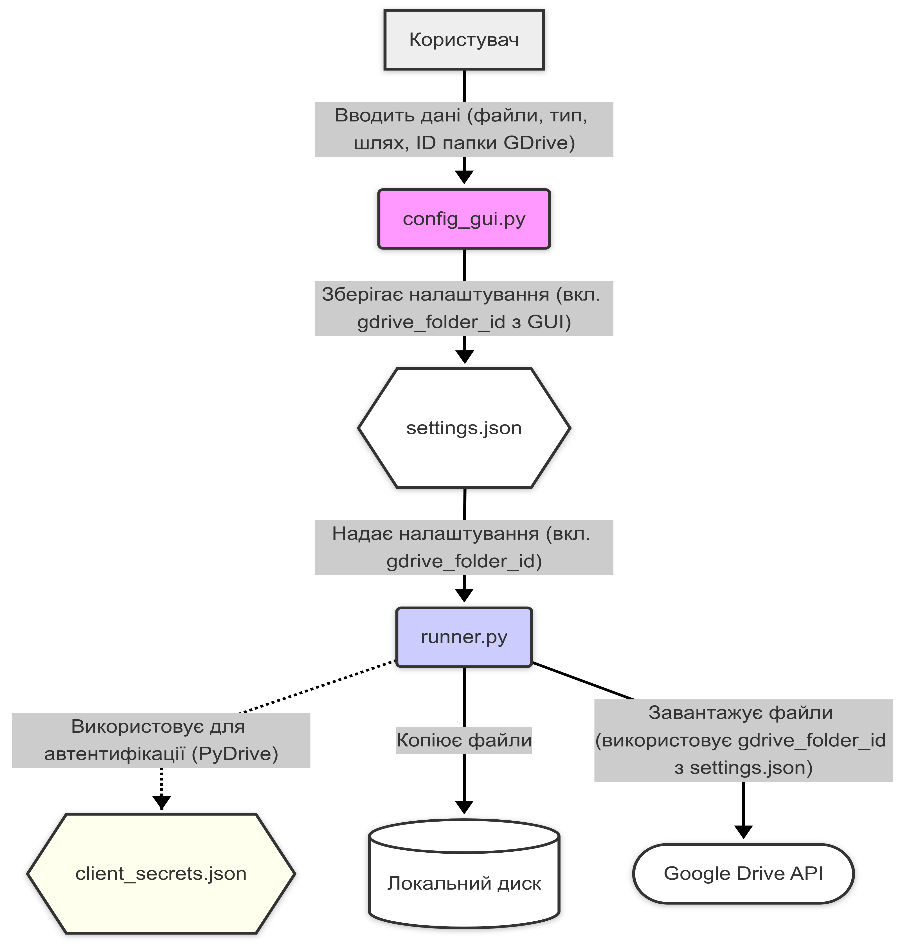
**РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕЗЕРВНОГО КОПІЮВАННЯ ДАНИХ**

**3.1 Загальна архітектура розробленої системи**

Розроблена програмна система призначена для автоматизації процесу резервного копіювання файлів користувача. Вона складається з двох основних модулів: графічного інтерфейсу користувача для налаштування параметрів резервного копіювання (config\_gui.py) та виконавчого модуля, що безпосередньо здійснює процес копіювання (runner.py). Конфігураційні дані, включаючи обрані файли, тип сховища, шлях до локальної папки та ідентифікатор папки Google Drive (який тепер вводиться користувачем через GUI), зберігаються у файлі settings.json. Для автентифікації у Google Drive використовується файл client\_secrets.json, який обробляється бібліотекою PyDrive.

**Основні функціональні можливості системи:**

* Вибір файлів для резервного копіювання.
* Вибір місця збереження резервних копій: локальний диск або хмарне сховище Google Drive.
* Введення ідентифікатора цільової папки на Google Drive через графічний інтерфейс.
* Збереження та завантаження налаштувань користувача.
* Виконання резервного копіювання згідно зі збереженими налаштуваннями.

**

*Діаграма 1* **Загальна архітектура системи резервного копіювання**

**Користувач** взаємодіє з config\_gui.py для налаштування параметрів, включаючи введення ID папки Google Drive.

Файл config\_gui.py зберігає всі обрані налаштування, включаючи ID папки, у файл settings.json.

Файл runner.py читає налаштування, включаючи ID папки, з settings.json для виконання операції резервного копіювання.

Для автентифікації з Google Drive API, runner.py (через бібліотеку PyDrive) неявно використовує файл client\_secrets.json.

