(7, 'weather', 'sol', 'сонце'), (8, 'transportation', 'bil', 'автомобіль'), (9, 'numbers', 'två', 'два'), (10, 'food', 'mjölk', 'молоко'), (11, 'countries', 'Norge', 'Норвегія'), (12, 'colors', 'blå', 'синій'), (13, 'animals', 'katt', 'кiт'), (14, 'professions', 'kock', 'кухар'), (15, 'weather', 'regn', 'дощ'), (16, 'transportation', 'cykel', 'велосипед'), (17, 'numbers', 'tre', 'три'), (18, 'food', 'bröd', 'хліб'), (19, 'countries', 'Ukraina', 'Україна'), (20, 'colors', 'gul', 'жовтий'), (21, 'animals', 'fågel', 'птах'), (22, 'professions', 'polis', 'поліцейський'), (23, 'weather', 'moln', 'xмapa'), (24, 'transportation', 'buss', 'автобус'), (25, 'numbers', 'fyra', 'чотири'), (26, 'food', 'ost', 'сир'), (27, 'countries', 'Finland', 'Фінляндія'), (28, 'colors', 'vit', 'білий'), (29, 'animals', 'kanin', 'кролик'), (30, 'professions', 'ingenjör', 'iнженер'), (31, 'weather', 'vind', 'вітер'), (32, 'transportation', 'tåg', 'потяг'), (33, 'numbers', 'fem', 'п'ять'), (34, 'food', 'köttbullar', 'котлети'), (35, 'countries', 'Danmark', 'Данія'), (36, 'colors', 'grön', 'зелений'), (37, 'animals', 'ko', 'корова'), (38, 'professions', 'lärare', 'вчитель'), (39, 'weather', 'snö', 'сніг'), (40, 'transportation', 'flygplan', 'літак'), (41, 'numbers', 'sex', 'шість'), (42, 'food', 'pasta', 'паста'), (43, 'countries', 'Spanien', 'Іспанія'), (44, 'colors', 'svart', 'чорний'), (45, 'animals', 'häst', 'кінь'), (46, 'professions', 'fotograf', 'φοτοΓραφ'), (47, 'weather', 'regnjacka', 'плащ'), (48, 'transportation', 'motorcykel', 'мотоцикл'), (49, 'numbers', 'sju', 'сім'), (50, 'food', 'ris', 'рис') ] 1. Наповнити БД словами (мінімум 50) та їхніми перекладами, використавши інформацію з довільних доступних джерел. ID слів не повинні повторюватись (ключове поле таблиці). Кожне слово має бути прив'язане до певної категорії. In [ ]: cur.executemany("INSERT INTO vocab VALUES(?,?,?,?)", word\_data) conn.commit() conn.close() 1. Створити новий графічний (віконний) проєкт мовою Python (через Tkinter). В заголовку форми вивести власне прізвище, ім'я та номер групи, номер ЛР. In [2]: import tkinter as tk **from** tkinter **import** ttk import sqlite3 root = tk.Tk()root.title("Кличлієв Кирило, група №2, ЛР№3") 1. У вікні програми зробити 2 вкладки (Таb): «Словник» і «Про автора». У «Про автора» вивести інформацію про себе в елемент Label (або аналогічний). Шрифт — Arial, 16 пт., жирний, вирівнювання по центру форми, вміст напису — мінімум 2 рядки тексту (ПІБ, група тощо). In [ ]: import webbrowser notebook = ttk.Notebook(root) tab1 = tk.Frame(notebook)tab2 = tk.Frame(notebook)notebook.add(tab1, text='Словник') notebook.add(tab2, text='Про автора') notebook.pack(fill='both', expand=True) # Вміст для вкладки 2

text='Кличлієв Кирило Сердарович\пЧернігів, Україна',

label5 = tk.Label(tab2, text='kyrylo.klychliiev@gmail.com', font=('Arial', 16, 'underline'))

label3 = tk.Label(tab2, text='GitHub', fg='black', cursor='hand2', font=('Arial', 16, 'underline'))

label4 = tk.Label(tab2, text='LinkedIn', fg='black', cursor='hand2', font=('Arial', 16, 'underline'))

tree = ttk.Treeview(tab1, columns=('id', 'category', 'swedish\_word', 'ukrainian\_word'), show='headings')

