МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**ЗВІТ**

**ПРО ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ №5**

з дисципліни «Програмування мовою Python»

на тему «Робота зі словником»

Виконав студент групи КН-45-5/1

Чесной В.С.

Перевірила Парфененко Ю.В.

Варіант 19

Суми 2025

**Завдання 1**

1. **Постановка задачі**

Відповідно до свого варіанту написати програму, яка створює словник для збереження даних із заданої предметної області та виконує обробку даних.

Для подання необхідної інформації можна використовувати комбінації словника та інших вбудованих об'єктів (кортежі, списки, рядки, числа тощо);

Реалізувати у вигляді окремих функцій користувача:

* виведення на екран всіх значень словника;
* додавання (видалення) нового запису до (зі) словника;
* перегляд вмісту словника за відсортованими ключами (перетворити об‘єкт подання ключів в список або скористатися функцією sorted, застосувавши її до словника, або об‘єкту, який повертає метод keys)
* розв’язання завдань відповідно до варіанту.

Словник задайте у програмному коді, кількість записів словника, якщо це не задано умовою задачі, від 5 до 10. Окрім полів, вказаних за умовою задачі, можна додавати інші поля, які описують предметну область.

Організуйте діалог по роботі зі словником.

Реалізувати обробку виключних ситуацій по введенню даних у словник та роботі зі словником, наприклад, видалення запису зі словника за неіснуючим ключем.

Задано дані про розклад поїздів, які проходять через певну станцію: номер поїзда, призначення (звідки куди, наприклад, Київ – Харків), час (години та хвилини) прибуття, час (години та хвилини) відправлення. Година та хвилина – цілі, додатні числа; значення годин не перевищує «23», значення хвилин – «59». Загальна кількість поїздів n=10. Поїзди приходять кожен день. Скласти програму, яка визначає, які поїзди (номер і призначення) стоять у визначений момент часу на станції.

1. **Текст програми**

"""

Програма для роботи з розкладом поїздів на станції

Варіант: визначення поїздів, які стоять на станції у визначений момент часу

"""

# Початковий словник з розкладом поїздів

trains\_schedule = {

"701К": {

"призначення": "Київ - Харків",

"прибуття": {"година": 8, "хвилина": 15},

"відправлення": {"година": 8, "хвилина": 45},

"тип": "швидкий"

},

"145Л": {

"призначення": "Львів - Одеса",

"прибуття": {"година": 10, "хвилина": 30},

"відправлення": {"година": 11, "хвилина": 10},

"тип": "пасажирський"

},

"89Д": {

"призначення": "Дніпро - Київ",

"прибуття": {"година": 14, "хвилина": 20},

"відправлення": {"година": 14, "хвилина": 50},

"тип": "експрес"

},

"234П": {

"призначення": "Полтава - Харків",

"прибуття": {"година": 9, "хвилина": 5},

"відправлення": {"година": 9, "хвилина": 25},

"тип": "приміський"

},

"567З": {

"призначення": "Запоріжжя - Львів",

"прибуття": {"година": 16, "хвилина": 40},

"відправлення": {"година": 17, "хвилина": 15},

"тип": "пасажирський"

},

"892Ч": {

"призначення": "Чернігів - Одеса",

"прибуття": {"година": 7, "хвилина": 50},

"відправлення": {"година": 8, "хвилина": 10},

"тип": "швидкий"

},

"321В": {

"призначення": "Вінниця - Суми",

"прибуття": {"година": 12, "хвилина": 0},

"відправлення": {"година": 12, "хвилина": 35},

"тип": "пасажирський"

},

"456М": {

"призначення": "Маріуполь - Київ",

"прибуття": {"година": 15, "хвилина": 25},

"відправлення": {"година": 16, "хвилина": 5},

"тип": "швидкий"

},

"678Х": {

"призначення": "Харків - Ужгород",

"прибуття": {"година": 11, "хвилина": 45},

"відправлення": {"година": 12, "хвилина": 20},

"тип": "пасажирський"

},

"123І": {

"призначення": "Івано-Франківськ - Київ",

"прибуття": {"година": 18, "хвилина": 30},

"відправлення": {"година": 19, "хвилина": 0},

"тип": "експрес"

}

}

def validate\_time(hour, minute):

