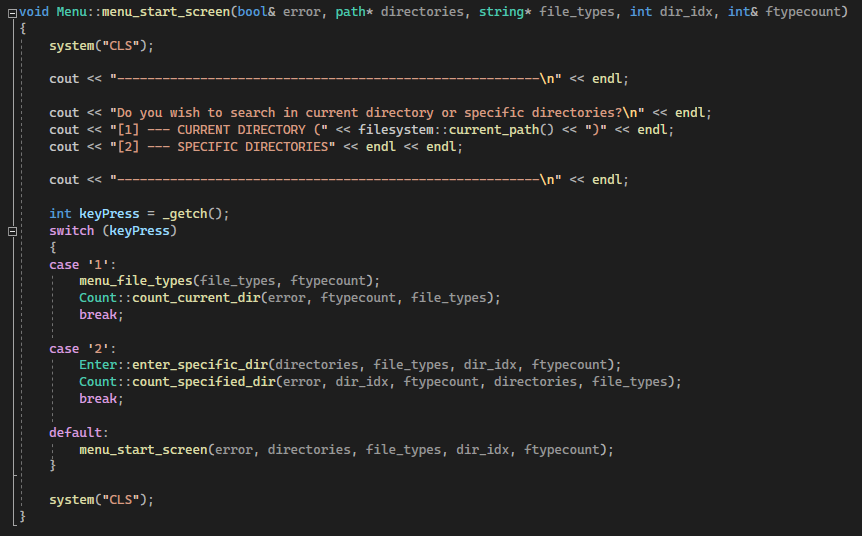
--------------------------------------------------------------------------------------------------------------------------

Розглянемо клас Menu, відповідальний за взаємодію користувача з певними елементами інтерфейсу:



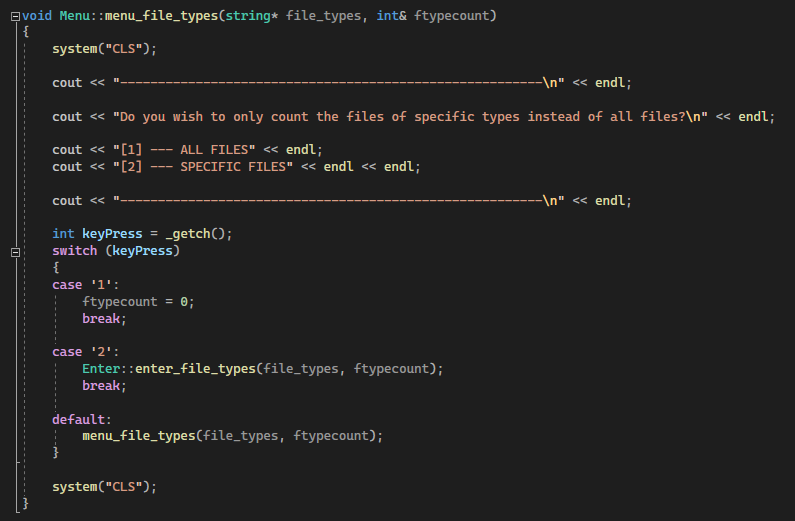
Використаємо модифікатор доступу public, зробимо функції статичними. Розглянемо кожну з них.

Функція “menu\_start\_screen”, відповідальна за меню вибору користувачем кількості робочих папок, коли необхідно, на початку роботи програми:



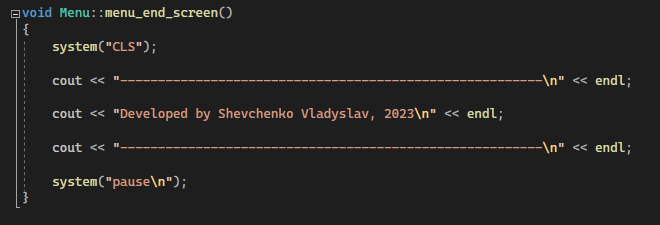
Після виведення елементів інтерфейсу, користувач зможе обрати будь-яку з запропонованих опцій, натискаючи клавішу на клавіатурі. Функція “\_getch()” буде отримувати інформацію стосовно натиснутої клавіші, і в залежності від того чи клавішою була 1, чи 2, програма працюватиме за одним із двох можливих шляхів роботи. Якщо натиснута клавіша не була 1 чи 2, програма ігнорує її і, завдяки рекурсії, пропонує користувачеві натиснути клавішу знов.

Функція “menu\_file\_types”, відповідальна за меню вибору користувачем наявності обмежень за форматами файлів під час підрахунку:



Функція працює за схожим до попередньої принципом, де єдина різниця в тому, що у разі відсутності обмежень стосовно форматів файлів функція повертатиме значення кількості обмежень “ftypecount” зі значенням 0, і лише в протилежному випадку викликатиме функцію для введення форматів шуканих файлів.

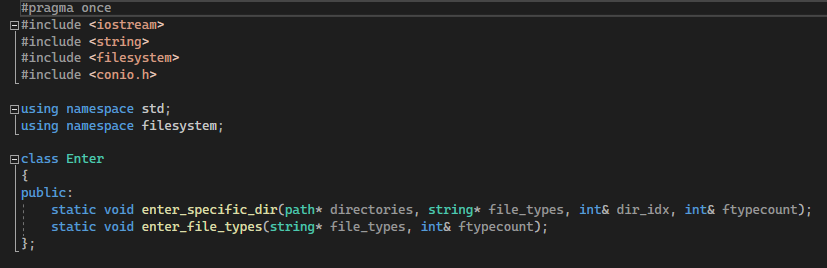
Функція “menu\_end\_screen”, відповідальна за екран завершення роботи програми:



Після виведення елементів інтерфейсу, використовується “system(“pause”)” для підтвердження користувачем завершення роботи програми.

