**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: “Екологічний моніторинг”

| **Студент групи** КП-01 | **Аванесов Тігран Каренович** | \_\_\_\_\_\_\_\_\_  (підпис) |
| --- | --- | --- |
| **Викладач**  **к.т.н, доцент кафедри СПіСКС** | **Радченко К.О.** | \_\_\_\_\_\_\_\_\_  (підпис) |

Захищено з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2021

# **Анотація**

Курсова робота складається з бази даних екологічних показників та консольного додатку для взаємодії з цією базою даних. Консольний додаток написано мовою програмування Python. Програма дозволяє виконувати певні операції над базою даних: зчитування, оновлення, запис, видалення, псевдовипадкова генерація записів, аналіз даних тощо.

У результаті розробки даної бази даних та даного консольного додатку було набуто практичні навички розробкисучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуто навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

# **Зміст**

Анотація 2

[Зміст](#_30j0zll) 3

[Вступ](#_1fob9te) 4

[1. Аналіз інструментарію для виконання курсової роботи](#_3znysh7) 5-6

[2. Структура бази даних](#_2et92p0) 6

[3. Опис програмного забезпечення](#_tyjcwt) 7

[3.1. Загальна структура програмного забезпечення](#_3dy6vkm) 7

[3.2. Опис основних алгоритмів роботи](#_1t3h5sf) 7

[4. Опис результатів аналізу предметної галузі](#_2s8eyo1) 8

[5. Аналіз додаткового функціоналу](#_17dp8vu) 8

[Висновки](#_3rdcrjn) 9

[Література](#_26in1rg) 10

[Додатки](#_lnxbz9) 10-20

[А. Графічні матеріали](#_35nkun2) 10-13

[Б. Фрагменти програмного коду](#_1ksv4uv) 14-17

# **Вступ**

Екологічний статус довкілля - це дуже важлива інформація, яка допомагає людям в багатьох сферах повсякденного життя. В контексті даної курсової роботи було розглянуто клімат міст України. База має такі сутності:

* Місто
* Температура повітря
* Вологість повітря
* Температура води

База даних була оптимізована шляхом індексації. Наявна функція

імпорту та експорту даних у форматі txt .

# **1. Аналіз інструментарію для виконання курсової роботи**

Для виконання даної роботи у якості системи керування базами даних було обрано PostgreSQL. Цю СУБД було обрано завдяки таким факторам:

* Відкрите ПЗ відповідає стандарту SQL - PostgreSQL - безкоштовне ПЗ з відкритим вихідним кодом.
* Підтримка великої кількості типів даних, включно з власними
* Цілісність даних з усіма необхідними обмеженнями
* Надійність, безпека
* PostgreSQL є об’єктно-реляційною СУБД, що надає певні переваги
* Сумісність з багатьма типами мереж
* Велика місткість
* Популярність даної СУБД – просто знайти вирішення потенційних проблем при розробці
* Opensource проект
* Розширення - існує можливість розширення функціоналу за рахунок своїх процедур

Для взаємодії з базою даних було обрано бібліотеку psycopg2, оскільки:

* Добре підходить для зручного використання у мові програмування Python
* Розроблена спеціально для PostgreSQL
* Найпопулярніша для взаємодії з PostgreSQL у мові програмування Python
* Має чітку, зрозумілу та вичерпну документацію з хорошими прикладами

Для візуалізації результатів аналізу даних було обрано бібліотеку tkinter, оскільки:

* Вона надає зручний інтерфейс для автоматичного будування графічних об’єктів
* Для графічних об’єктів наявна можливість дуже гнучкого налаштування з великою кількістю опцій для вигляду
* Наявна можливість будувати надзвичайно різноманітні графічні об’єкти
* Наявна чітка, зрозуміла та вичерпна документація з хорошими прикладами побудови різних графічних об’єктів з різними налаштуваннями

# **2. Структура бази даних**

База даних має такі таблиці з полями:

1. location - локація
2. temp\_air – температура повітря
3. temp\_water – температура води
4. humidity\_air– вологість повітря

# **3. Опис програмного забезпечення**

## **3.1. Загальна структура програмного забезпечення**

Розроблене програмне забезпечення містить такі компоненти:

1. База даних, що зберігає інформацію про серії, набори та мініфігурки у чотирьох таблицях
2. Засоби псевдовипадкової генерації даних
3. Засоби імпорту та експорту даних (зокрема у форматі txt)
4. Засоби аналізу даних

## 

## 

## **3.2. Опис основних алгоритмів роботи**

При псевдовипадковій генерації даних для кожної таблиці процес генерації був побудований так, аби генерувалися дані у певних (нормальних) межах для обраної предметної галузі та для обраної структури бази даних.

Для генерації даних шляхом імпорту відбувається обробка вхідних даних, для забезпечення коректного їх внесення у таблиці, оскільки стиль датасетів, обраних для цієї мети, не повністю відповідає структурі бази даних.

Для аналізу безпеки довкілля було використано максимальні допустимі значення кліматичних показників (або константи)

При аналізі виводиться інформація чи є це місто безпечним / не безпечним. Аналізована інформація експортується у форматі txt.

# **4. Опис результатів аналізу предметної галузі**

У розробленому консольному додатку наявний такий аналіз даних, що містяться у базі:

* Аналіз екологічних показників в деяких містах України та порівняння їх з нормою

# **5. Аналіз додаткового функціоналу**

У даній курсовій роботі при реалізації консольного додатку для взаємодії з базою даних було також реалізовано додатковий функціонал на додачу до того, який зазначений у вимогах до даного проекту, а саме можливості імпортування та експортування даних у базу даних та з неї у форматі txt.

Приклад роботи імпорту та експорту наведено у додатках.

# **Висновки**

Під час виконання даної курсової роботи виконано таку роботу та отримано такі результати:

* Була розроблена псевдовипадкова генерація для всіх таблиць, яка генерує значення у реалістичному діапазоні
* Були розроблені засоби для аналізу даних
* Були розроблені засоби генерації записів на основі готових датасетів (імпорт) а також засоби виводу аналізованих даних (експорт) у форматі txt
* Був розроблений зручний консольний інтерфейс

У результаті виконання даної курсової роботи було набуто практичні навички розробкисучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуто навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

Завдяки виконанню даної роботи було здобуто вміння розробляти програмне забезпечення для реляційних баз даних, відбулося оволодіння основами використання СУБД, а також інструментальними засобами підтримки розробки додатків для подібних баз даних.

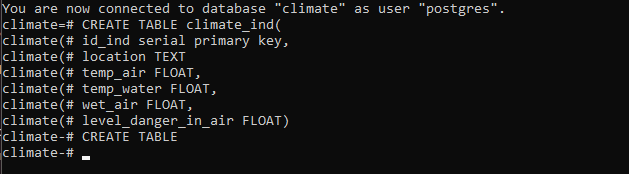
# **Література**

1. PostgreSQL 12.5 Documentation [Електронний ресурс] / The PostgreSQL Global Development Group // PostgreSQL: The World's Most Advanced Open Source Relational Database – Електрон. дані. – Режим доступу : <https://www.postgresql.org/docs/12/index.html>.
2. Python Graphical User Interfaces with Tk: <https://docs.python.org/3/library/tk.html>