"""Перевірка коректності введеного часу"""

if not isinstance(hour, int) or not isinstance(minute, int):

raise ValueError("Година та хвилина повинні бути цілими числами")

if hour < 0 or hour > 23:

raise ValueError("Година повинна бути в діапазоні від 0 до 23")

if minute < 0 or minute > 59:

raise ValueError("Хвилина повинна бути в діапазоні від 0 до 59")

return True

def time\_to\_minutes(hour, minute):

"""Конвертація часу у хвилини для зручності порівняння"""

return hour \* 60 + minute

def format\_time(time\_dict):

"""Форматування часу для виведення"""

return f"{time\_dict['година']:02d}:{time\_dict['хвилина']:02d}"

def display\_all\_trains(schedule):

"""Виведення на екран всіх значень словника"""

if not schedule:

print("\n❌ Розклад порожній!")

return

print("\n" + "="\*80)

print("📋 РОЗКЛАД ПОЇЗДІВ НА СТАНЦІЇ")

print("="\*80)

for train\_number, info in schedule.items():

print(f"\n🚂 Поїзд №{train\_number}")

print(f" Маршрут: {info['призначення']}")

print(f" Прибуття: {format\_time(info['прибуття'])}")

print(f" Відправлення: {format\_time(info['відправлення'])}")

print(f" Тип: {info['тип']}")

print("\n" + "="\*80)

def add\_train(schedule):

"""Додавання нового запису до словника"""

print("\n➕ ДОДАВАННЯ НОВОГО ПОЇЗДА")

print("-" \* 40)

try:

train\_number = input("Введіть номер поїзда (наприклад, 701К): ").strip()

if not train\_number:

raise ValueError("Номер поїзда не може бути порожнім")

if train\_number in schedule:

raise ValueError(f"Поїзд з номером {train\_number} вже існує в розкладі")

destination = input("Введіть призначення (наприклад, Київ - Харків): ").strip()

if not destination:

raise ValueError("Призначення не може бути порожнім")

# Введення часу прибуття

arrival\_hour = int(input("Введіть годину прибуття (0-23): "))

arrival\_minute = int(input("Введіть хвилину прибуття (0-59): "))

validate\_time(arrival\_hour, arrival\_minute)

# Введення часу відправлення

departure\_hour = int(input("Введіть годину відправлення (0-23): "))

departure\_minute = int(input("Введіть хвилину відправлення (0-59): "))

validate\_time(departure\_hour, departure\_minute)

# Перевірка, що час відправлення пізніше прибуття

if time\_to\_minutes(departure\_hour, departure\_minute) <= time\_to\_minutes(arrival\_hour, arrival\_minute):

raise ValueError("Час відправлення повинен бути пізніше часу прибуття")

train\_type = input("Введіть тип поїзда (швидкий/пасажирський/експрес/приміський): ").strip()

if not train\_type:

train\_type = "пасажирський"

# Додавання поїзда до розкладу

schedule[train\_number] = {

"призначення": destination,

"прибуття": {"година": arrival\_hour, "хвилина": arrival\_minute},

"відправлення": {"година": departure\_hour, "хвилина": departure\_minute},

"тип": train\_type

}

print(f"\n✅ Поїзд {train\_number} успішно додано до розкладу!")

except ValueError as e:

print(f"\n❌ Помилка введення: {e}")

except Exception as e:

print(f"\n❌ Непередбачена помилка: {e}")

def delete\_train(schedule):

"""Видалення запису зі словника"""

print("\n➖ ВИДАЛЕННЯ ПОЇЗДА З РОЗКЛАДУ")

print("-" \* 40)

try:

train\_number = input("Введіть номер поїзда для видалення: ").strip()

if not train\_number:

raise ValueError("Номер поїзда не може бути порожнім")

if train\_number not in schedule:

raise KeyError(f"Поїзд з номером {train\_number} не знайдено в розкладі")

# Підтвердження видалення

print(f"\n🚂 Поїзд: {train\_number} - {schedule[train\_number]['призначення']}")

confirmation = input("Ви впевнені, що хочете видалити цей поїзд? (так/ні): ").strip().lower()

if confirmation in ['так', 'yes', 'y', 'т']:

del schedule[train\_number]

print(f"\n✅ Поїзд {train\_number} успішно видалено з розкладу!")

