ЗМІСТ

[ЛАБОРАТОРНА РОБОТА № 1](#_Toc122955409) 2

[ЛАБОРАТОРНА РОБОТА № 2](#_Toc122955410) 5

[ЛАБОРАТОРНА РОБОТА № 3](#_Toc122955411) 11

[ЛАБОРАТОРНА РОБОТА № 4](#_Toc122955411) 19

Лабораторна робота № 1

«Компоненти ПК та їхня сумісність»

1. Постановка завдання.

Користуючись реальними даними про ціни та характеристики компонентів (rozetka.com.ua або інший прайс-лист комплектуючих за вибором, за умови погодження з викладачем), підібрати конфігурацію ПК відповідно до вимог індивідуального варіанту. Компоненти повинні бути сумісні між собою, а сумарна вартість приблизно відповідати вказаній (можливі незначні відхилення в той або інший бік). У звіті надати скриншоти з ціною і детальними характеристиками комплектуючих. Обґрунтувати вибір комплектуючих згідно відповідності до задачі, яка вирішуватиметься. Навести розрахунок мінімальної потужності блока живлення.

Мінімальний список компонентів:

* Материнська плата;
* Процесор (кулер в комплекті або окремо);
* Відеокарта (дискретна або вбудована, за доцільністю);
* Оперативна пам'ять;
* Корпус, блок живлення (разом або окремо);
* Жорсткий (HDD) та/або твердотільний (SSD) диск.

Індивідуальні варіанти завдань

| № | Задача | Чіпсет | Відео | ОЗП (RAM), Гб | HDD | SSD | Орієнтовна вартість |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | Дім, «на майбутнє» | Intel | Дискретне AMD | 32 | Так | Так | 30000 |

1. Результат виконання завдання.

Результат виконання завдання наведено на рисунках 1.1 ­­– 1.2.

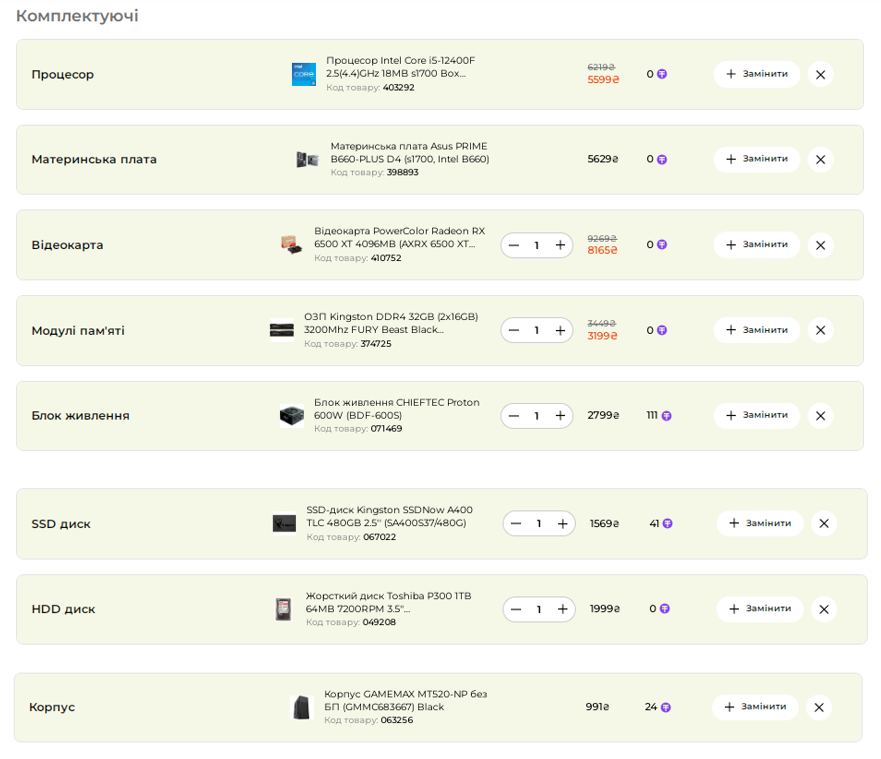


Рисунок 1.1 – Конфігурація ПК відповідно до вимог індивідуального варіанту

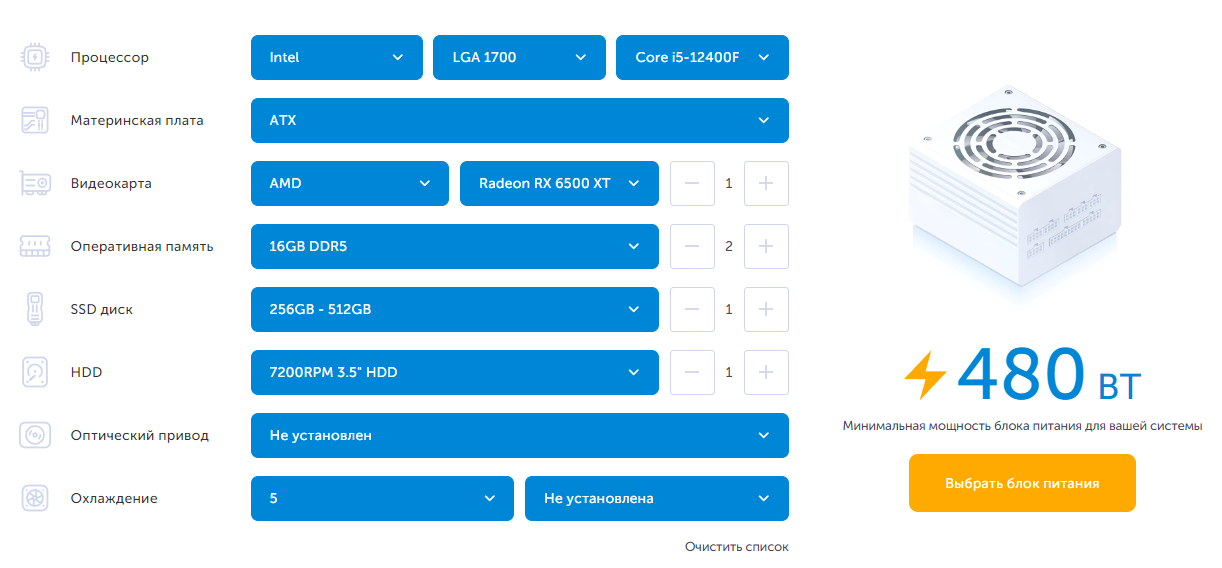


Рисунок 1.2 – Розрахунок мінімальної потужності блока живлення

Лабораторна робота № 2

«Робота з віртуальною машиною»

1. Постановка завдання.
2. Встановити віртуальну машину (VirtualBox або Parallels або VMWare Workstation).
3. Встановити на віртуальну машину операційну систему Windows. Щоб уникнути проблем з налаштуваннями, рекомендується Windows 10 або новіше. В операційній системі обов’язково назвати користувача своїм ім’ям.
4. У встановленій на віртуальну машину операційній системі встановити гостьові доповнення (наприклад VMWare пропонує їх при запуску VM). Що змінилося в роботі віртуальної машини?
5. Налаштувати доступ в Інтернет на віртуальній машині.
6. Налаштувати спільний каталог між реальною і віртуальною ОС, назвати цей каталог своїм ім’ям. Як ще можна отримати доступ до файлової системи віртуальної ОС?
7. Порівняти з Windows Sandbox
8. Результат виконання завдання.

Результат виконання завдання наведено на рисунках 2.1 ­­– 2.6.