Резервні копії зберігаються або на **Локальний диск**, або завантажуються на **Google Drive** у папку, ID якої вказав користувач.

**3.2 Модуль конфігурації (config\_gui.py)**

Модуль config\_gui.py надає користувачеві графічний інтерфейс для налаштування параметрів резервного копіювання. Ключовою зміною в цій версії є додавання можливості для користувача вказувати ідентифікатор (ID) цільової папки на Google Drive безпосередньо в інтерфейсі програми. Цей модуль використовує бібліотеку tkinter.

**Файл: config\_gui.py (версія з полем введення ID папки)**

Python

# Файл: config\_gui.py

# GUI-програма для створення та збереження налаштувань резервного копіювання

# з полем для введення ID папки Google Drive

import tkinter as tk

from tkinter import filedialog, messagebox

import json

import os

SETTINGS\_FILE = "settings.json"

class ConfigApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Налаштування резервного копіювання")

self.root.geometry("500x400") # Трохи збільшив висоту для нового поля

self.files = []

self.destination = ""

self.backup\_to\_drive = tk.BooleanVar()

self.gdrive\_folder\_id\_var = tk.StringVar() # Змінна для зберігання ID папки з поля вводу

# Завантаження попередніх налаштувань, якщо вони є

self.load\_initial\_settings()

tk.Label(root, text="Оберіть файли для резервного копіювання:").pack(pady=5)

tk.Button(root, text="Обрати файли", command=self.select\_files).pack()

tk.Label(root, text="Куди зберігати резервну копію?").pack(pady=10)

# Прив'язуємо команду для відображення/приховування поля ID

tk.Radiobutton(root, text="На диск", variable=self.backup\_to\_drive, value=False, command=self.toggle\_gdrive\_id\_input).pack()

tk.Radiobutton(root, text="У хмару (Google Drive)", variable=self.backup\_to\_drive, value=True, command=self.toggle\_gdrive\_id\_input).pack()

# Створюємо рамку для поля вводу ID папки, щоб керувати її видимістю

self.gdrive\_id\_frame = tk.Frame(root)

# Не пакуємо її одразу, видимість буде контролюватися toggle\_gdrive\_id\_input

tk.Label(self.gdrive\_id\_frame, text="ID папки Google Drive:").pack(side=tk.LEFT, padx=5)

self.gdrive\_id\_entry = tk.Entry(self.gdrive\_id\_frame, textvariable=self.gdrive\_folder\_id\_var, width=40)

self.gdrive\_id\_entry.pack(side=tk.LEFT)

# Розміщуємо рамку після радіокнопок

self.gdrive\_id\_frame.pack(pady=5)

tk.Button(root, text="Обрати папку призначення (для локального диска)", command=self.select\_destination).pack(pady=5)

tk.Button(root, text="Зберегти налаштування", command=self.save\_settings, bg="green", fg="white").pack(pady=15)

self.status = tk.Label(root, text="")

self.status.pack()

# Початкове налаштування видимості поля ID

self.toggle\_gdrive\_id\_input()

def load\_initial\_settings(self):

"""Завантажує ID папки Google Drive з settings.json при запуску, якщо файл існує."""

if os.path.exists(SETTINGS\_FILE):

try:

with open(SETTINGS\_FILE, "r", encoding="utf-8") as f:

settings = json.load(f)

# Встановлюємо значення для StringVar, яке автоматично оновить поле Entry

self.gdrive\_folder\_id\_var.set(settings.get("gdrive\_folder\_id", ""))

# Встановлюємо значення для backup\_to\_drive, щоб радіокнопка була коректною

self.backup\_to\_drive.set(settings.get("backup\_to\_drive", False))

# Зберігаємо інші налаштування, якщо потрібно їх використовувати при старті

self.files = settings.get("files", [])

self.destination = settings.get("destination", "")

except json.JSONDecodeError:

print(f"Попередження: Не вдалося розпарсити {SETTINGS\_FILE}. Будуть використані значення за замовчуванням.")

except Exception as e:

print(f"Попередження: Помилка при завантаженні початкових налаштувань: {e}")

def toggle\_gdrive\_id\_input(self):

"""Показує або приховує поле для введення ID папки Google Drive."""

if self.backup\_to\_drive.get(): # Якщо обрано Google Drive

self.gdrive\_id\_frame.pack(pady=5) # Показуємо рамку з полем вводу

else:

self.gdrive\_id\_frame.pack\_forget() # Приховуємо рамку

def select\_files(self):

self.files = list(filedialog.askopenfilenames(title="Оберіть файли"))

if self.files:

self.status.config(text=f"Обрано {len(self.files)} файлів.")