1. Створити SQL-запит до БД, який наповнює таблицю всіма словами згідно зі структурою таблиці vocab. Запит повинен

1. Створити в проєкті під'єднання до БД, записавши у глобальну змінну шлях до файлу (абсолютний чи відносний).

font=('Arial', 16, 'bold'))

webbrowser.open('https://www.linkedin.com/in/kyrylo-klychliiev/')

webbrowser.open("https://github.com/klychliiev")

label3.bind('<Button-1>', lambda e: open\_github())

label4.bind('<Button-1>', lambda e: open\_github())

1. У вкладці «Словник» додати наступні графічні елементи:

cur.execute("SELECT DISTINCT category FROM vocab")
unique\_values = [value[0] for value in cur.fetchall()]

combo\_box = ttk.Combobox(tab1, values=unique\_values)

1. Наповнити спадний список (ComboBox) назвами категорій слів.

виконуватись відразу при запуску програми, без додаткових дій користувача.

tree.insert('', 'end', values=(idx + 1, row[1], row[2], row[3]))

1. При виведенні слів іноземною мовою в таблицю на формі відсортувати їх за абеткою за зростанням.

1. Завдання на бонусні бали: Переконатися, що при сортуванні дотримано правил, прийнятих у цій мові (зокрема порядку

The pronunciation of the names of the letters (that does not necessarily coincide with the sound it represents) is as follows:

**Letter Name** 

m

0

q

r

[εm:]

[εn:]

[u:]

[pe:]

[k<del>u</del>:]

[ær:]

In addition to the basic twenty-six letters, A-Z, the Swedish alphabet includes Å, Ä, and Ö at the end. They are distinct letters in

**Ukrainian Word** 

Іспанія

чорний

Швеція

три

два

потяг

Україна

вітер

білий

яблуко

1. Завдання на бонусні бали: Створити обробник для реакції на зміну обраної користувачем комірки в таблиці на формі, а

1. Завдання на бонусні бали: Створити обробник спадного списку з SQL-запитом: після зміни обраного варіанту таблиця

query = "SELECT \* FROM vocab WHERE category = ? ORDER BY swedish\_word COLLATE NOCASE"

· скріншоти результатів роботи, де видно, якими даними заповнені графічні елементи інтерфейсу після виконання SQL-

Файл db creation.py - створення та наповнення БД

саме: виводити в Label обране в таблиці слово іноземною мовою та через тире — його переклад.

При сортуванні правила були дотримані. У нашій базі даних є одне слово з діакритикою - äpple - яке було автоматично

Μ

Ν

O

Q

R

Letter

U

W

Χ

S

u

Name

[ES:]

[te:]

[<del>u</del>:]

[ve:]

[ɛks]

['dêb:ɛlˌve:]

Letter Name

У

å

ä

ö

[y:]

[o:]

[:3]

[ø:]

[ˈsɛ̂:ta]

Υ

Z Å

Ä

Ö

# запит до SQL-бази із для сортування шведських слів містить у функції fetch\_data

літер із діакритикою). За необхідності скористатися налаштуваннями database collation.

query = "SELECT \* FROM vocab ORDER BY swedish\_word COLLATE NOCASE"

Letter Name

g

i

k

[ge:]

[ho:]

[i:]

[ji:]

[ko:]

[ɛlː]

**Swedish Word** 

Spanien

svart

Sverige

tre

två

tăg

Ukraina

vind

vit

äpple

1. Завдання на бонусні бали: Реалізувати в таблиці смуги прокрутки.

# Функція, яка викликається при виборі елемента у таблиці

scrollbar = ttk.Scrollbar(tab1, orient="vertical", command=tree.yview)

label1.config(text=f"{swedish\_word} - {ukrainian\_word}")

очищується, і до неї завантажуються лише слова, що належать до обраної категорії.