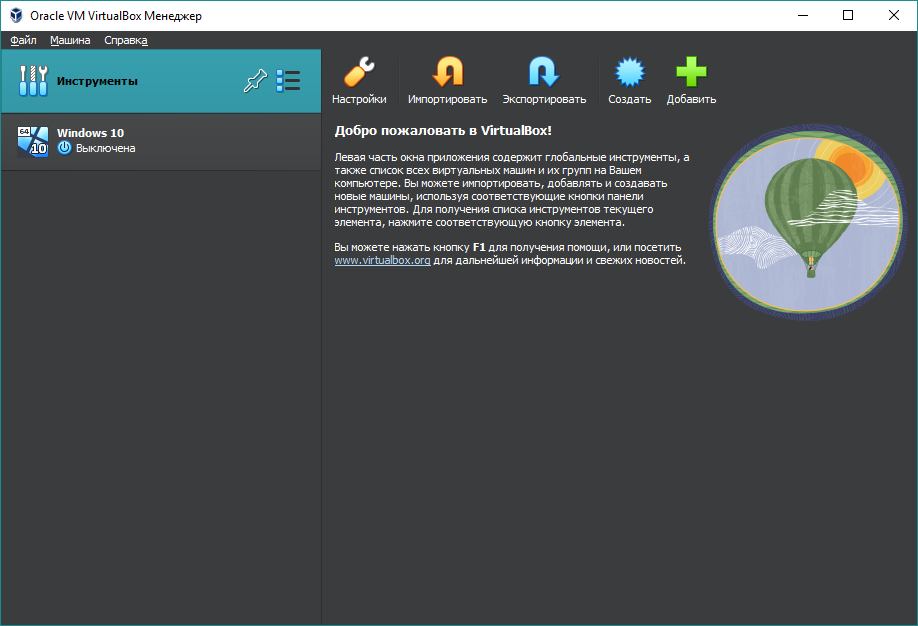


Рисунок 2.1 – Результат встановлення VM VirtualBox

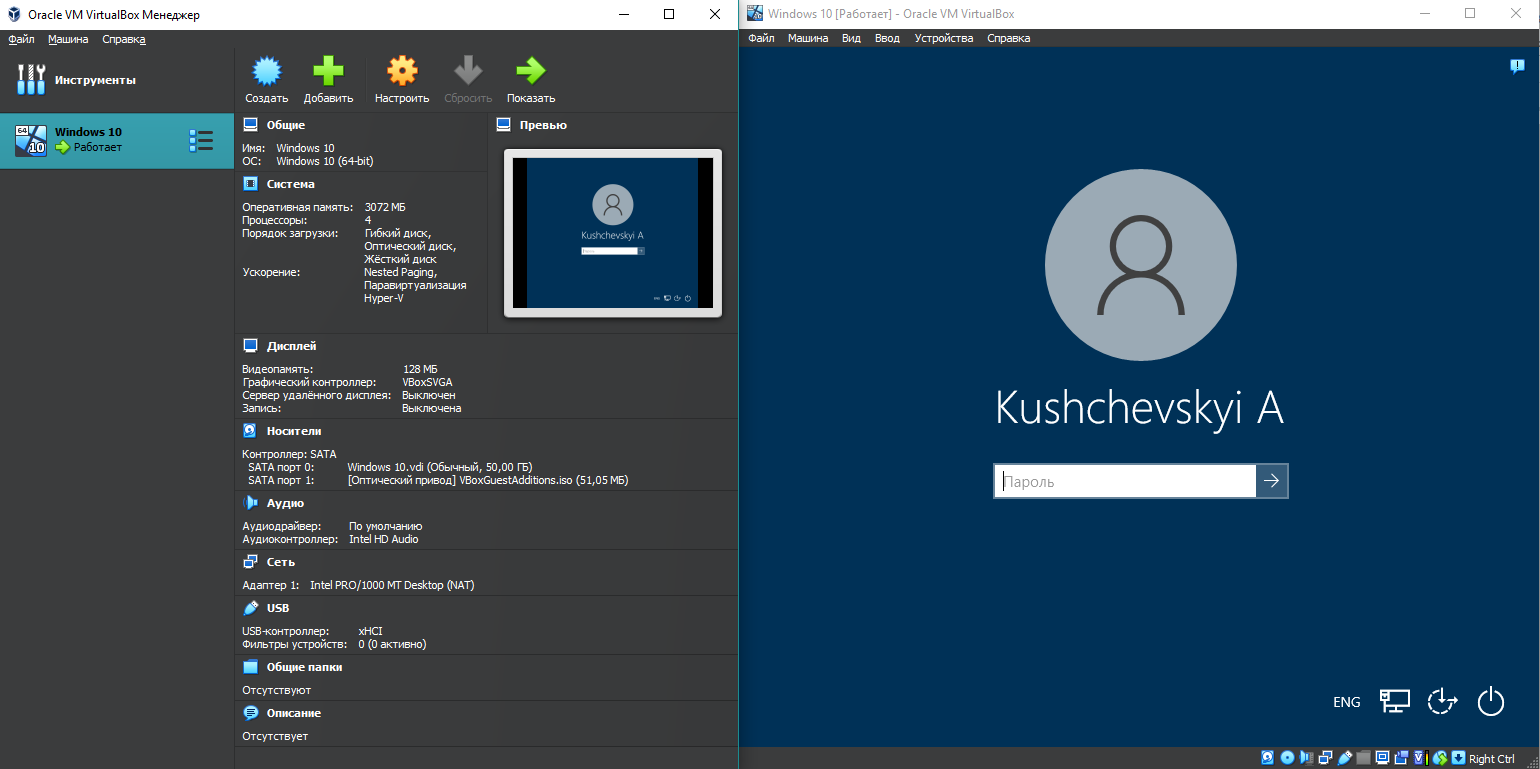
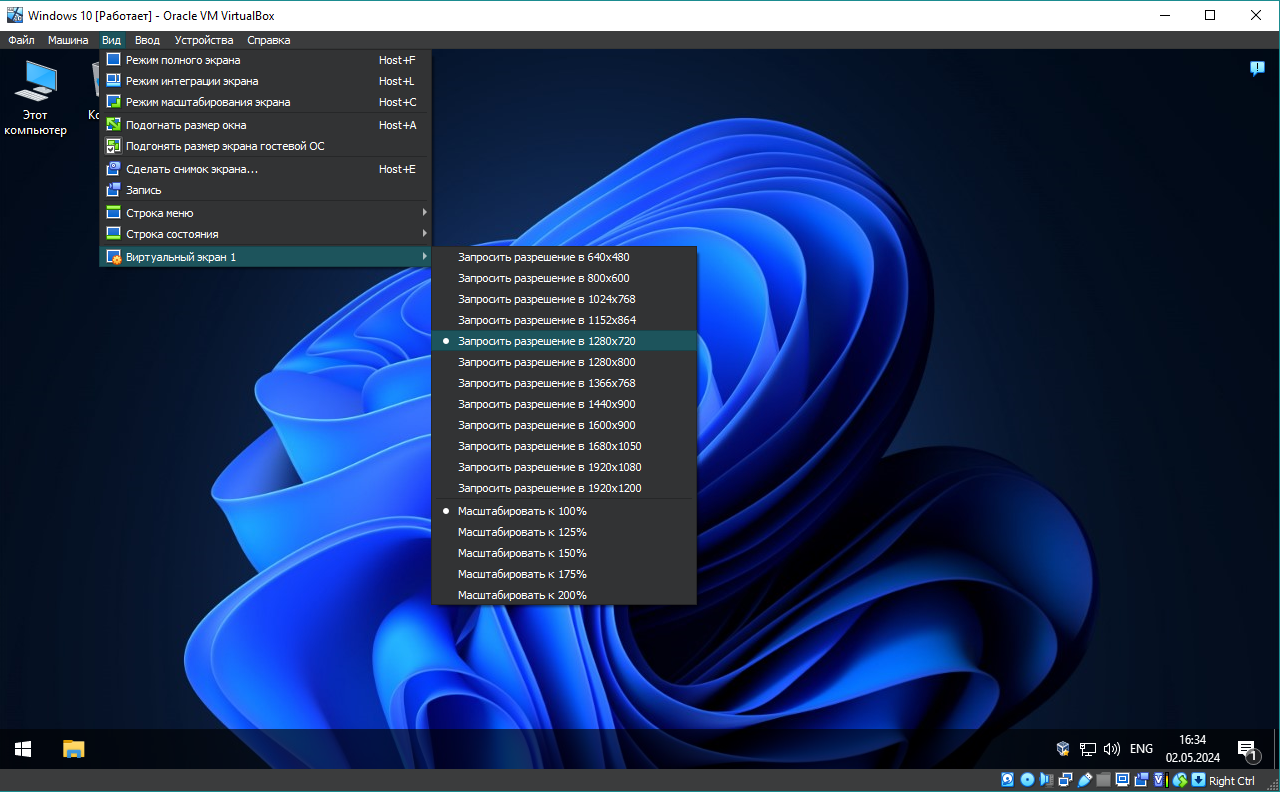
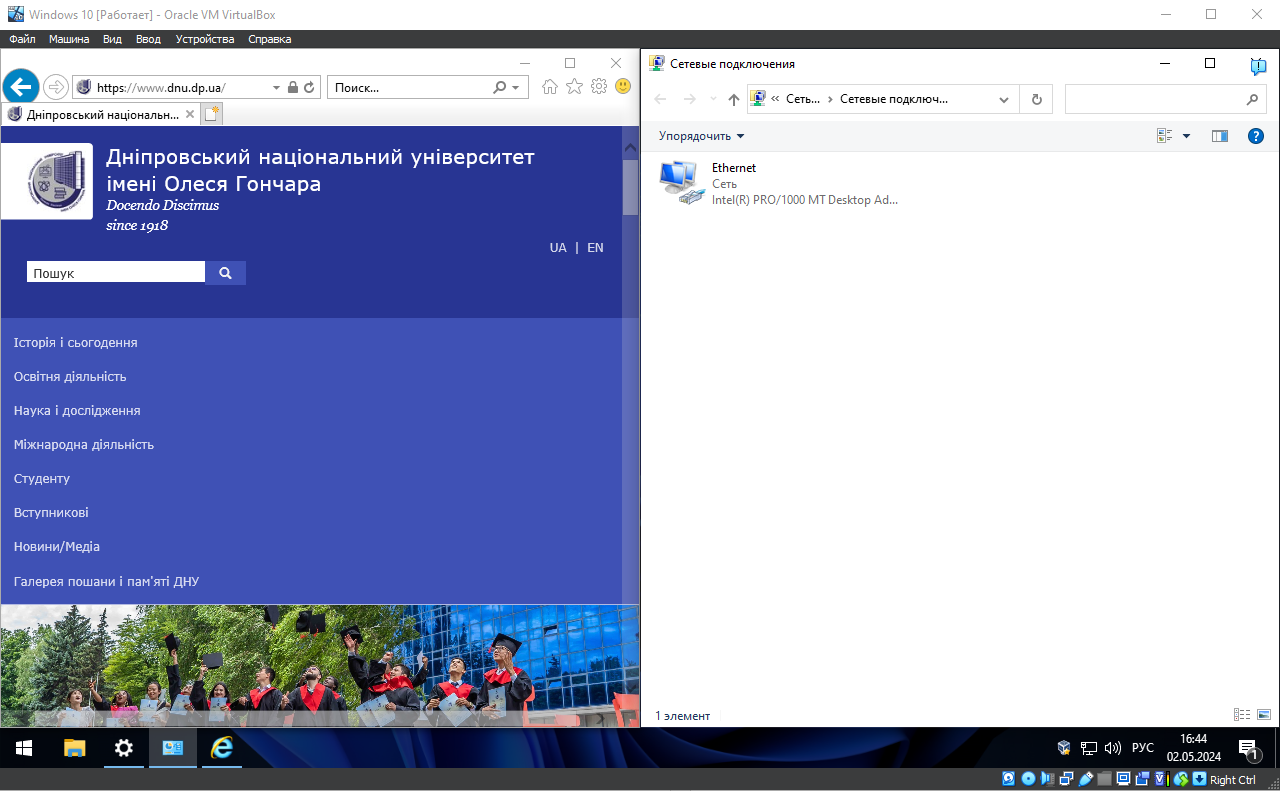