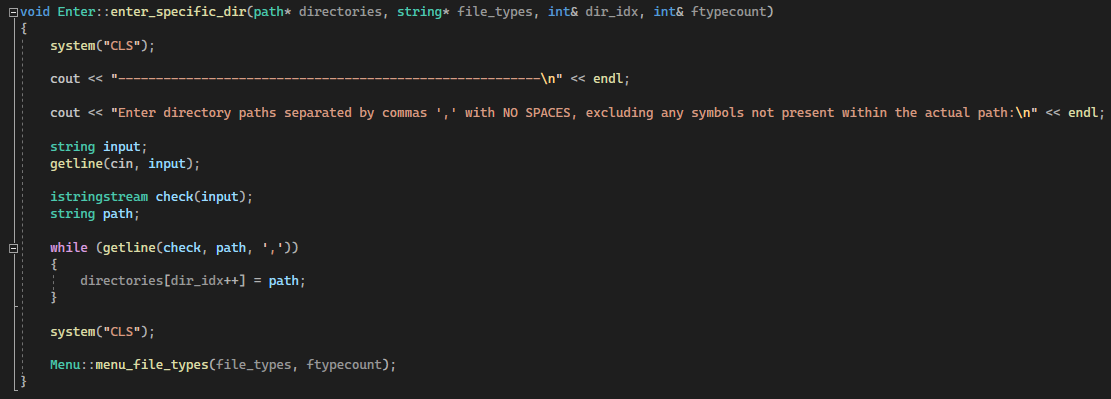
--------------------------------------------------------------------------------------------------------------------------

Розглянемо клас Enter, відповідальний за введення користувачем даних і їх обробку:



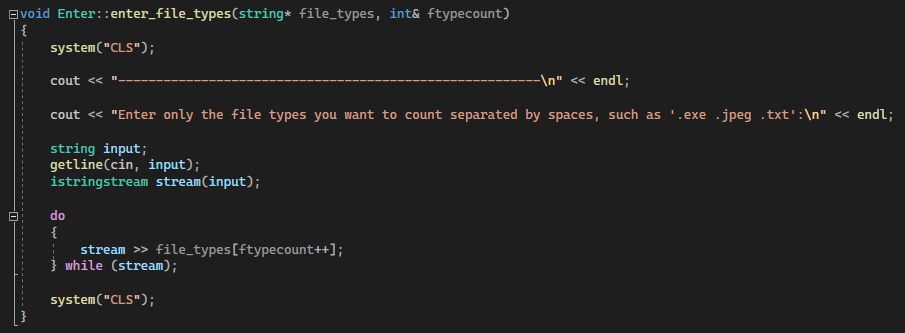
Використаємо модифікатор доступу public, зробимо функції статичними. Розглянемо кожну з них.

Функція “enter\_specific\_dir”, відповідальна за введення користувачем папок, в яких відбуватиметься пошук файлів:



Після виведення елементів інтерфейсу, отримаємо введені користувачем дані завдяки методу “getline”, розділимо дані як одне ціле на об’єкти, кожен з яких репрезентуватиме індивідуальний шлях до кожної з зазначених папок, використовуючи змінну типу istringstream і винесення даних в масив “directories”, використовуючи метод “getline” з комою у якості роздільника (‘,’) між об’єктами. Нарешті, функція викликатиме вищезгаданий метод “menu\_file\_types”, де користувач матиме змогу обрати, чи хоче, щоб підрахунок відбувався лише для файлів певних типів.

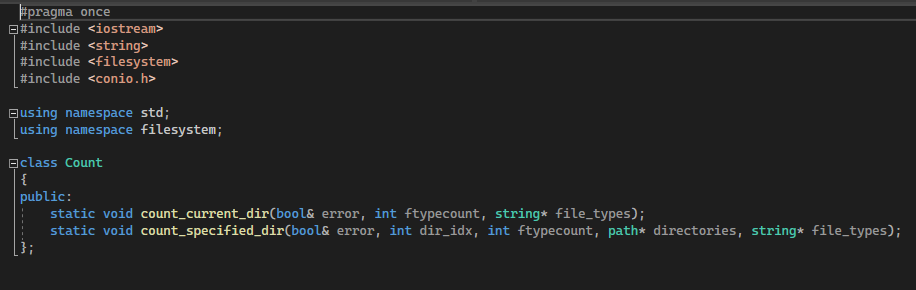
Функція “enter\_file\_types”, відповідальна за введення користувачем типів файлів, за якими відбуватиметься підрахунок, викликається раніше згаданим методом ”menu\_file\_types”:



Після виведення елементів інтерфейсу, метод використовує схожим на попередню функцію чином метод “getline” і змінну типу istringstream для розділення виведених даних за прогалинами і виведення сформованих індивідуальних елементів в масив “file\_types”.

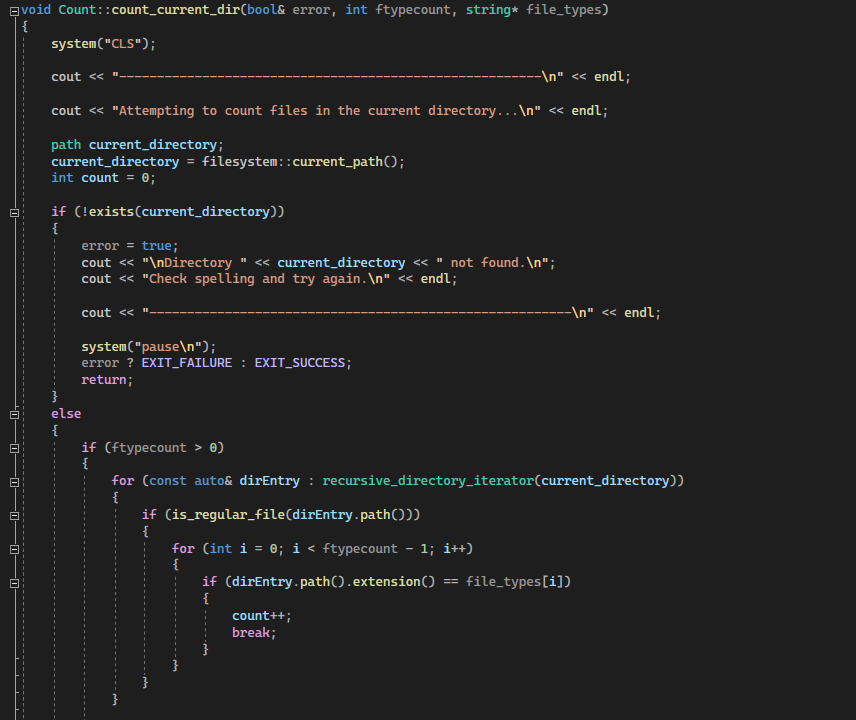
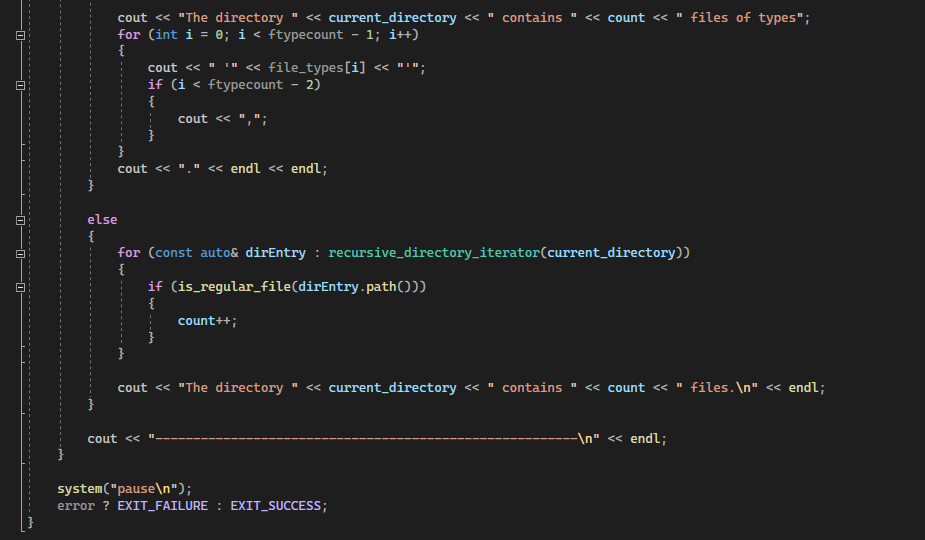
--------------------------------------------------------------------------------------------------------------------------