G

Н

П

Κ

Swedish and are sorted after Z as shown above.<sup>[2]</sup>

query = "SELECT \* FROM vocab ORDER BY swedish\_word COLLATE NOCASE"

combo\_box = ttk.Combobox(tab1, values=unique\_values)

# Функція для заповнення таблиці даними

for idx, row in enumerate(data):

# Функція для отримання даних з бази даних

conn = sqlite3.connect(DB\_PATH)

cursor = conn.cursor()

cursor.execute(query)
rows = cursor.fetchall()

cursor.close()
conn.close()
return rows

fill\_table(fetch\_data())

Letters [edit]

Letter Name

a

C

d

[a:]

[be:]

[seː]

[de:]

[e:]

[εf:]

Å, Ä and Ö [edit]

винесено в кінець списку.

Category

countries

colors

countries

numbers

numbers

transportation

countries

weather

colors

food

# Створення Scrollbar

def on\_tree\_select(event):
 item = tree.focus()

# Функція для очищення таблиці

tree.delete(i)

def on\_combobox\_change(event):

cursor = conn.cursor()

rows = cursor.fetchall()

for i in tree.get\_children():

if values:

def clear\_table():

clear\_table()

cursor.close()
conn.close()

fill\_table(rows)

# Зміна події Combobox

1. Помістити у звіт із ЛР:

запитів.

In [ ]:

Код програми

import sqlite3

 $word_data = [$ 

cur = conn.cursor()

conn = sqlite3.connect('swed-ukr.db')

category TEXT, swedish\_word TEXT,

(1, 'numbers', 'ett', 'один'), (2, 'food', 'äpple', 'яблуко'),

(4, 'colors', 'röd', 'червоний'), (5, 'animals', 'hund', 'собака'),

(9, 'numbers', 'två', 'два'), (10, 'food', 'mjölk', 'молоко'),

(12, 'colors', 'blå', 'синій'), (13, 'animals', 'katt', 'кіт'),

(20, 'colors', 'gul', 'жовтий'), (21, 'animals', 'fågel', 'птах'),

(25, 'numbers', 'fyra', 'чотири'),

(28, 'colors', 'vit', 'білий'), (29, 'animals', 'kanin', 'кролик'),

(26, 'food', 'ost', 'сир'),

cur.execute('''CREATE TABLE IF NOT EXISTS vocab (id INTEGER PRIMARY KEY,

ukrainian\_word TEXT)''')

(3, 'countries', 'Sverige', 'Швеція'),

(6, 'professions', 'läkare', 'лікар'), (7, 'weather', 'sol', 'сонце'), (8, 'transportation', 'bil', 'автомобіль'),

(15, 'weather', 'regn', 'дощ'), (16, 'transportation', 'cykel', 'велосипед'),

(22, 'professions', 'polis', 'поліцейський'), (23, 'weather', 'moln', 'хмара'), (24, 'transportation', 'buss', 'автобус'),

(27, 'countries', 'Finland', 'Фінляндія'),

(30, 'professions', 'ingenjör', 'iнженер'),

(31, 'weather', 'vind', 'вітер'), (32, 'transportation', 'tåg', 'потяг'), (33, 'numbers', 'fem', 'п'ять'),

(34, 'food', 'köttbullar', 'котлети'), (35, 'countries', 'Danmark', 'Данія'),

(39, 'weather', 'snö', 'сніг'),

(41, 'numbers', 'sex', 'шість'), (42, 'food', 'pasta', 'паста'),

(49, 'numbers', 'sju', 'сім'), (50, 'food', 'ris', 'рис')

]

In [ ]:

conn.commit()

conn.close()

import tkinter as tk
from tkinter import ttk

# 3'єднання з базою даних DB\_PATH = "swed-ukr.db"

cursor = conn.cursor()

cursor.execute(query)
rows = cursor.fetchall()

# Функція для очищення таблиці

tree.delete(i)

def on\_combobox\_change(event):

cursor = conn.cursor()

rows = cursor.fetchall()