else:

self.status.config(text="Файли не обрано.")

def select\_destination(self):

self.destination = filedialog.askdirectory(title="Оберіть папку")

if self.destination:

self.status.config(text=f"Папка призначення: {self.destination}")

else:

self.status.config(text="Папку призначення не обрано.")

def save\_settings(self):

if not self.files:

messagebox.showerror("Помилка", "Не вибрано файлів.")

return

gdrive\_id = self.gdrive\_folder\_id\_var.get().strip() # Отримуємо ID з поля вводу та видаляємо зайві пробіли

if self.backup\_to\_drive.get(): # Якщо обрано Google Drive

if not gdrive\_id:

messagebox.showerror("Помилка", "Вкажіть ID папки Google Drive.")

return

elif not self.destination: # Якщо обрано локальний диск, але не вказано папку

messagebox.showerror("Помилка", "Вкажіть папку призначення для локального збереження.")

return

settings = {

"files": self.files,

"backup\_to\_drive": self.backup\_to\_drive.get(),

"destination": self.destination if not self.backup\_to\_drive.get() else "", # Зберігаємо папку призначення тільки для локального бекапу

"gdrive\_folder\_id": gdrive\_id if self.backup\_to\_drive.get() else "" # Зберігаємо ID тільки для Google Drive

}

try:

with open(SETTINGS\_FILE, "w", encoding="utf-8") as f:

json.dump(settings, f, indent=4)

self.status.config(text="Налаштування збережено.")

messagebox.showinfo("Готово", "Налаштування збережено!")

except Exception as e:

messagebox.showerror("Помилка збереження", f"Не вдалося зберегти налаштування: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = ConfigApp(root)

root.mainloop()

**3.2.1 Імпорт необхідних бібліотек та глобальні налаштування**

Імпорт бібліотек (tkinter, filedialog, messagebox, json, os) залишається таким же, як і в попередній версії. SETTINGS\_FILE = "settings.json": визначає ім'я файлу для збереження налаштувань.

**3.2.2 Клас ConfigApp**

Основна логіка графічного інтерфейсу.

**3.2.2.1 Конструктор \_\_init\_\_(self, root)**

Конструктор ініціалізує головне вікно та його компоненти.

* self.root.geometry("500x400"): Розмір вікна збільшено для розміщення нового поля вводу.
* self.gdrive\_folder\_id\_var = tk.StringVar(): Створюється спеціальна змінна tkinter типу StringVar. Ця змінна буде пов'язана з полем введення для ID папки Google Drive, дозволяючи легко отримувати та встановлювати його значення.
* self.load\_initial\_settings(): Викликається новий метод для завантаження раніше збережених налаштувань (включаючи ID папки) при запуску програми.

Далі створюються та розміщуються елементи інтерфейсу:

* Елементи для вибору файлів та папки призначення залишаються без змін.
* Радіокнопки для вибору типу сховища ("На диск" або "У хмару (Google Drive)") тепер мають параметр command=self.toggle\_gdrive\_id\_input. Це означає, що при зміні вибору радіокнопки буде викликатися метод toggle\_gdrive\_id\_input.
* **Поле для введення ID папки Google Drive:**

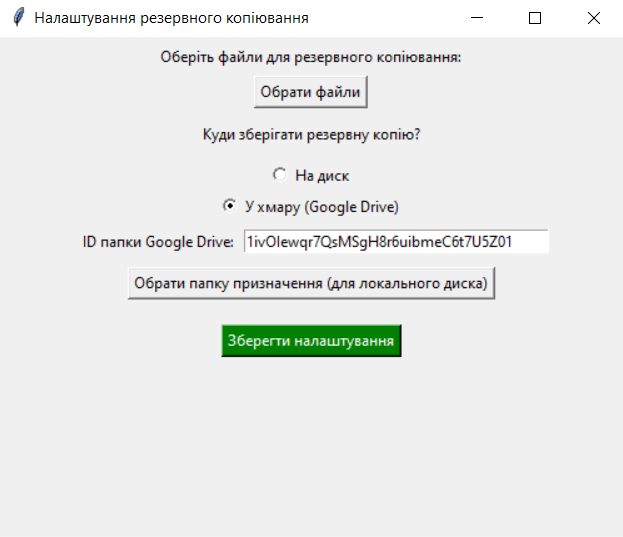
self.gdrive\_id\_frame = tk.Frame(root): Створюється об'єкт tk.Frame. Рамка використовується як контейнер для мітки та поля введення ID папки, що дозволяє керувати їх видимістю як єдиним цілим.