Розглянемо клас Count, відповідальний за основні операції роботи програми, в даному випадку – підрахунок кількості файлів в папки/папках:



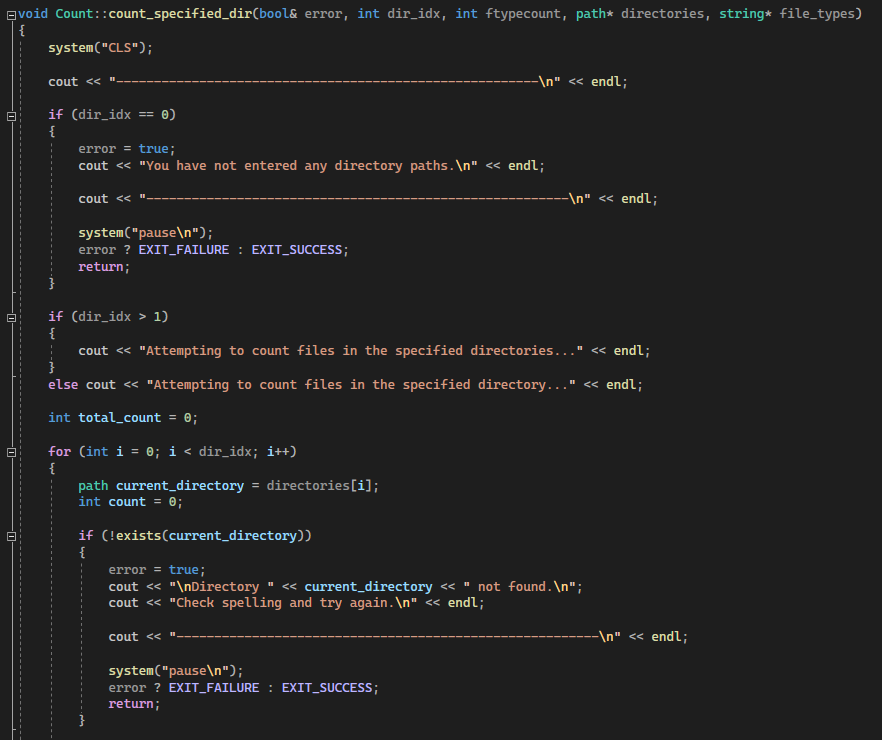
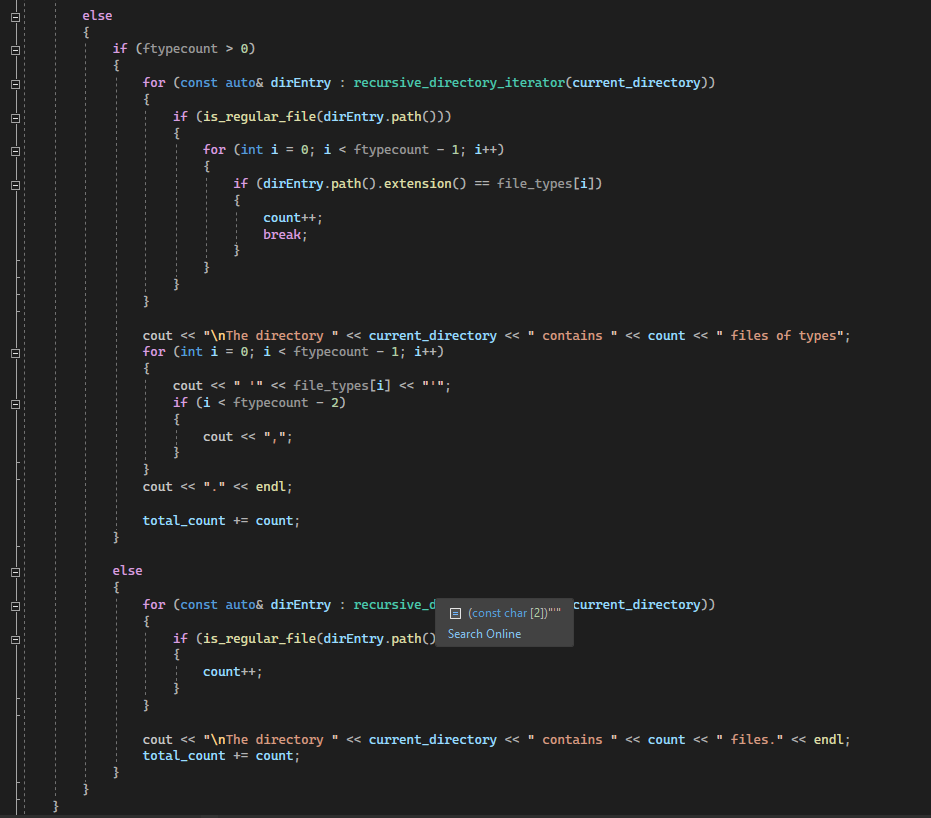
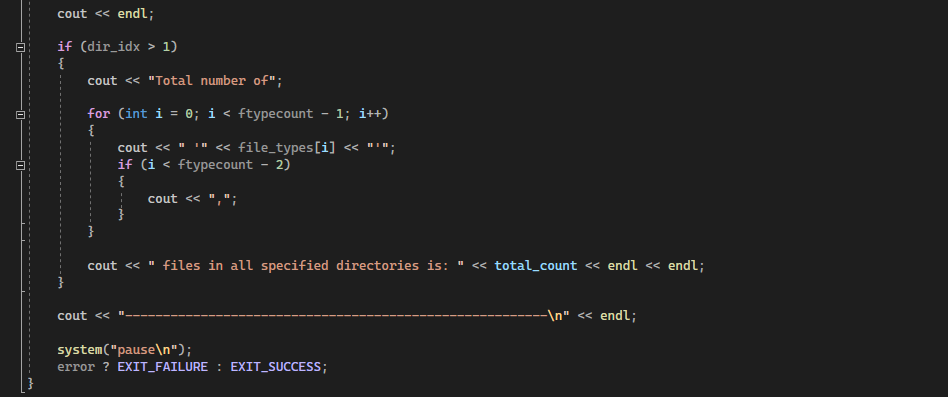
Використаємо модифікатор доступу public, зробимо функції статичними. Розглянемо кожну з них.