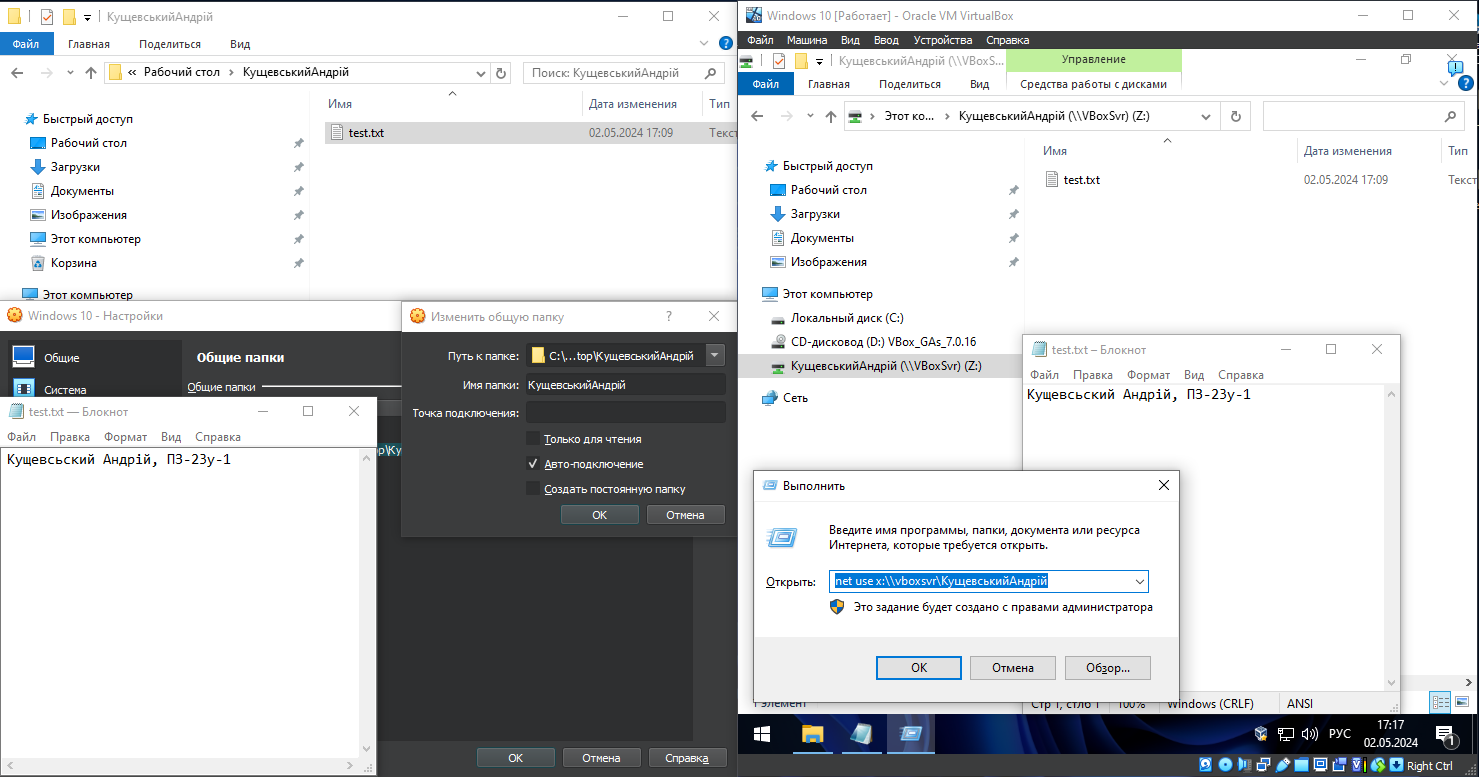
Рисунок 2.2 – Результат встановлення на віртуальну машину операційної системи Windows 10



Рисунок 2.3 – Результат встановлення гостьових доповнень на віртуальну машину

Рисунок 2.4 – Результат коректної зміни роздільної здатності екрану завдяки встановленню гостьових доповнень

Рисунок 2.5 – Налаштований доступ в Інтернет на віртуальній машині

Рисунок 2.6 – Налаштований спільний каталог між реальною і віртуальною ОС

1. Відмінності між VirtualBox та Windows Sandbox.

Windows Sandbox та VirtualBox - це обидва інструменти для віртуалізації, але вони мають дещо різні функції та використання. Ось деякі ключові відмінності між ними:

1. Інтеграція з операційною системою:

* Windows Sandbox - це вбудований інструмент у Windows, який дозволяє запускати ізольовані від основної ОС середовища для тестування програм або виконання сумнівних файлів. Він швидко створюється та використовується, але має обмежений функціонал порівняно з повноцінними віртуальними машинами;
* VirtualBox - це окремий продукт, який потрібно встановлювати окремо на операційну систему. Він надає більш широкий функціонал, включаючи можливість запускати різні операційні системи одночасно, налаштовувати ресурси віртуальних машин тощо.

1. Функціональність:

* Windows Sandbox: зазвичай використовується для швидкого створення ізольованого середовища для виконання сумнівних файлів або програм;
* VirtualBox: надає повноцінну віртуалізацію, що дозволяє вам запускати різні операційні системи, налаштовувати мережеві з'єднання, використовувати образи віртуальних машин тощо.

1. Налаштування:

* Windows Sandbox: має обмежені можливості налаштування, оскільки це вбудований інструмент;
* VirtualBox: надає широкий спектр налаштувань, таких як розмір оперативної пам'яті, обсяг жорсткого диска, кількість процесорних ядер тощо.

1. Доступність:

* Windows Sandbox: доступний тільки на Windows;
* VirtualBox: доступний для широкого спектру операційних систем, включаючи Windows, macOS, Linux та інші.

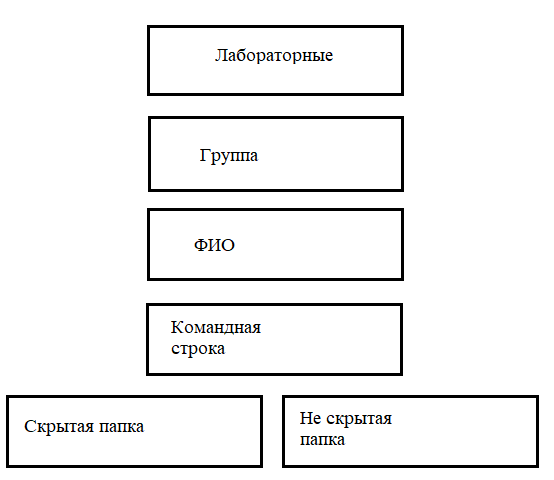
Лабораторна робота № 3

«Командна строка Windows. Batch скрипти»

1. Постановка завдання.

Частина 1

Відкрийте консоль і, використовуючи команди mkdir, створіть структуру каталогів:

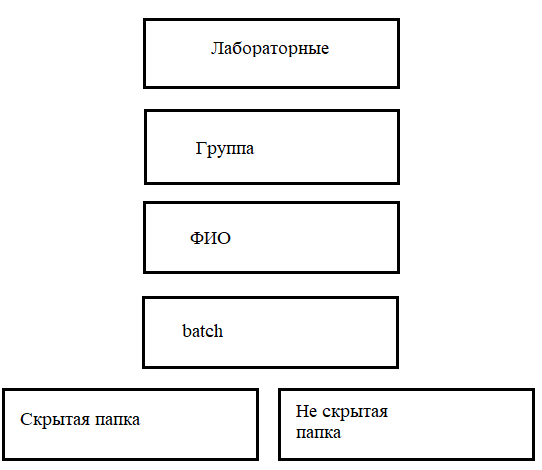


За допомогою команди ATTRIB зробіть каталог «Скрытая папка» прихованим.