# root.geometry("600x400+100+50")

notebook.add(tab1, text='Словник')
notebook.add(tab2, text='Про автора')
notebook.pack(fill='both', expand=True)

notebook = ttk.Notebook(root)
tab1 = tk.Frame(notebook)
tab2 = tk.Frame(notebook)

# Елементи для першої вкладки

label1.pack(padx=20, pady=20)

conn = sqlite3.connect(DB\_PATH)

combo\_box.pack(padx=20, pady=10)

# Таблиця (Treeview) у вкладці 1

tree.heading('#0', text='ID')

tree.heading('#0', text='ID')
tree.heading('id', text='ID')

def on\_tree\_select(event):
 item = tree.focus()

if values:

# Зміна події Combobox

# Вміст для вкладки 2 label2 = tk.Label(tab2,

label2.pack()

label5.pack()

label3.pack()

label4.pack()

root.mainloop()

Словник

ID

1

2

3

4

5

6

7

8

9

10

ID

1

2

3

4

5

6

7

8

9

10

Словник

ID 1

2

3

4

5

6

Словник

def open\_github():

def open\_linkedin():

label2.configure(anchor='center')

fill\_table(fetch\_data())

# Створення Scrollbar

combo\_box.set("Оберіть категорію слів")

scrollbar.pack(side="right", fill="y")

tree.configure(yscrollcommand=scrollbar.set)

tree.heading('category', text='Category')

values = tree.item(item, 'values')

tree.bind('<<TreeviewSelect>>', on\_tree\_select)

swedish\_word = values[2]
ukrainian\_word = values[3]

tree.pack(fill='both', expand=True)

tree.heading('swedish\_word', text='Swedish Word')
tree.heading('ukrainian\_word', text='Ukrainian Word')

# Функція, яка викликається при виборі елемента у таблиці

combo\_box.bind('<<ComboboxSelected>>', on\_combobox\_change)

webbrowser.open("https://github.com/klychliiev")

label3.bind('<Button-1>', lambda e: open\_github())

label4.bind('<Button-1>', lambda e: open\_github())

Скріншоти результатів роботи програми:

Вигляд вкладки "Словник" після запуску програми

Про автора

Category

transportation

colors

food

transportation

transportation

countries

numbers

numbers

countries

transportation

Словник Про автора

Category

transportation

colors

food

transportation

transportation

countries

numbers

numbers

countries

transportation

Про автора

Category

colors

colors

colors

colors

colors

colors

Про автора

програми наведені вище.

переклад висвічуються в полі Label

font=('Arial', 16, 'bold'))

label1.config(text=f"{swedish\_word} - {ukrainian\_word}")

webbrowser.open('https://www.linkedin.com/in/kyrylo-klychliiev/')

Кличлієв Кирило, група №2, ЛР№3

Обране слово буде відображене тут

Оберіть категорію слів 🚚

**Swedish Word** 

bil

blå

bröd

buss

cykel

Danmark

ett

fem

Finland

flygplan

Кличлієв Кирило, група №2, ЛР№3

cykel - велосипед

Оберіть категорію слів 🖵

**Swedish Word** 

bil

blå

bröd

buss

cykel

Danmark

ett

fem

Finland

flygplan

Кличлієв Кирило, група №2, ЛР№3

cykel - велосипед

Вигляд вкладки "Словник" після сортування за категорією "colors"

colors

Вигляд вкладки "Про автора". Посилання на github та linkedin дійсні

Swedish Word

blå

grön

gul

röd

svart

vit

Кличлієв Кирило, група №2, ЛР№3

Кличлієв Кирило Сердарович Чернігів, Україна

kyrylo.klychliiev@gmail.com

GitHub

LinkedIn

text='Кличлієв Кирило Сердарович\пЧернігів, Україна',

label5 = tk.Label(tab2, text='kyrylo.klychliiev@gmail.com', font=('Arial', 16, 'underline'))

label3 = tk.Label(tab2, text='GitHub', fg='black', cursor='hand2', font=('Arial', 16, 'underline'))

label4 = tk.Label(tab2, text='LinkedIn', fg='black', cursor='hand2', font=('Arial', 16, 'underline'))