Функція “count\_current\_dir”, відповідальна за підрахунок кількості файлів в поточній папці:

Зазначається перемінна типу path зі значенням шляху до поточної папки. Для менеджменту можливих помилок з ініціалізацією змінної, використовується функція “exists” в рамках if-else, де в else відбувається основний підрахунок загальної кількості файлів в папки, в залежності від введених обмежень на типи файлів, якщо такі існують, з використанням ітератора і функцій “is\_regular\_file” та “path”, і виведення цієї кількості. Нарешті, змінній “error” надається значення, в залежності від успішності роботи програми.

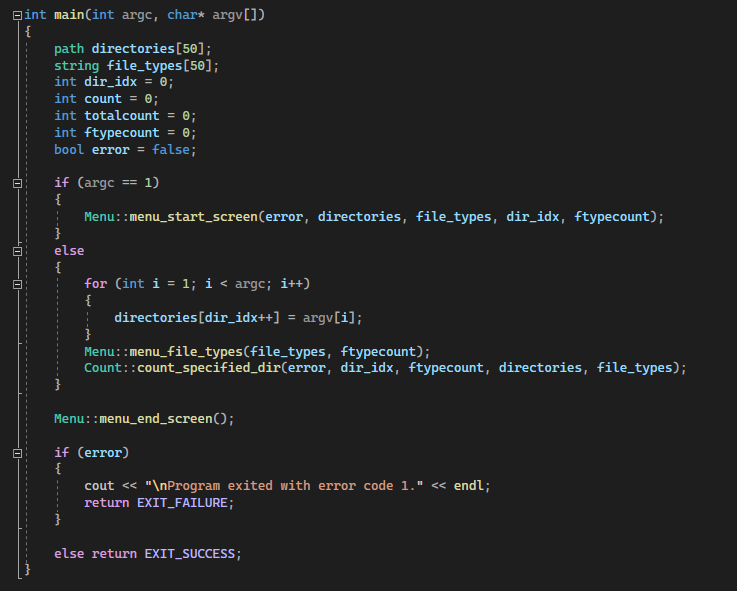
Функція “count\_specified\_dir”, відповідальна за підрахунок кількості файлів в зазначених папках:

Функція має фундаментально ідентичні алгоритми до попередньої, відрізняється здебільшого використанням змінної “totalcount” для підрахунку загальної кількості у всіх папках, додатковим сценаріям виведенням даних і блокам менеджменту помилок, як, наприклад відсутність введених даних, або відсутність зазначеної папки в системі тощо.

--------------------------------------------------------------------------------------------------------------------------

Нарешті, розглянемо функцію main та її алгоритми:



Функція використовує основні методи класів Menu та Count, більшість з яких використовується для початку або завершення роботи інтерфейсу і основних елементів програми. Через умову if-else на початку, передбачено введення папок пошуку в консолі Windows як параметрів файлу.

Нарешті, якщо в будь-якому з методів сталася передбачена помилка, main виведе текстове повідомлення, і завершить роботу програми з кодом 1. В іншому випадку, робота програми завершиться з кодом 0.

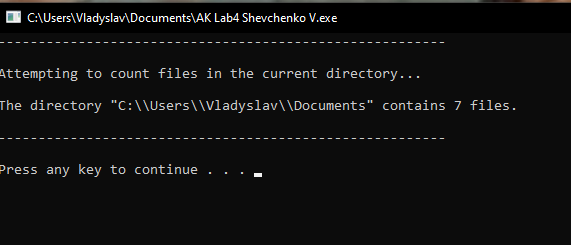
--------------------------------------------------------------------------------------------------------------------------

Таким чином, ми написали працездатний і функціональний код для програми, що вестиме підрахунок кількості файлів будь-яких форматів в поточному або зазначених каталогах, виводитиме підказки користувачеві, формуватиме коди завершення і враховуватиме при підрахунку файли з атрибутами.

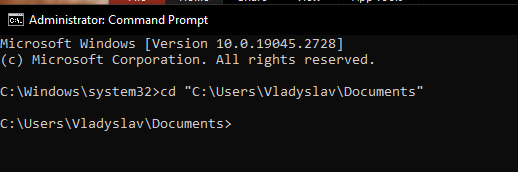
Для доведення, протестуємо програму в різних сценаріях, запуском безпосередньо .exe файлу та його викликом через консоль Windows для кожного.