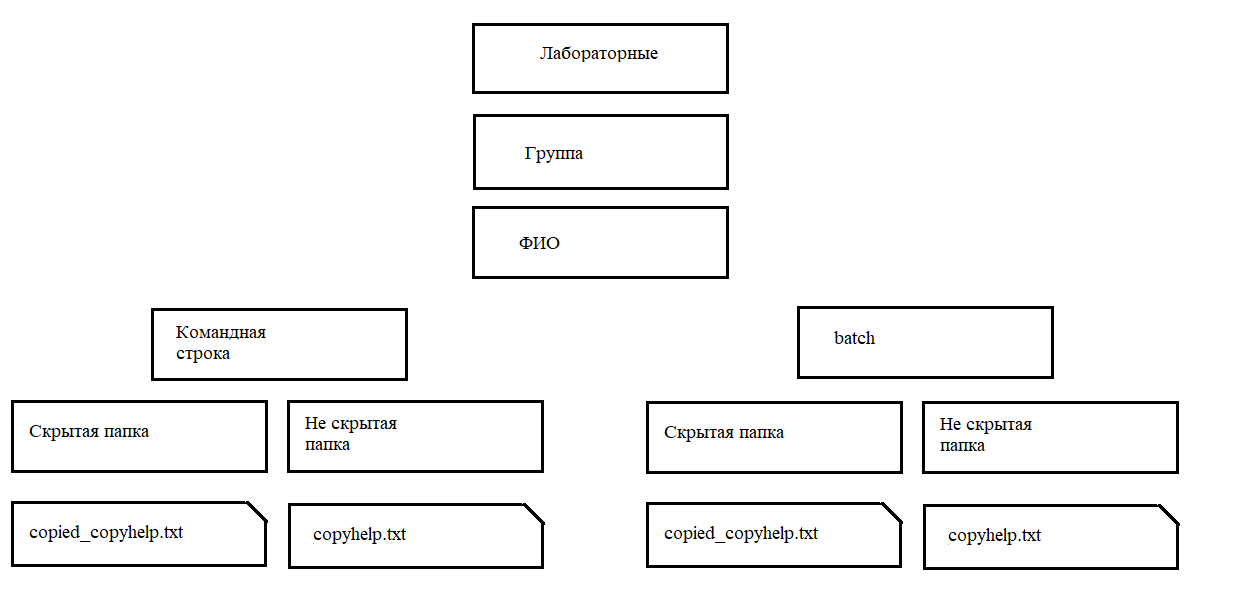
Спробуйте вивести на екран допомогу з використання команди xcopy. На консолі ви побачите список ключів командного рядка для використання цієї команди. Тепер спробуйте зробити те же саме, але використовуючи операцію >, спрямуйте вивід з консолі до файла copyhelp.txt, який знаходиться в каталозі «Не скрытая папка».

Використовуючи команду xcopy, скопіюйте файл copyhelp.txt до каталога «Скрытая папка», при цьому перейменувавши його в copied\_copyhelp.txt.

Створіть batch файл, який виконує ті ж самі операції, але в каталозі Лабораторные\Группа\ФИО\batch:



В результаті структура каталогів і файлів повинна виглядати так:



Створити окремий batch-файл, який відкриватиме файл зі звітом в новому вікні і очікуватиме на закриття вікна.

Частина 2

Використовуючи batch команди (за згодою викладача іншу скриптову мову) створити програму, яка виконує наступне:

Варіант 10. Знайти в каталозі підкаталог з заданим ім'ям.

1. Результат виконання завдання.

Результат виконання завдання наведено на рисунках 3.1 ­­– 3.8.

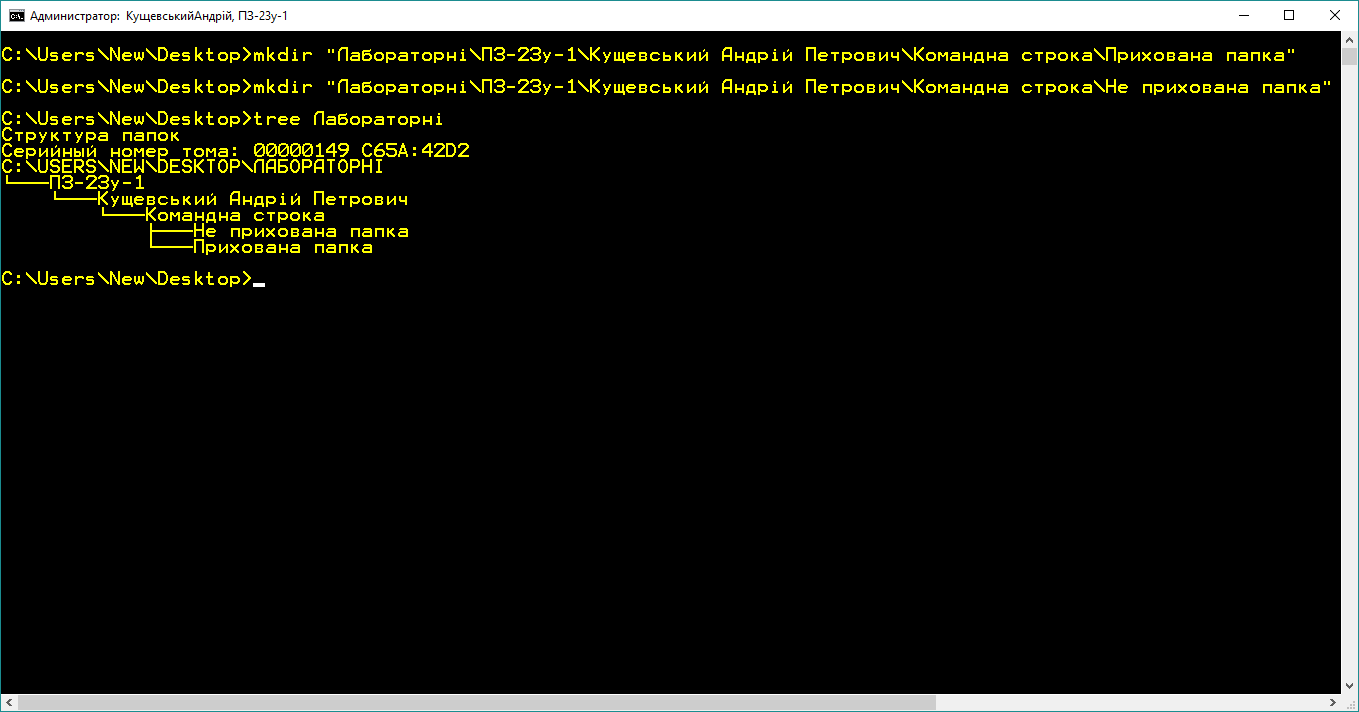


Рисунок 3.1 – Створена структура каталогів за допомогою mkdir

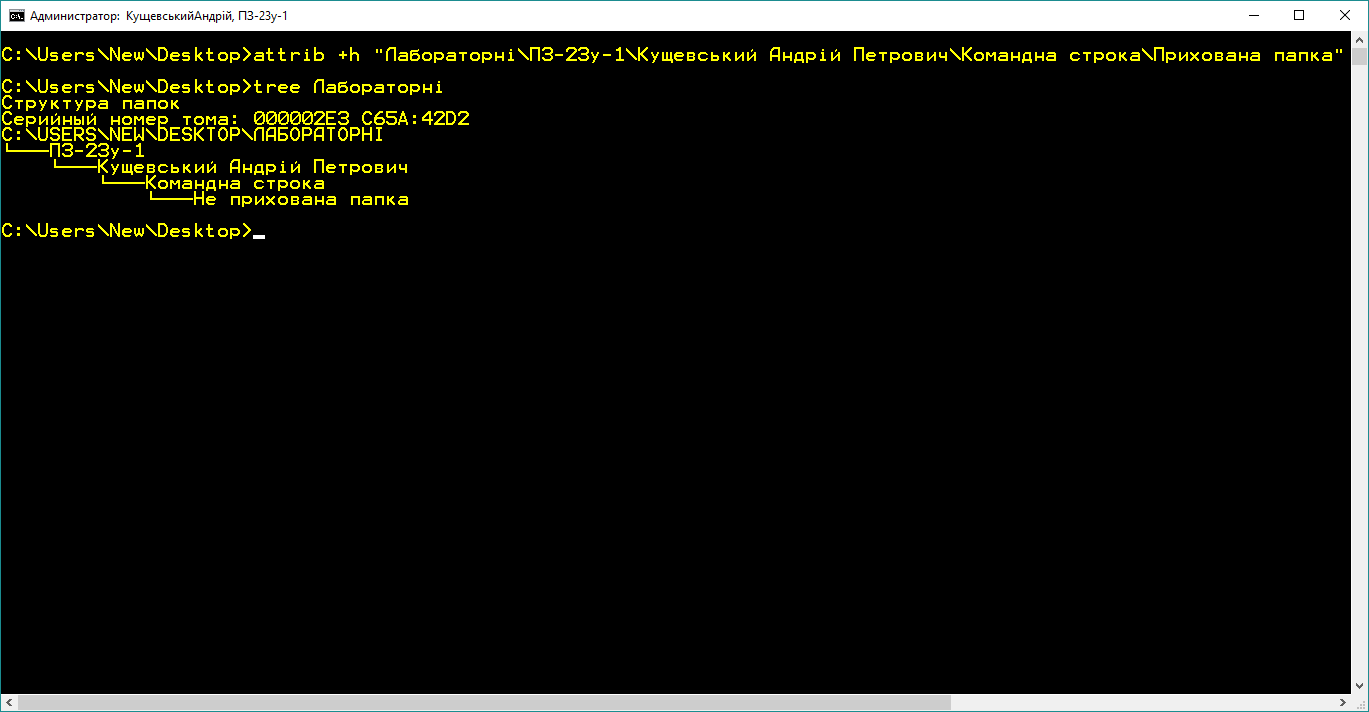


Рисунок 3.2 – За допомогою команди ATTRIB каталог «Прихована папка» зроблений прихованим