**Ukrainian Word** 

автомобіль

синій

хліб автобус

велосипед

Данія

один

п'ять

Фінляндія

літак

**Ukrainian Word** 

автомобіль

синій

хліб

автобус

велосипед

Данія

один

п'ять

Фінляндія

літак

**Ukrainian Word** 

синій

зелений

жовтий

червоний

чорний

білий

Висновки

кольори, числа, погода, країни, їжа, тварини, професії. Після формування БД swed-ukr.db (містить шведські слова з відповідними

Під час виконання лабораторної робота №3 було створення електронний словник з графічним інтерфейсом. Для цього

БД лексем заповнювалася за допомогою згенерованих ChatGPT-ом списків слів, згрупованих за 8 категоріями: транспорт,

українськими перекладами, ID слів та їхні категорії), було створено віконний проект за допомогою tkinter. Код та вигляд

використовувалася допоміжна бібліотеки webdriver для формування гіперпосилань в нашому графічному застосунку.

використовувалися бібліотеки tkinter (для створення GUI) та sqlite3 (для створення та взаємодії з БД). Також

Вигляд вкладки "Словник" після того, як користувач обирає рядок в таблиці. Шведське слово та відповідне українське слово-

\_ 🗆 🔀

8

# Combobox у вкладці 1

cur = conn.cursor()

conn.close()

for i in tree.get\_children():

# Функція для заповнення таблиці даними

for idx, row in enumerate(data):

# Функція, яка викликається при зміні Combobox

cursor.execute(query, (selected\_category,))

root.title("Кличлієв Кирило, група №2, ЛР№3")

cur.execute("SELECT DISTINCT category FROM vocab")
unique\_values = [value[0] for value in cur.fetchall()]

combo\_box = ttk.Combobox(tab1, values=unique\_values)

label1 = tk.Label(tab1, text='Oбране слово буде відображене тут')

scrollbar = ttk.Scrollbar(tab1, orient="vertical", command=tree.yview)

tree.column('category', width=100, stretch=False, anchor='center')
tree.column('swedish\_word', width=200, stretch=False, anchor='center')
tree.column('ukrainian\_word', width=200, stretch=True, anchor='center')

tree.column('#0', width=50, stretch=False, anchor='center')
tree.column('id', width=50, stretch=False, anchor='center')

selected\_category = combo\_box.get()

conn = sqlite3.connect(DB\_PATH)

cursor.close()
conn.close()
return rows

def clear\_table():

def fill\_table(data):

clear\_table()

cursor.close()
conn.close()

fill\_table(rows)

# Створення вікна root = tk.Tk()

# Створення Notebook

# Функція для отримання даних з бази даних

conn = sqlite3.connect(DB\_PATH)

import webbrowser
import sqlite3

def fetch\_data():

(36, 'colors', 'grön', 'зелений'), (37, 'animals', 'ko', 'корова'), (38, 'professions', 'lärare', 'вчитель'),

(43, 'countries', 'Spanien', 'Іспанія'),

(47, 'weather', 'regnjacka', 'плащ'),

(40, 'transportation', 'flygplan', 'літак'),

(44, 'colors', 'svart', 'чорний'), (45, 'animals', 'häst', 'кінь'), (46, 'professions', 'fotograf', 'фотограф'),

(48, 'transportation', 'motorcykel', 'мотоцикл'),

cur.executemany("INSERT INTO vocab VALUES(?,?,?,?)", word\_data)

query = "SELECT \* FROM vocab ORDER BY swedish\_word COLLATE NOCASE"

tree.insert('', 'end', values=(idx + 1, row[1], row[2], row[3]))

query = "SELECT \* FROM vocab WHERE category = ? ORDER BY swedish\_word COLLATE NOCASE"

tree = ttk.Treeview(tab1, columns=('id', 'category', 'swedish\_word', 'ukrainian\_word'), show='headings')