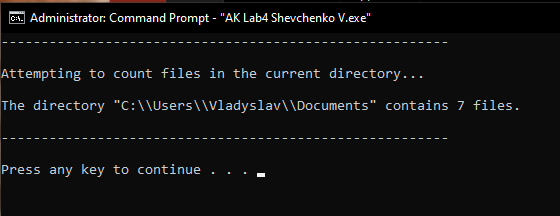
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 1.* Рахування і виведення кількості всіх файлів в поточній папці.

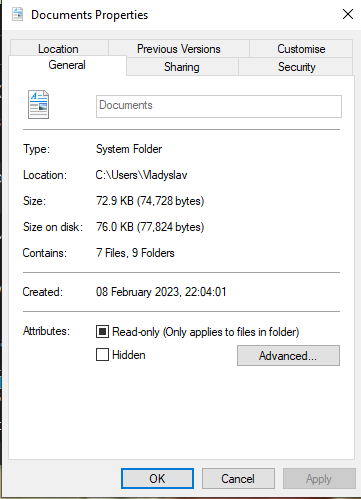


Зазначимо ту ж саму папку як поточну, в консолі Windows:



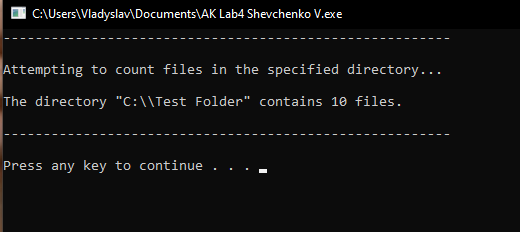


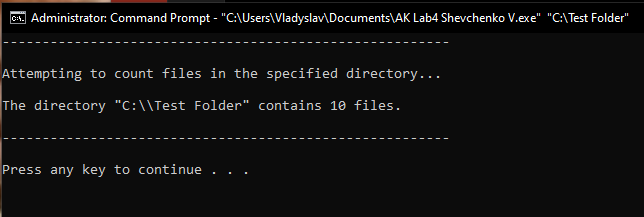
Звіримо результати, переконаємося, що вони правдиві:



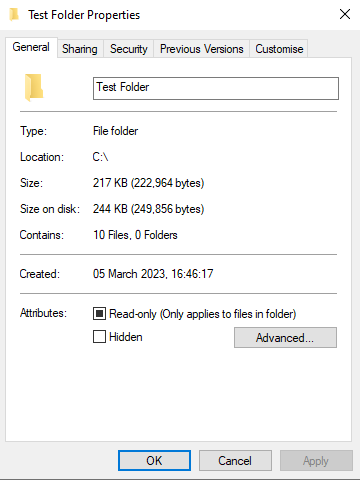
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 2.* Рахування і виведення кількості всіх файлів в зазначеній користувачем папці.



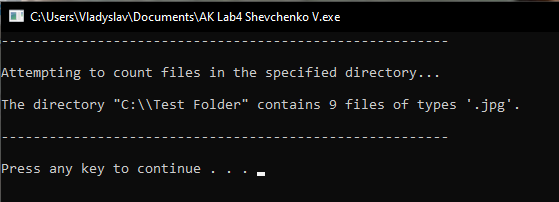


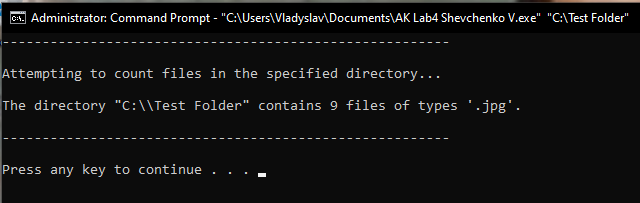
Звіримо результати, переконаємося, що вони правдиві:



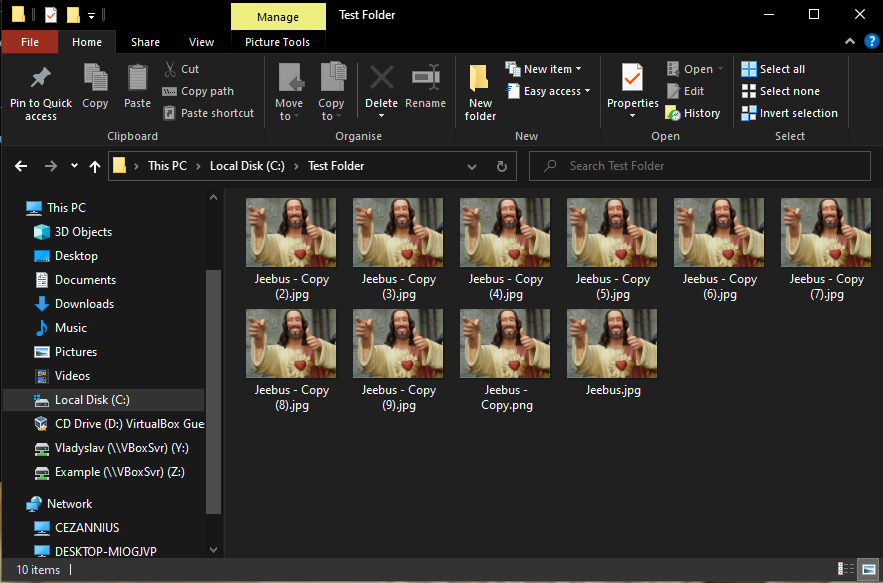
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 3.* Рахування і виведення кількості файлів типу “.jpg” в зазначеній користувачем папці.





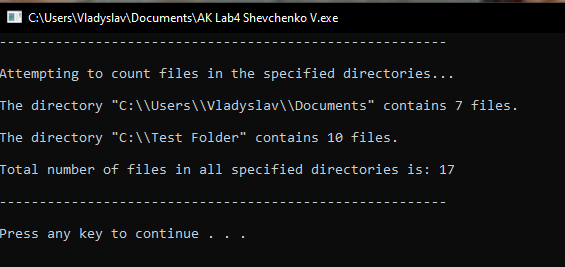
Відкриємо папку ”Test Folder”:

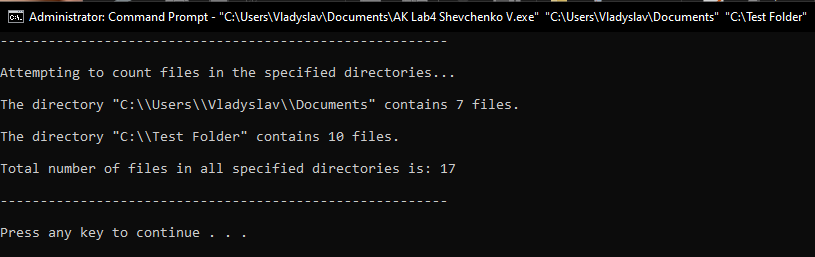


Побачимо в ній 10 файлів, лише 9 з яких мають розширення .jpg, згадаємо, що 9 і є результатом роботи програми.