Рисунок 3.3 – Виведена на екран допомога з використання команди xcopy

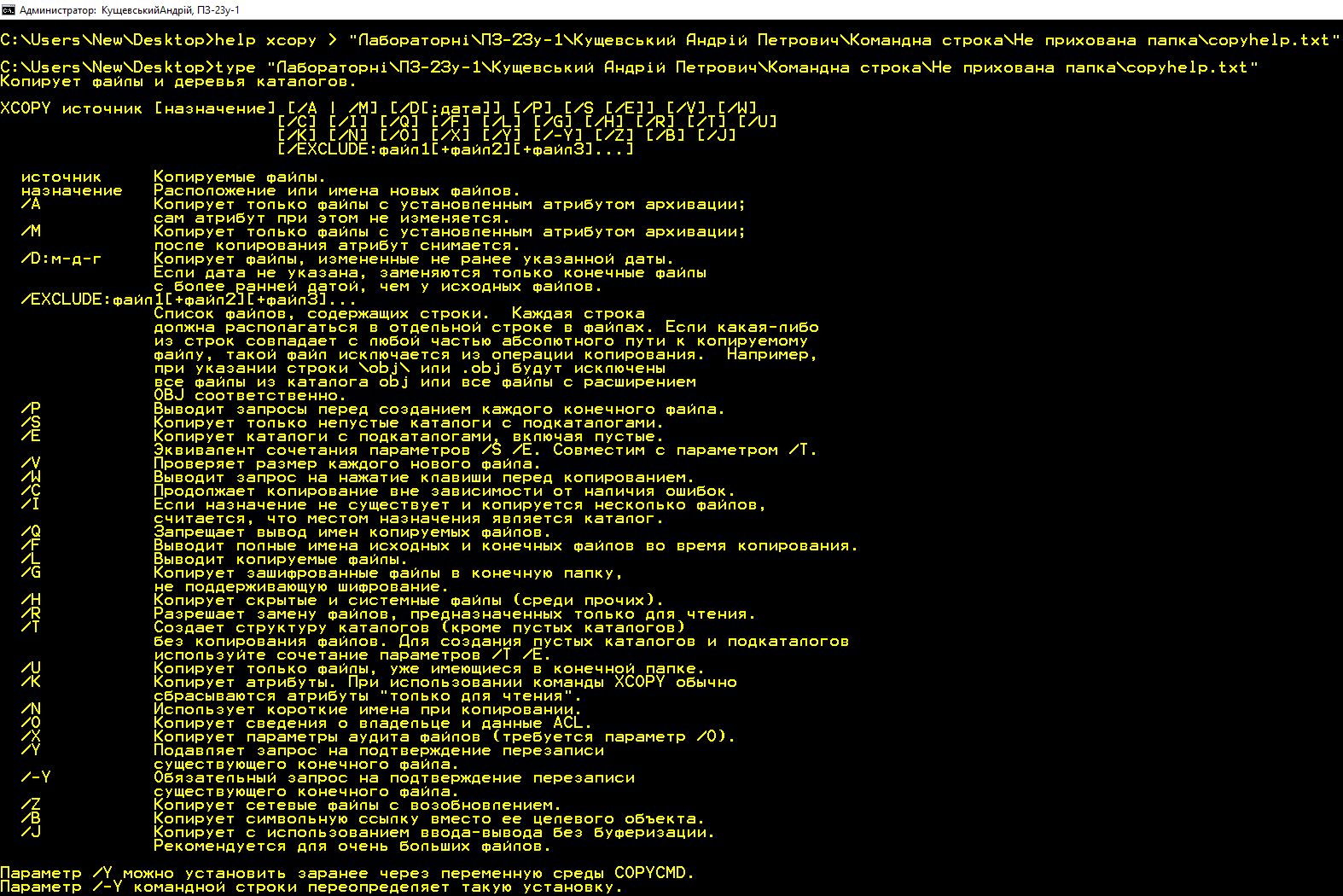


Рисунок 3.4 – Спрямований вивід з консолі до файла copyhelp.txt, який знаходиться в каталозі «Не прихована папка»



Рисунок 3.5 – Використовуючи команду xcopy, скопійований файл copyhelp.txt до каталога «Прихована папка», при цьому перейменувавши його в copied\_copyhelp.txt

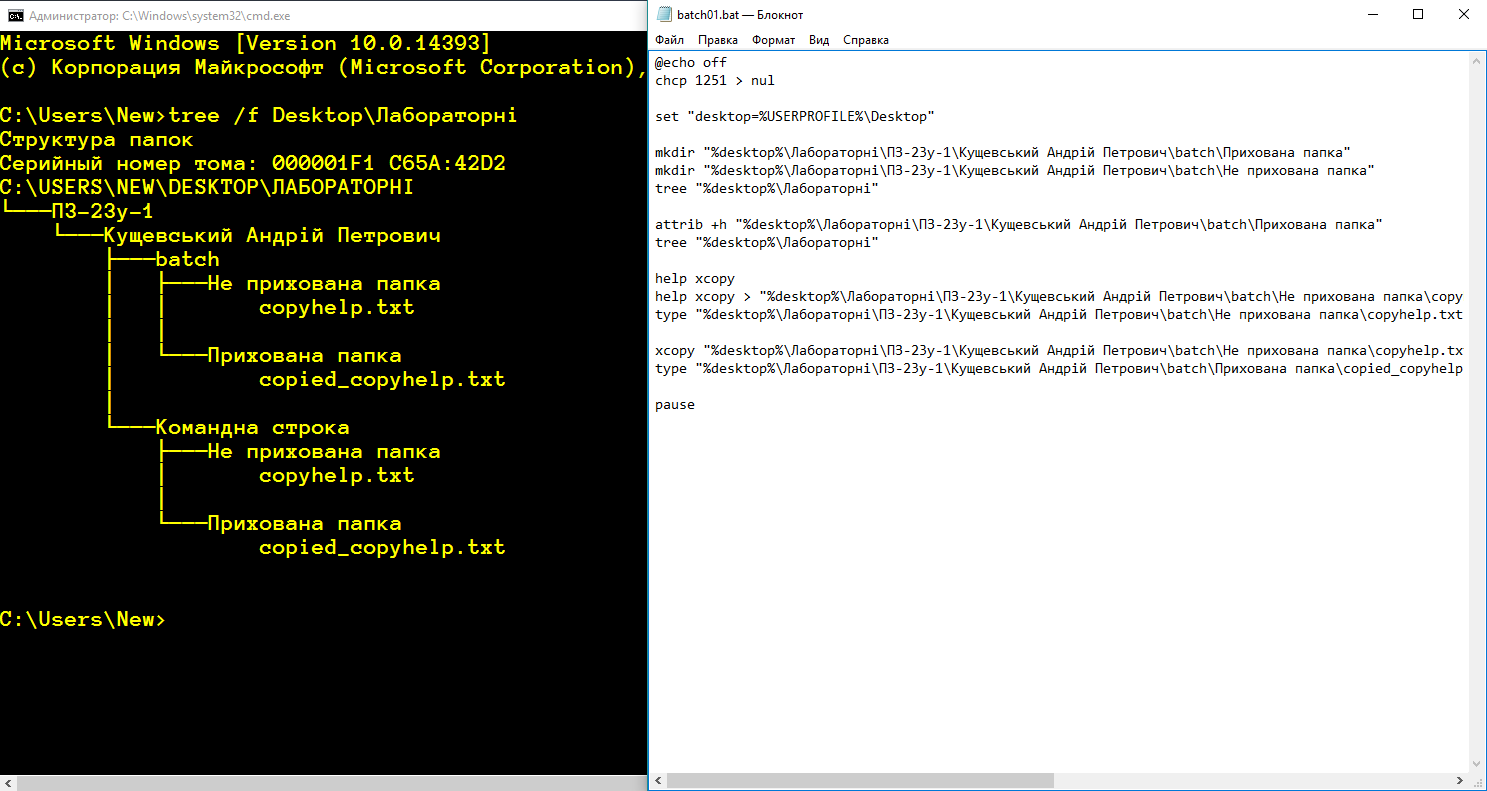


Рисунок 3.6 – Результат створення і виконання batch файлу, який виконує ті ж самі операції, але в каталозі «Лабораторні\Група\ПІБ\batch»

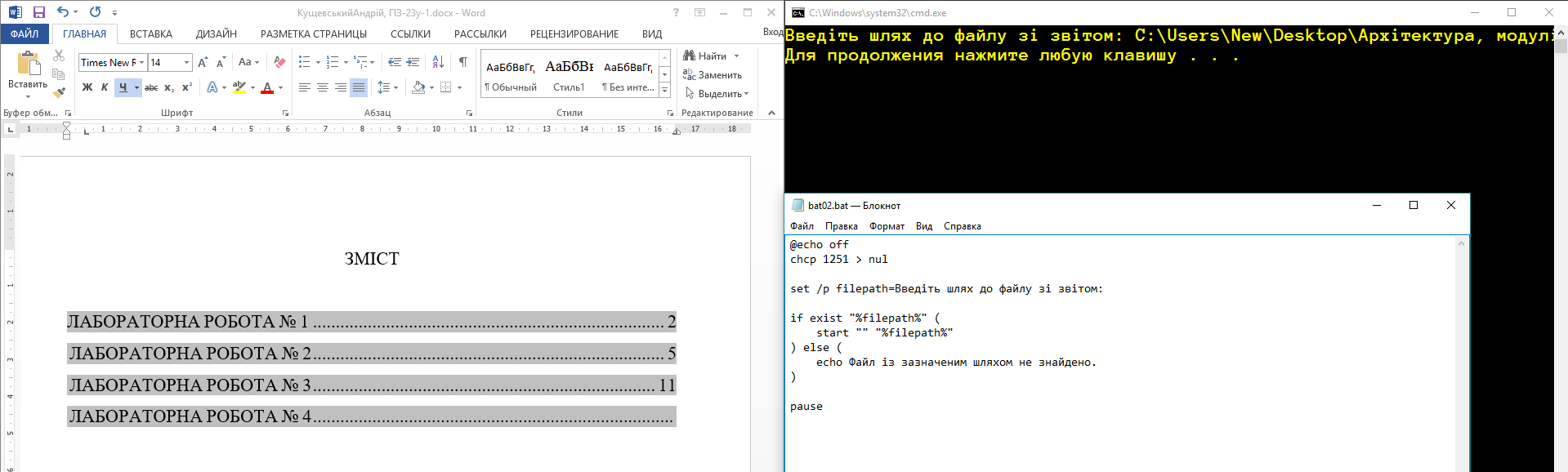


Рисунок 3.7 – Результат створення окремого batch-файлу, який відкриває файл зі звітом в новому вікні і очікуватиме на закриття вікна