tk.Label(self.gdrive\_id\_frame, text="ID папки Google Drive:").pack(side=tk.LEFT, padx=5): Текстова мітка "ID папки Google Drive:", розміщена всередині рамки.

self.gdrive\_id\_entry = tk.Entry(self.gdrive\_id\_frame, textvariable=self.gdrive\_folder\_id\_var, width=40): Створюється поле введення (tk.Entry). Параметр textvariable пов'язує це поле зі змінною self.gdrive\_folder\_id\_var. Будь-які зміни в полі введення автоматично оновлюватимуть значення self.gdrive\_folder\_id\_var, і навпаки. width=40 встановлює ширину поля.

self.gdrive\_id\_frame.pack(pady=5): Рамка з міткою та полем вводу розміщується у головному вікні.

self.toggle\_gdrive\_id\_input(): Викликається наприкінці конструктора для встановлення початкової видимості поля введення ID папки залежно від стану радіокнопок (який міг бути завантажений з settings.json).

**

Зображення 2 Графічний інтерфейс модуля конфігурації

**3.2.2.2 Метод load\_initial\_settings(self)**

Цей новий метод відповідає за завантаження налаштувань з файлу settings.json при запуску програми.

Перевіряє, чи існує файл SETTINGS\_FILE.

Якщо файл існує, він читається, і дані JSON десеріалізуються.

self.gdrive\_folder\_id\_var.set(settings.get("gdrive\_folder\_id", "")): Значення для ID папки Google Drive (якщо воно є у файлі налаштувань) встановлюється у змінну self.gdrive\_folder\_id\_var. Це автоматично відобразить ID у відповідному полі введення. Якщо ключ gdrive\_folder\_id відсутній, встановлюється порожній рядок.

self.backup\_to\_drive.set(settings.get("backup\_to\_drive", False)): Встановлюється попередньо обраний стан радіокнопки.

Також завантажуються списки файлів (self.files) та шлях до локальної папки (self.destination).

Обробляються можливі помилки при читанні або парсингу файлу.

**3.2.2.3 Метод toggle\_gdrive\_id\_input(self)**

Цей метод керує видимістю рамки (self.gdrive\_id\_frame), що містить поле для введення ID папки Google Drive.

if self.backup\_to\_drive.get():: Якщо обрано опцію "У хмару (Google Drive)" (тобто self.backup\_to\_drive має значення True):

self.gdrive\_id\_frame.pack(pady=5): Рамка (разом з міткою та полем введення) відображається у вікні.

else:: Якщо обрано опцію "На диск":

self.gdrive\_id\_frame.pack\_forget(): Рамка приховується.

**3.2.2.4 Методи select\_files(self) та select\_destination(self)**

Ці методи залишаються функціонально такими ж, як і в попередній версії, відповідаючи за вибір файлів та папки призначення відповідно. Додано оновлення статусної мітки у випадку, якщо файли або папка не були обрані.

**3.2.2.5 Метод save\_settings(self)**

Цей метод зазнав змін для обробки ID папки Google Drive, введеного користувачем.

gdrive\_id = self.gdrive\_folder\_id\_var.get().strip(): Отримується значення з поля введення ID папки через змінну self.gdrive\_folder\_id\_var. Метод .get() повертає поточний текст з поля, а .strip() видаляє можливі зайві пробіли на початку та в кінці рядка.

* **Перевірка даних:**

Перевірка наявності обраних файлів (if not self.files:) залишається.

if self.backup\_to\_drive.get():: Якщо обрано збереження на Google Drive:

if not gdrive\_id:: Перевіряється, чи введено ID папки. Якщо поле порожнє, виводиться повідомлення про помилку.

elif not self.destination:: Якщо обрано локальне збереження (тобто self.backup\_to\_drive.get() повернуло False), перевіряється, чи вказано папку призначення.

* **Формування словника налаштувань:**

Python

settings = {

"files": self.files,

"backup\_to\_drive": self.backup\_to\_drive.get(),

"destination": self.destination if not self.backup\_to\_drive.get() else "",

"gdrive\_folder\_id": gdrive\_id if self.backup\_to\_drive.get() else ""

}

Словник settings тепер містить:

* + "gdrive\_folder\_id": значення, введене користувачем у поле self.gdrive\_id\_entry (якщо обрано Google Drive), або порожній рядок (якщо обрано локальний диск).
  + "destination": зберігається лише якщо обрано локальний бекап, інакше – порожній рядок.
* Збереження у файл JSON та повідомлення користувачу залишаються аналогічними, але додано блок try-except для обробки можливих помилок під час збереження файлу.