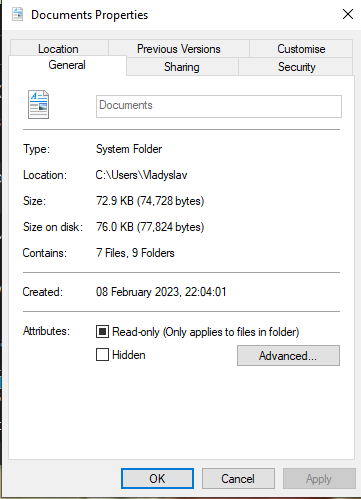
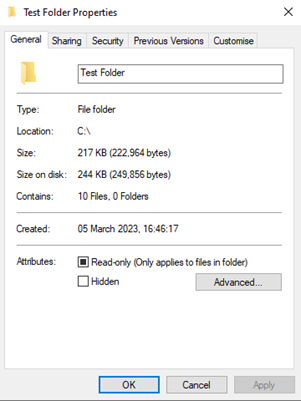
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 4.* Рахування і виведення кількості всіх файлів в двох зазначених користувачем папках.



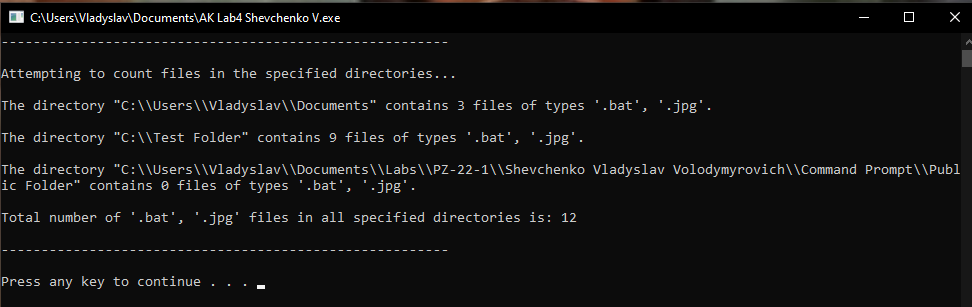


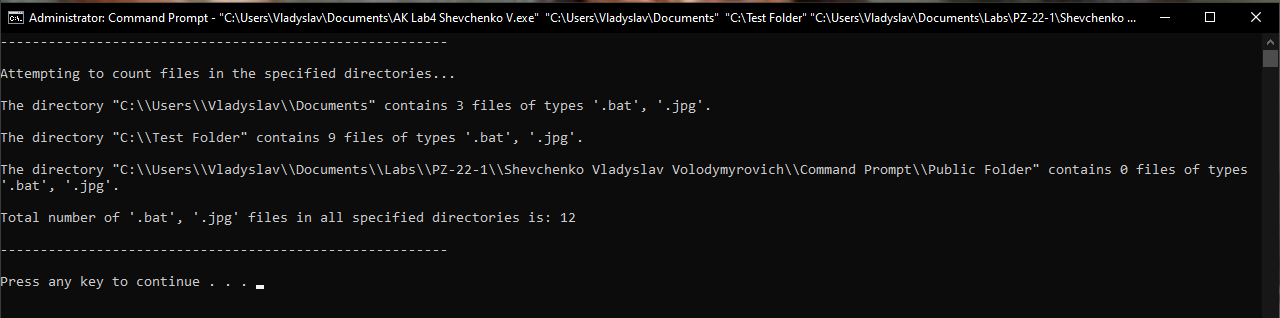
Звіримо результати, переконаємося, що вони правдиві:

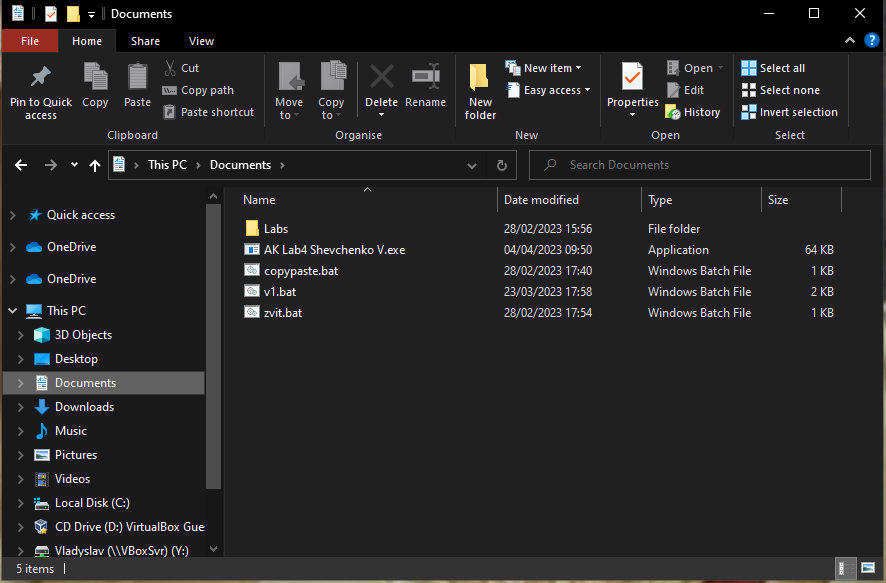
--------------------------------------------------------------------------------------------------------------------------