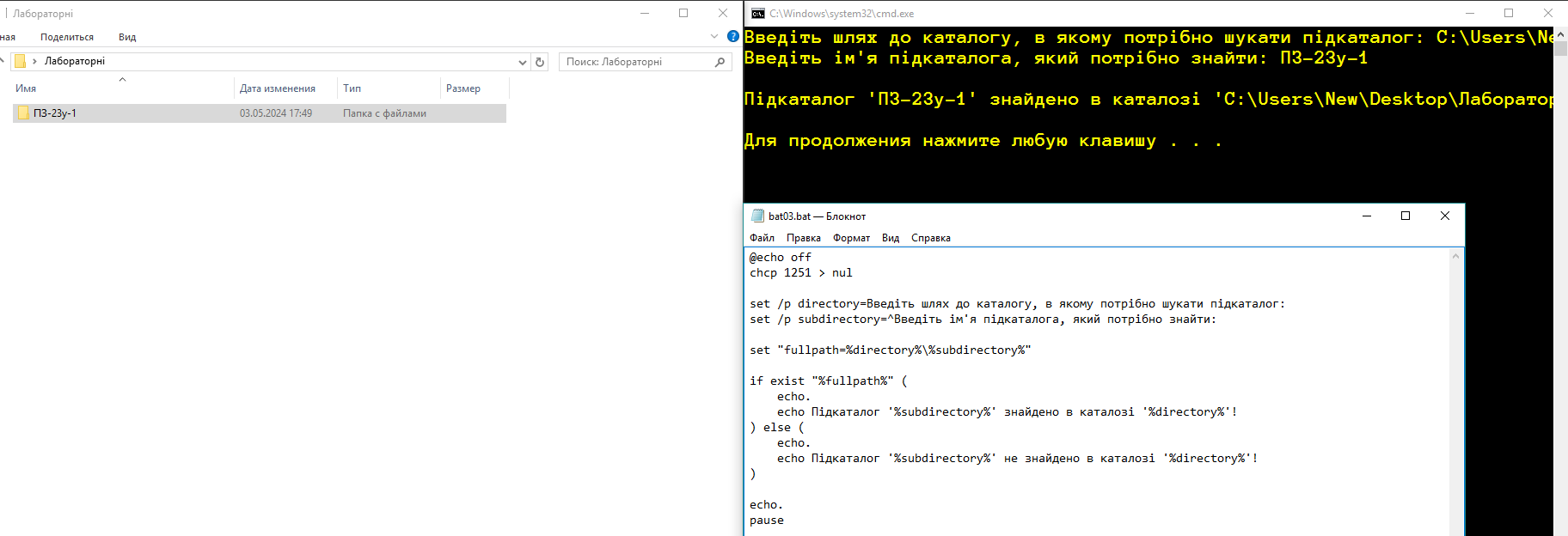


Рисунок 3.8 – Результат створення програми, використовуючи batch команди, яка виконує наступне: знайти в каталозі підкаталог з заданим ім'ям

1. Вихідний текст batch-файлів

Текст batch-файлів представлено в лістингах 3.1 – 1.3.

Лістинг 3.1 – bat01.bat

@echo off

chcp 1251 > nul

set "desktop=%USERPROFILE%\Desktop"

mkdir "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Прихована папка"

mkdir "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Не прихована папка"

tree "%desktop%\Лабораторні"

attrib +h "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Прихована папка"

tree "%desktop%\Лабораторні"

help xcopy

help xcopy > "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Не прихована папка\copyhelp.txt"

type "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Не прихована папка\copyhelp.txt"

xcopy "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Не прихована папка\copyhelp.txt" "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Прихована папка\copied\_copyhelp.txt"

type "%desktop%\Лабораторні\ПЗ-23у-1\Кущевський Андрій Петрович\batch\Прихована папка\copied\_copyhelp.txt"

echo.

pause

Лістинг 3.2 – bat02.bat

@echo off

chcp 1251 > nul

set /p filepath=Введіть шлях до файлу зі звітом:

if exist "%filepath%" (

start "" "%filepath%"

) else (

echo.

echo Файл із зазначеним шляхом не знайдено.

)

echo.

pause

Лістинг 3.3 – bat03.bat

@echo off

chcp 1251 > nul

set /p directory=Введіть шлях до каталогу, в якому потрібно шукати підкаталог:

set /p subdirectory=^Введіть ім'я підкаталога, який потрібно знайти:

set "fullpath=%directory%\%subdirectory%"

if exist "%fullpath%" (

echo.

echo Підкаталог '%subdirectory%' знайдено в каталозі '%directory%'!

) else (

echo.

echo Підкаталог '%subdirectory%' не знайдено в каталозі '%directory%'!

)

echo.

pause

Лабораторна робота № 4

«Робота з файловою системою у мовах програмування високого рівня»

1 Постановка завдання.

Використовуючи мову програмування високого рівня (C, C++, C#, іншу мову за згодою з викладачем, але не скриптові мови) створити програму, яка виконує наступне:

Варіант 10. Знайти в каталозі підкаталог з заданим ім'ям.

1. Блок-схема алгоритму рішення задачі.

Блок-схема алгоритму рішення задачі представлена на рисунку 4.1.

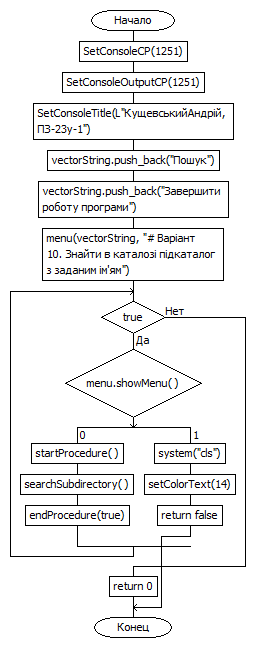


Рисунок 4.1 – Блок-схема рішення задачі

1. Вихідний текст програми.

Текст програми представлено в лістингах 4.1 – 4.3.

Лістинг 4.1 – main.cpp

#include "function.h"

#include "menu.h"

using namespace std;

wstring stringToWideString(const string& str)

{

wstring wideStr(str.begin(), str.end());

return wideStr;

}

bool searchSubdirectory()

{

string directoryName;

string subdirectoryName;

bool searchHiddenDirectories;

bool searchReadOnlyDirectories;

std::cout << "Введіть шлях до каталогу: ";

getline(cin, directoryName);

std::cout << "Введіть ім'я підкаталогу, який потрібно знайти: ";

getline(cin, subdirectoryName);

std::cout << std::endl;

std::cout << "Шукати підкаталоги з встановленим атрибутом 'Скритий'? (1 - Так, 0 - Ні): ";

std::cin >> searchHiddenDirectories;

std::cout << "Шукати підкаталоги з встановленим атрибутом 'Тільки для читання'? (1 - Так, 0 - Ні): ";

std::cin >> searchReadOnlyDirectories;

std::cout << std::endl;

string command = "tree /f " + directoryName;

system(command.c\_str());

string searchPath = directoryName + "\\\*";

if (!searchHiddenDirectories)

{

searchPath += " /A:-H";

}

if (!searchReadOnlyDirectories)

{

searchPath += " /A:R";

}

WIN32\_FIND\_DATAW findData;

wstring wSearchPath = stringToWideString(searchPath);

HANDLE hFind = FindFirstFileW(wSearchPath.c\_str(), &findData);

if (hFind != INVALID\_HANDLE\_VALUE)

{

do

{

if ((findData.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY) && wstring(findData.cFileName) == stringToWideString(subdirectoryName))

{

setColorText(10);

std::cout << "Підкаталог '" << subdirectoryName << "' знайдено в каталозі '" << directoryName << "'!" << std::endl;

setColorText(14);

FindClose(hFind);

return true;

}

} while (FindNextFileW(hFind, &findData) != 0);

FindClose(hFind);

}

setColorText(12);

std::cout << "Підкаталог '" << subdirectoryName << "' не знайдено в каталозі '" << directoryName << "'!" << std::endl;

setColorText(14);

return false;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

SetConsoleTitle(L"КущевськийАндрiй, ПЗ-23у-1");

std::vector<std::string> vectorString;

vectorString.push\_back("Пошук");

vectorString.push\_back("Завершити роботу програми");