Файл interface.py - створення графічного інтерфейсу та взаємодія зі створеною БД

(11, 'countries', 'Norge', 'Норвегія'),

(14, 'professions', 'kock', 'кухар'),

(17, 'numbers', 'tre', 'три'), (18, 'food', 'bröd', 'хліб'), (19, 'countries', 'Ukraina', 'Україна'),

scrollbar.pack(side="right", fill="y")

tree.configure(yscrollcommand=scrollbar.set)

values = tree.item(item, 'values')

tree.bind('<<TreeviewSelect>>', on\_tree\_select)

# Функція, яка викликається при зміні Combobox

cursor.execute(query, (selected\_category,))

combo\_box.bind('<<ComboboxSelected>>', on\_combobox\_change)

· фрагменти коду проєкту, написані власноруч (обробники подій, SQL-запити тощо);

selected\_category = combo\_box.get()

conn = sqlite3.connect(DB\_PATH)

swedish\_word = values[2]
ukrainian\_word = values[3]

Α

В

C

D

Е

F

ID

41

42

43

44

45

46

47

48

49

50

In [ ]:

In [ ]:

In [ ]:

label1 = tk.Label(tab1, text='Обране слово буде відображене тут')

3BIT

із лабораторної роботи №3

з дисципліни "Технології програмування об'єктів лінгвістичної предметної галузі"

на тему: Створення електронних словників із графічним інтерфейсом

Варіант №2-05

Мета

Середовище розробки

Дані з індивідуального варіанту 2-05:

Хід роботи

1. Створити базу даних (БД) SQLite для двомовного словника (іноземна мова береться за індивідуальним варіантом — див.

1. Зробити мінімум 4 категорії, до яких належатимуть слова (наприклад, кольори, предмети, їжа і т.д.). Кожне слово в БД

У нас 8 категорій слів: transportation, colors, food, numbers, counties, animals, professions, weather

нижче). БД містить одну таблицю під назвою vocab. Структура таблиці:

При цьому типи даних обрати на власний розсуд відповідно до вмісту колонок.

Мова програмування: Python

Мова ОС: англійська

Мова: шведська

v ID;

In [ ]:

In [1]:

v категорія;

v слово іноземною мовою;

import sqlite3

word\_data = [

cur = conn.cursor()

v переклад українською мовою.

conn = sqlite3.connect('swe-ukr.db')

category TEXT,
swedish\_word TEXT,

буде належати до однієї з категорій.

(1, 'numbers', 'ett', 'один'), (2, 'food', 'äpple', 'яблуко'),

(3, 'countries', 'Sverige', 'Швеція'), (4, 'colors', 'röd', 'червоний'), (5, 'animals', 'hund', 'собака'), (6, 'professions', 'läkare', 'лікар'),

ukrainian\_word TEXT)''')

Назва та версія IDE: Visual Studio Code 1.78.2 Назва та версія ОС: Linux Mint 21.2 Cinnamon

Метою лабораторної роботи №3 є створення електронного словника з графіїчним інтерфейсом засобами мови програмування Python, а саме бібліотек tkinter (власне для створення графічного інтерфейсу) та sqlite (взаємодія з БД в Python-програмах)

In [2]

label2 = tk.Label(tab2,

label2.pack()

label5.pack()

label3.pack()

label4.pack()

v напис (Label);

In [ ]:

v спадний список (ComboBox);

# 3'єднання з базою даних DB\_PATH = "swed-ukr.db"

# Combobox у вкладці 1

cur = conn.cursor()

conn.close()

In [ ]: import sqlite3

In [ ]:

In [ ]:

In [ ]:

# Елементи для першої вкладки

label1.pack(padx=20, pady=20)

conn = sqlite3.connect(DB\_PATH)

combo\_box.pack(padx=20, pady=10)

# Таблиця (Treeview) у вкладці 1

tree.heading('#0', text='ID')

DB\_PATH = "swed-ukr.db"

cursor = conn.cursor()

def fill\_table(data):

def fetch\_data():

conn = sqlite3.connect(DB\_PATH)

combo\_box.set("Оберіть категорію слів")

v таблицю для даних.

def open\_github():

def open\_linkedin():

label2.configure(anchor='center')