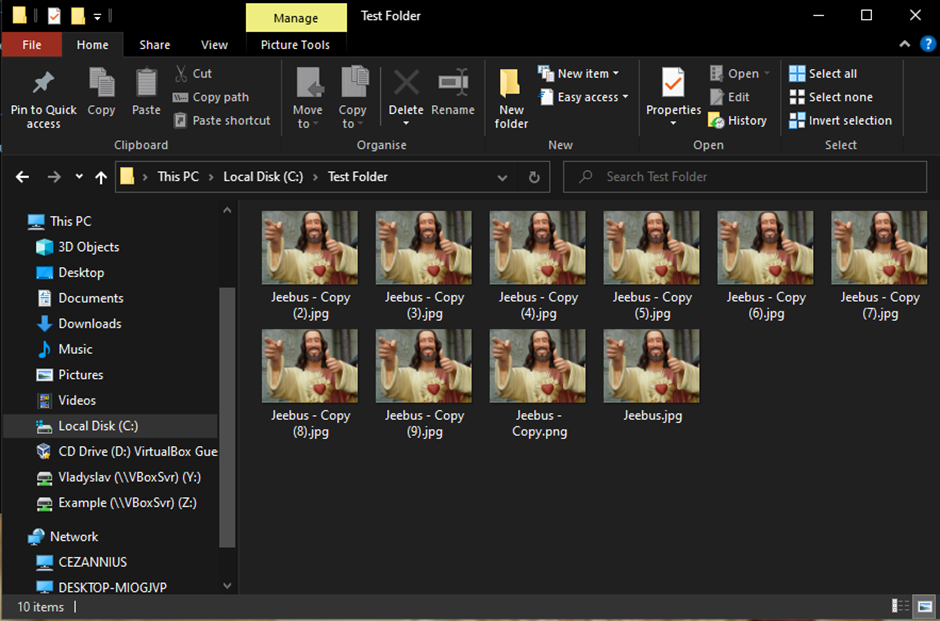
*Сценарій 5.* Рахування і виведення кількості файлів типів “.bat” та “.jpg” в трьох зазначених користувачем папках.

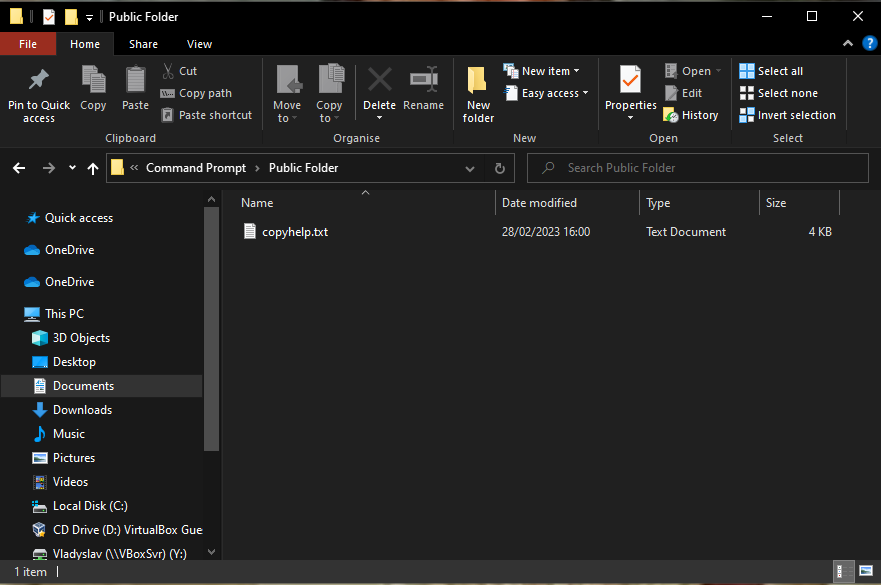




Відкриємо папки “Documents”, “Test Folder”, “Public Folder”; переконаємося в правильності виведених даних:



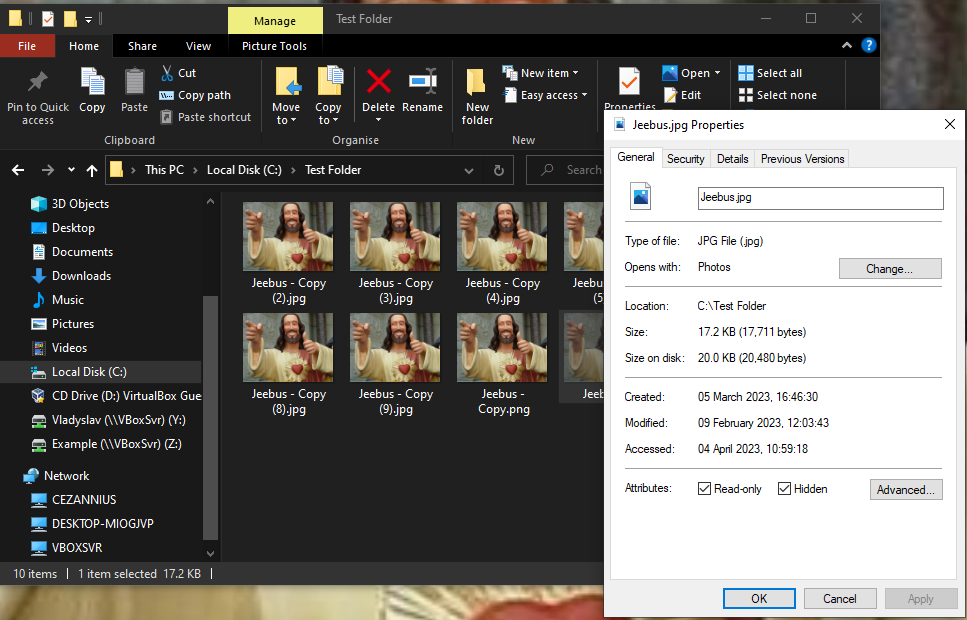


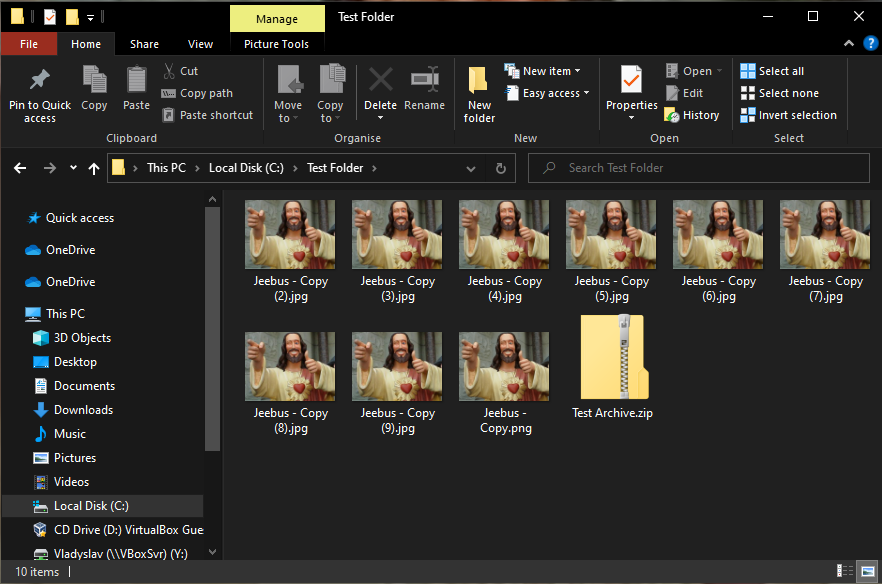


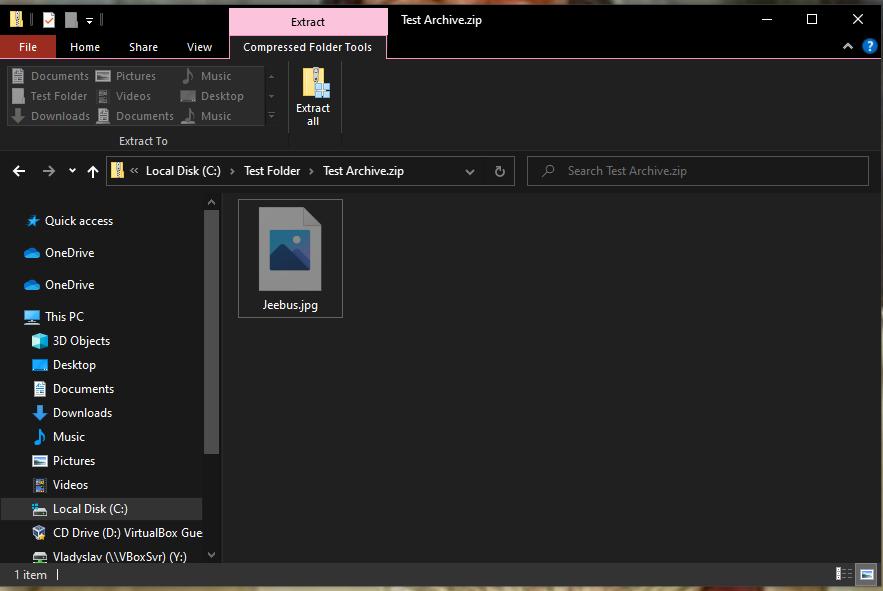
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 6.* Рахування і виведення кількості всіх файлів і атрибутів в зазначеній користувачем папці.

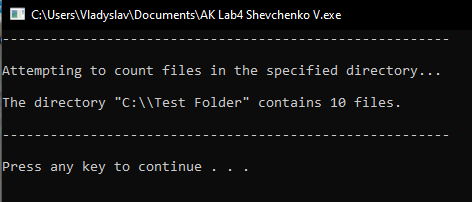
Для перевірки працездатності програми з файлами з атрибутами («тільки для читання», «прихований», «архівний») додамо атрибути “Read-only” і “Hidden” до файлу “Jeebus.jpg” папки “Test Folder” і занесемо файл в архів:

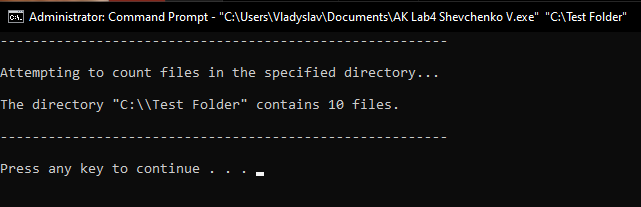




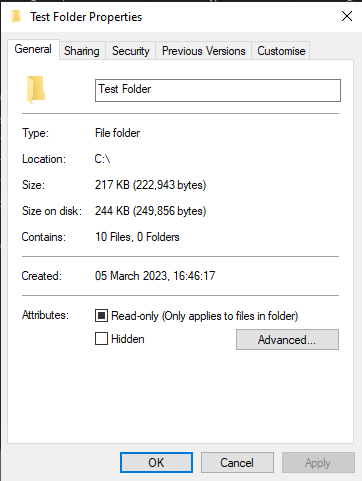


Перевіримо роботу програми зі зміненою папкою ”Test Folder”:



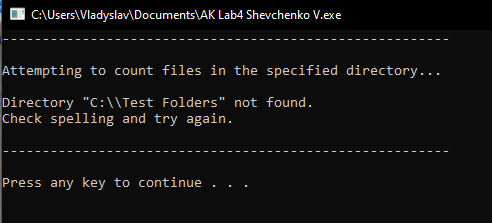


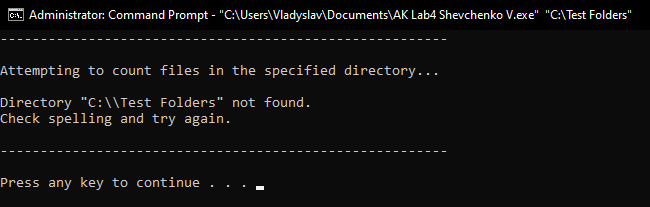
Звіримо результати, переконаємося, що вони правдиві:



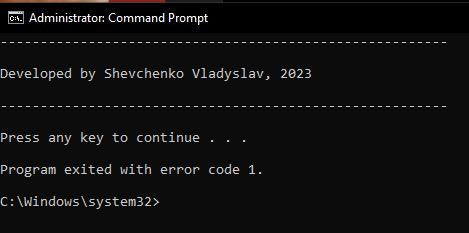
--------------------------------------------------------------------------------------------------------------------------

*Сценарій 7.* Рахування і виведення кількості всіх файлів в некоректно зазначеній користувачем папці.





Код помилки буде зображено у разі завершення роботи програми:



--------------------------------------------------------------------------------------------------------------------------

**Висновок:** я навчився роботі з файловою системою у мовах програмування високого рівня, а саме С++ з використанням бібліотеки filesystem, створивши програму, що підраховує сумарний обсяг файлів в каталозі та має максимальний необхідний за вимогами завдання функціонал.