Menu menu(vectorString, "# Варіант 10. Знайти в каталозі підкаталог з заданим ім'ям");

while (true)

{

switch (menu.showMenu())

{

case 0:

startProcedure();

searchSubdirectory();

endProcedure(true);

break;

case 1:

system("cls");

setColorText(14);

return false;

}

}

return 0;

}

Лістинг 4.2 – menu.h

#pragma once

#include "function.h"

constexpr auto UP = 72;

constexpr auto DOWN = 80;

constexpr auto ENTER = 13;

class Menu

{

protected:

std::vector<std::string> vectorString;

std::string title;

private:

void cursorStrMenu(std::string str, int color, bool flag)

{

std::string cursor = "> ";

setColorText(color);

if (flag)

{

str = cursor + str;

std::cout << str;

}

else

{

std::cout << str;

for (int i = 0; i < 100; ++i)

std::cout << " ";

}

}

public:

Menu(std::vector<std::string> vectorString, std::string title) { this->vectorString = vectorString; this->title = title; }

~Menu() {}

unsigned int showMenu()

{

short x = 0;

short y = 0;

int key = 0;

int activeMenu = 0;

unsigned int count = 0;

while (true)

{

gotoxy(10, 0);

setColorText(14);

std::cout << title << std::endl << std::endl;

setCursorState(false);

x = 10;

y = 3;

for (int i = 0; i < this->vectorString.size(); ++i)

{

if (i == activeMenu)

{

gotoxy(x, ++y);

cursorStrMenu(vectorString[i], 10, true);

}

else

{

gotoxy(x, ++y);

cursorStrMenu(vectorString[i], 14, false);

}

}

key = \_getch();

switch (key)

{

case UP:

(activeMenu > 0) ? --activeMenu : activeMenu = this->vectorString.size() - 1;

break;

case DOWN:

(activeMenu < this->vectorString.size() - 1) ? ++activeMenu : activeMenu = 0;

break;

case ENTER:

return activeMenu;

}

}

}

};

Лістинг 4.3 – function.h

#pragma once

#include <iostream>

#include <iomanip>

#include <conio.h>

#include <vector>

#include <string>

#include <regex>

#include <ctime>

#include <Windows.h>

using namespace std;

void clearCinBuff()

{

std::cin.clear();

std::cin.sync();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

}

void setCursorState(bool flag)

{

\_CONSOLE\_CURSOR\_INFO info;

info.bVisible = flag;

info.dwSize = 20;

SetConsoleCursorInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &info);

}

void setColorText(short item)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), item);

}

void setErrorColor()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 12);

}

void setDefaultColor()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

}

void startProcedure()

{

system("cls");

clearCinBuff();

setCursorState(true);

setColorText(14);

}

void endProcedure(bool isAfterInput)

{

clearCinBuff();

if (isAfterInput)

std::cout << std::endl;

else

std::cout << std::endl << std::endl;

system("pause");

setCursorState(false);

setColorText(14);

system("cls");

}

void gotoxy(short x, short y)

{

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), { x, y });

}

1. Опис тестових прикладів.

Результат виконання завдання наведено на рисунках 4.1 ­­– 4.4.