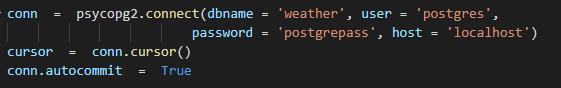
# **Додатки**

## **А. Графічні матеріали**

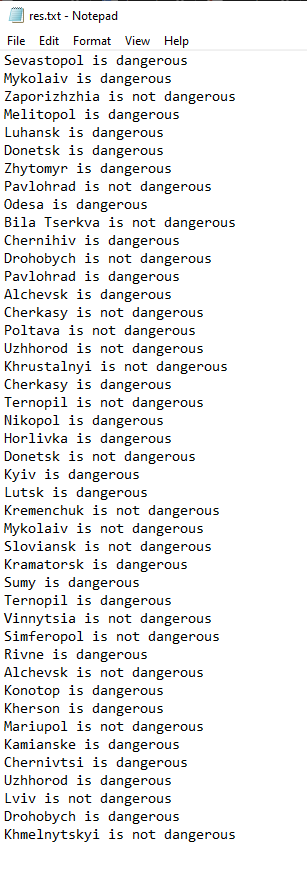
Структура бази даних



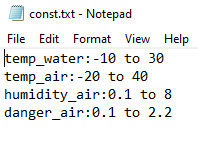
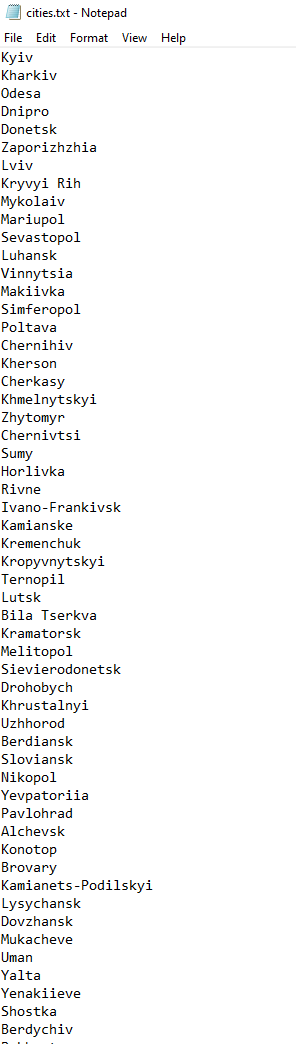
Підключення через код



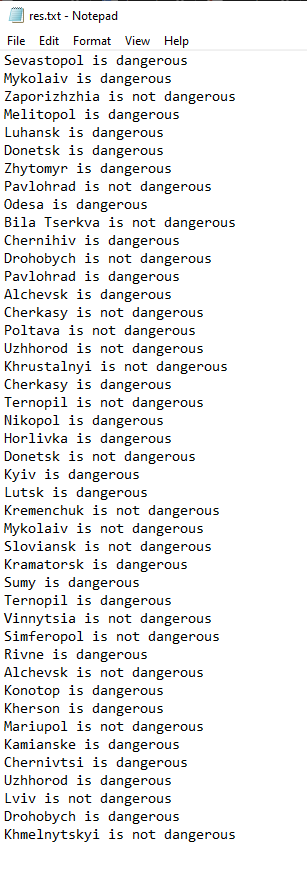
Приклад аналізу



Приклад імпорту



Приклад експорту



## **Б. Фрагменти програмного коду**

| module.py |
| --- |
| filelocals = 0  dirname = 0  fileconsts = 0  consts = 0  locals = 0  res = 0  dirname = 0 |

| weather.py |
| --- |
| import psycopg2  import random  import tkinter  from tkinter import \*  from tkinter.ttk import \*  from tkinter.filedialog import askopenfilename, askdirectory  import tkinter.font as font  import tkinter.messagebox  import module    conn = psycopg2.connect(dbname = 'weather', user = 'postgres',  password = 'postgrepass', host = 'localhost')  cursor = conn.cursor()  conn.autocommit = True      def danger\_to\_file(lst):  fin = open(module.dirname + '/res.txt', 'w')  for item in lst:  fin.write(item + "\n")    def read\_constants\_from\_file():  fin = open(module.fileconsts, 'r')  res = {}  for line in fin:  line.replace(' ', '')  lst = line.split(':')  lst\_from\_to = lst[1].split('to')  res[lst[0]] = (float(lst\_from\_to[0]), float(lst\_from\_to[1]))  return res      def get\_lst\_locals():  fin = open(module.filelocals, 'r')  lst = []  for line in fin:  lst.append(line)  return lst    def random\_input(constants):  lst = get\_lst\_locals()  for i in range(60):  locate = (lst[int(random.uniform(0,45))]).replace('\n', '')  temp\_wat = float(random.uniform((constants['temp\_water'][0]) - 1,(constants['temp\_water'][1])+1))  temp\_air = float(random.uniform((constants['temp\_air'][0]) -1,(constants['temp\_air'][1]) + 2))  humidity\_air = float(random.uniform((constants['humidity\_air'][0]),(constants['humidity\_air'][1]) + 1))  danger = float(random.uniform((constants['danger\_air'][0]),(constants['danger\_air'][1]))+1)  if humidity\_air<0:  humidity\_air = 0.01  if danger < 0:  danger = 0.01  cursor.execute("INSERT INTO weather\_stats(location, temp\_air, temp\_water, humidity\_air, danger\_air) VALUES ('{}', '{}', {}, {}, {})".format(locate, temp\_wat, temp\_air, humidity\_air, danger))      def analyze(constants):  cursor.execute('SELECT \* FROM weather\_stats')  id\_danger = []  for row in cursor:  if row[1] < ((constants['temp\_water'])[0]) or row[1] > ((constants['temp\_water'])[1]):  id\_danger.append(str(row[5]) + " is dangerous")  continue  if row[2] < ((constants['temp\_air'])[0]) or row[2] > ((constants['temp\_air'])[1]):  id\_danger.append(str(row[5]) + " is dangerous")  continue  if row[3] < ((constants['humidity\_air'])[0]) or row[3] > ((constants['humidity\_air'])[1]):  id\_danger.append(str(row[5]) + " is dangerous")  continue  if row[4] < ((constants['danger\_air']))[0] or row[4] > ((constants['danger\_air'])[1]):  id\_danger.append(str(row[5]) + " is dangerous")  continue  id\_danger.append(str(row[5]) + " is not dangerous")  dangers = set(id\_danger)  danger\_to\_file(dangers)      def clear():  cursor.execute(f"DELETE FROM weather\_stats")      def browse\_const():  module.fileconsts = askopenfilename()  module.const['text'] = 'File containing const values was chosen'      def browse\_local():  module.filelocals = askopenfilename()  module.locals['text'] = 'File with location names was chosen'    def browse\_resp():  module.dirname = askdirectory(parent = window, initialdir = "/",title = 'Please select a directory')  module.res['text'] = "Directory for result output was chosen"      def start():  clear()  constants = read\_constants\_from\_file()  random\_input(constants)  analyze(constants)  tkinter.messagebox.showinfo(title = "Success", message = "Data was analized and saved")    window = Tk()  window.geometry('680x550')  window.resizable(False, False)  window.title("Weather analysis")    myFont = font.Font(family = "Courier", size = 15, weight = "bold")    module.const = Label(window, text = "File containing const values wasn\'t chosen", font = myFont)  module.const.place(relx = 0.5, rely = 0.1, anchor = CENTER)  module.locals = Label(window, text = "File containing location names wasn\'t chosen", font = myFont)  module.locals.place(relx = 0.5, rely = 0.35, anchor = CENTER)  module.res = Label(window, text = "Directory for result output wasn\'t chosen", font = myFont)  module.res.place(relx = 0.5, rely = 0.6, anchor = CENTER)    btn\_consts = Button(text = "Choose file", width = 30, command = browse\_const).place(relx = 0.5, rely = 0.2, anchor = CENTER)  btn\_locals = Button(text = "Choose file", width = 30, command = browse\_local).place(relx = 0.5, rely = 0.45, anchor = CENTER)  btn\_locals = Button(text = "Choose folder", width = 30, command = browse\_resp).place(relx = 0.5, rely = 0.7, anchor = CENTER)  btn\_start = Button(text = "Begin analysis", width = 45, command = start).place(relx = 0.5, rely = 0.85, anchor = CENTER)    window.mainloop() |