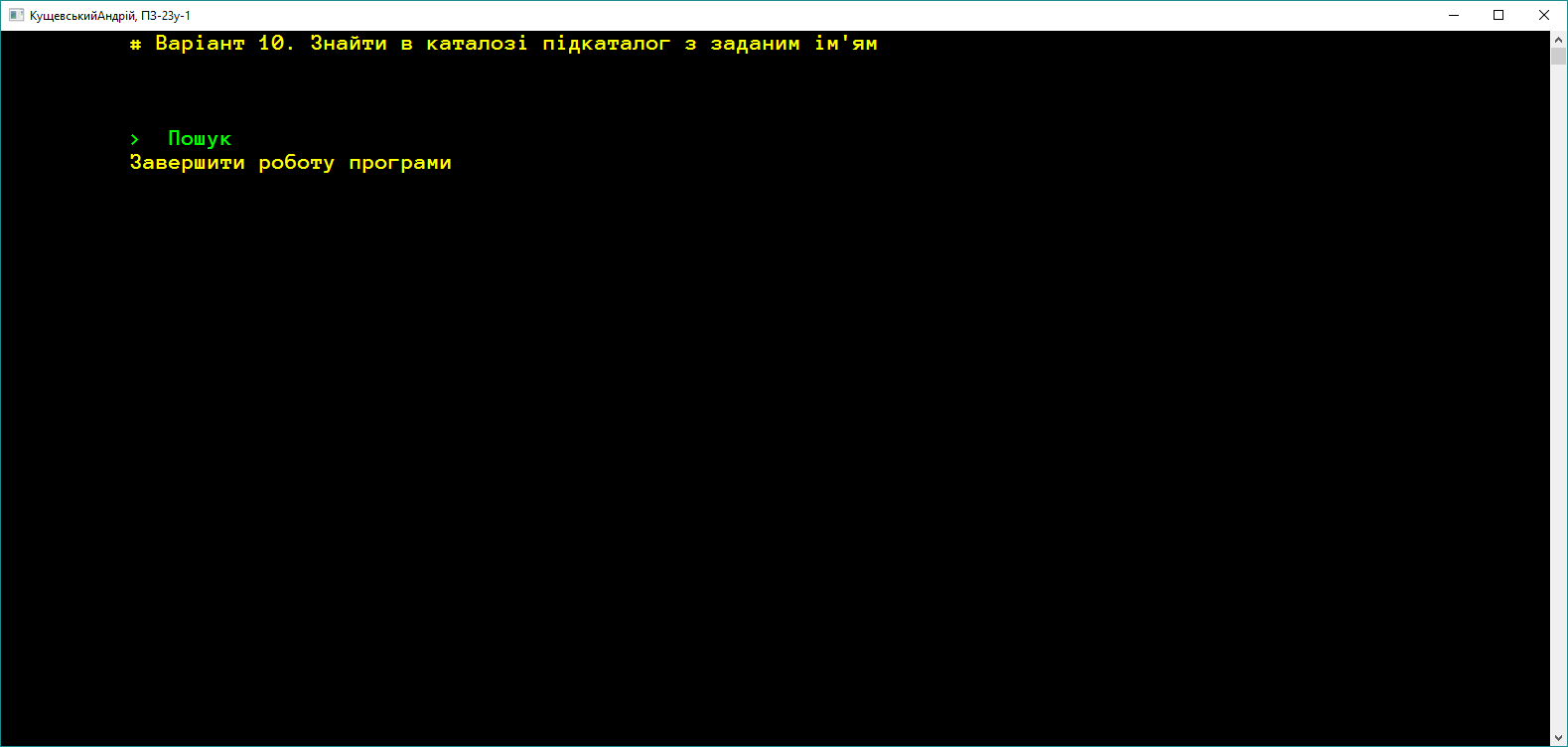


Рисунок 4.1 – Головне меню програми

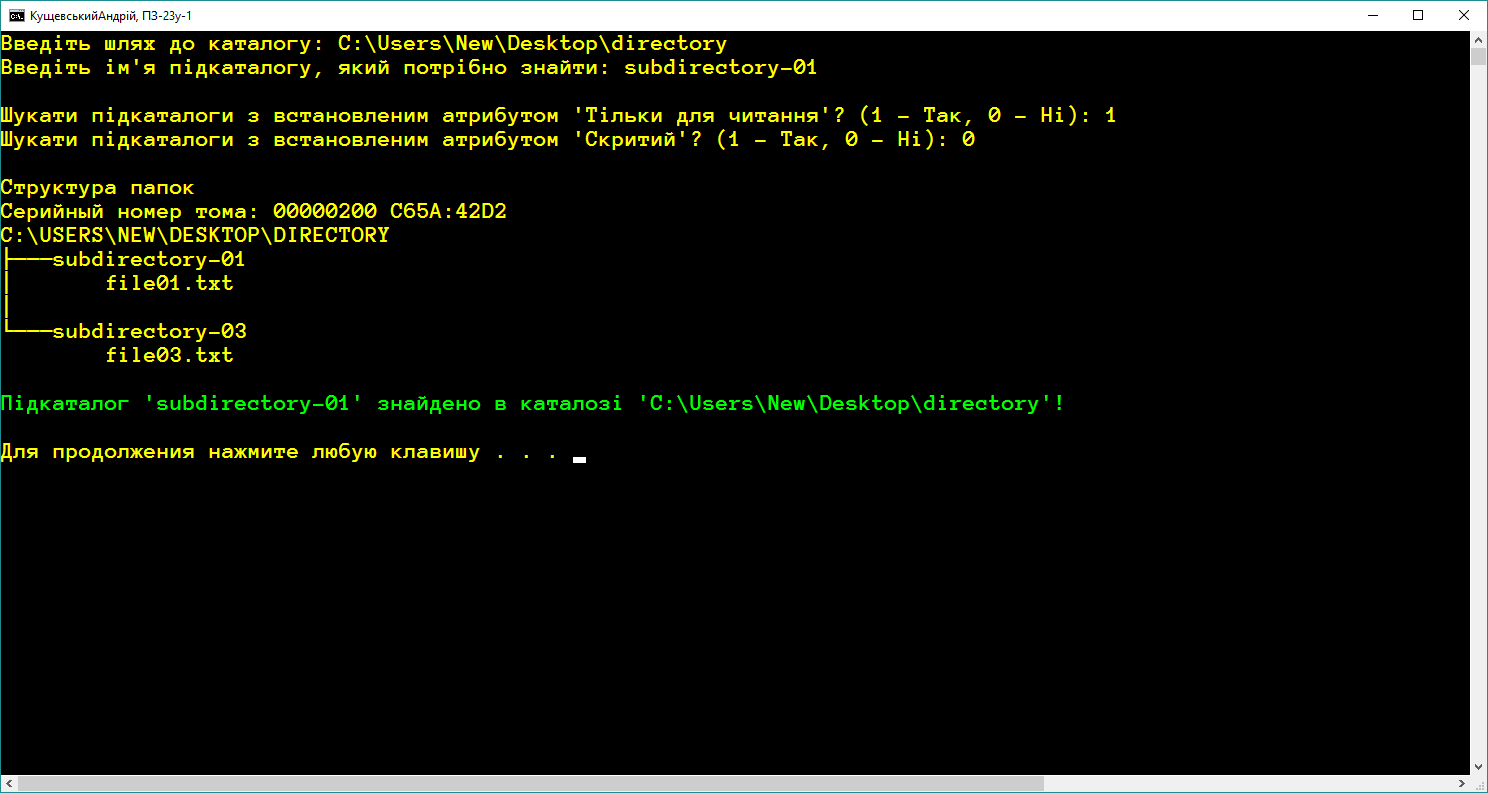


Рисунок 4.2 – Результат пошуку підкаталогу з атрибутом «Тільки для читання»

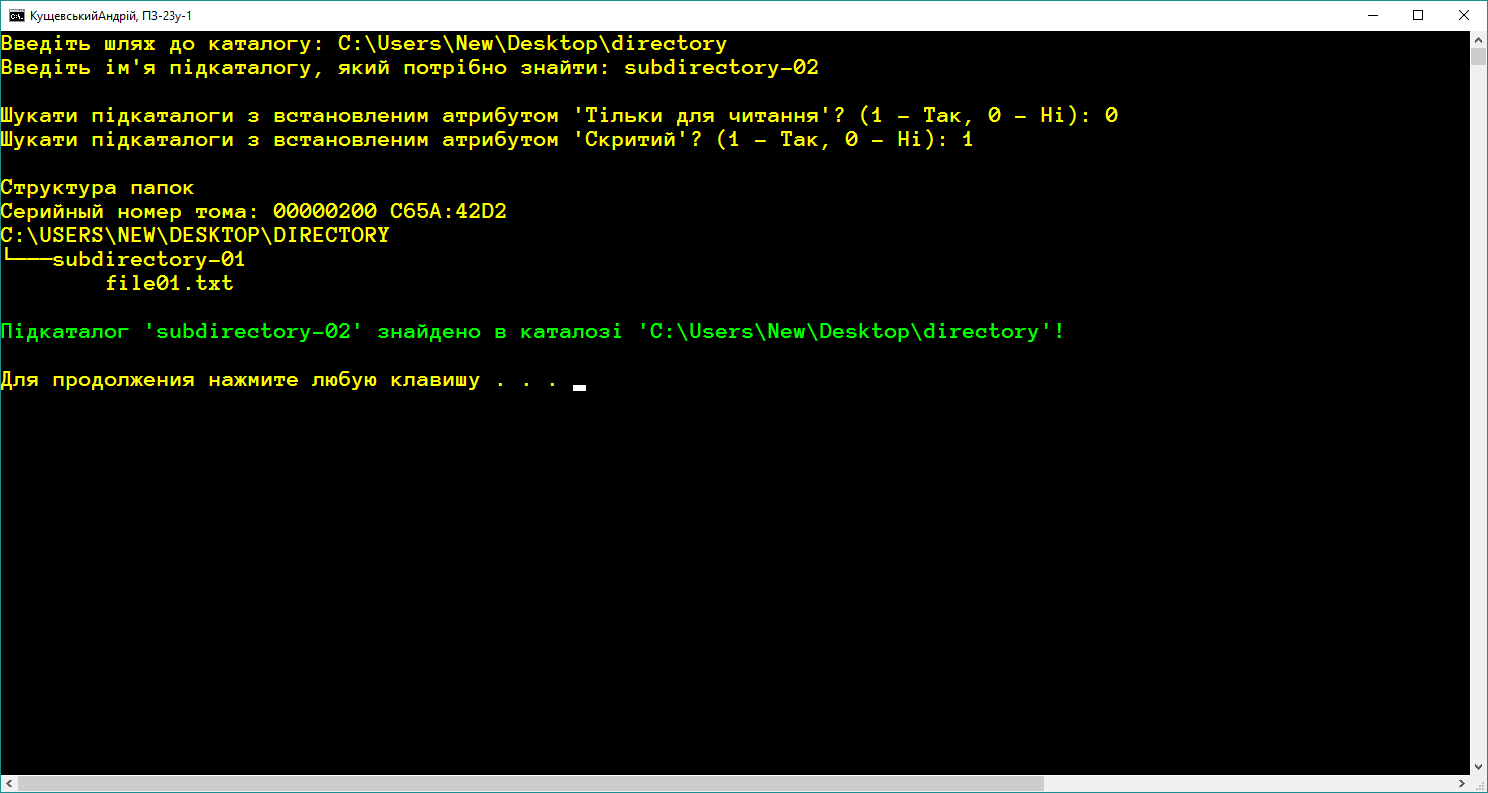


Рисунок 4.3 – Результат пошуку підкаталогу з атрибутом «Скритий»

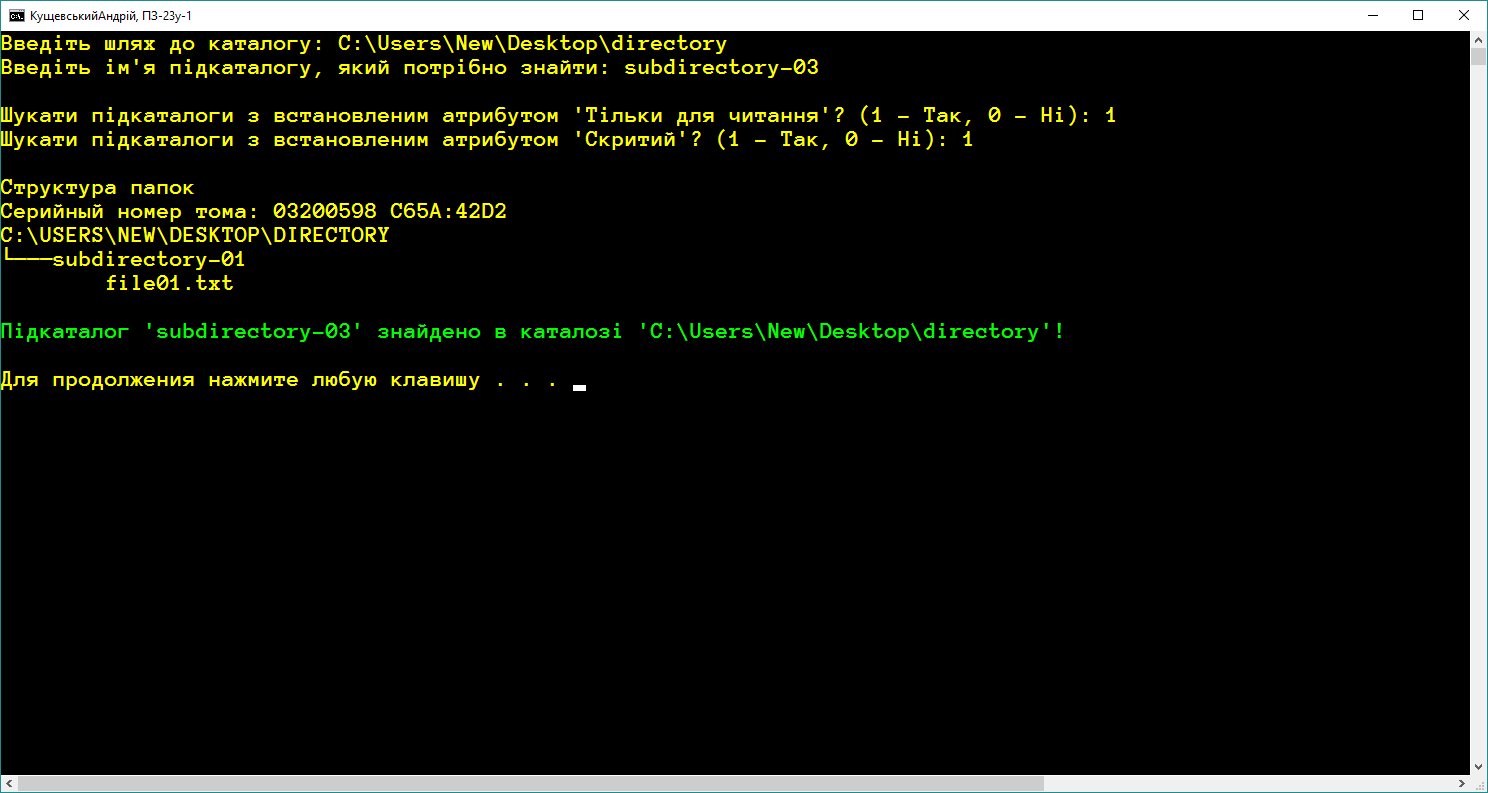


Рисунок 4.4 – Результат пошуку підкаталогу з атрибутом «Скритий» та «Тільки для Читання»