else:

print("\n❌ Видалення скасовано")

except KeyError as e:

print(f"\n❌ Помилка: {e}")

except Exception as e:

print(f"\n❌ Непередбачена помилка: {e}")

def display\_sorted\_trains(schedule):

"""Перегляд вмісту словника за відсортованими ключами"""

if not schedule:

print("\n❌ Розклад порожній!")

return

print("\n" + "="\*80)

print("📋 РОЗКЛАД ПОЇЗДІВ (ВІДСОРТОВАНО ЗА НОМЕРОМ)")

print("="\*80)

sorted\_keys = sorted(schedule.keys())

for train\_number in sorted\_keys:

info = schedule[train\_number]

print(f"\n🚂 Поїзд №{train\_number}")

print(f" Маршрут: {info['призначення']}")

print(f" Прибуття: {format\_time(info['прибуття'])}")

print(f" Відправлення: {format\_time(info['відправлення'])}")

print(f" Тип: {info['тип']}")

print("\n" + "="\*80)

def find\_trains\_at\_station(schedule):

"""

Визначення поїздів, які стоять на станції у визначений момент часу

"""

print("\n🔍 ПОШУК ПОЇЗДІВ НА СТАНЦІЇ")

print("-" \* 40)

try:

check\_hour = int(input("Введіть годину для перевірки (0-23): "))

check\_minute = int(input("Введіть хвилину для перевірки (0-59): "))

validate\_time(check\_hour, check\_minute)

check\_time = time\_to\_minutes(check\_hour, check\_minute)

trains\_at\_station = []

for train\_number, info in schedule.items():

arrival\_time = time\_to\_minutes(

info['прибуття']['година'],

info['прибуття']['хвилина']

)

departure\_time = time\_to\_minutes(

info['відправлення']['година'],

info['відправлення']['хвилина']

)

# Поїзд на станції, якщо час перевірки між прибуттям та відправленням

if arrival\_time <= check\_time <= departure\_time:

trains\_at\_station.append({

'номер': train\_number,

'призначення': info['призначення'],

'прибуття': format\_time(info['прибуття']),

'відправлення': format\_time(info['відправлення'])

})

print(f"\n⏰ Час перевірки: {check\_hour:02d}:{check\_minute:02d}")

print("=" \* 60)

if trains\_at\_station:

print(f"\n✅ На станції знаходиться {len(trains\_at\_station)} поїзд(ів):\n")

for train in trains\_at\_station:

print(f"🚂 Поїзд №{train['номер']}")

print(f" Маршрут: {train['призначення']}")

print(f" Прибув: {train['прибуття']}, відправиться: {train['відправлення']}\n")

else:

print("\n❌ У вказаний час на станції немає поїздів")

print("=" \* 60)

except ValueError as e:

print(f"\n❌ Помилка введення: {e}")

except Exception as e:

print(f"\n❌ Непередбачена помилка: {e}")

def display\_menu():

"""Відображення головного меню"""

print("\n" + "="\*60)

print("🚉 СИСТЕМА УПРАВЛІННЯ РОЗКЛАДОМ ПОЇЗДІВ")

print("="\*60)

print("1. Показати всі поїзди")

print("2. Показати поїзди (відсортовано за номером)")

print("3. Додати новий поїзд")

print("4. Видалити поїзд")

print("5. Знайти поїзди на станції у певний час")

print("0. Вихід")

print("="\*60)

def main():

"""Головна функція програми з діалоговим інтерфейсом"""

print("\n")

print("ВІТАЄМО У СИСТЕМІ УПРАВЛІННЯ РОЗКЛАДОМ ПОЇЗДІВ!")

while True:

display\_menu()

try:

choice = input("\nВиберіть опцію (0-5): ").strip()

if choice == '1':

display\_all\_trains(trains\_schedule)

elif choice == '2':

display\_sorted\_trains(trains\_schedule)

elif choice == '3':

add\_train(trains\_schedule)

elif choice == '4':

delete\_train(trains\_schedule)

elif choice == '5':

find\_trains\_at\_station(trains\_schedule)

elif choice == '0':

print("\n")

print("Дякуємо за використання системи! До побачення!")

break

else:

print("\n❌ Невірний вибір! Будь ласка, виберіть опцію від 0 до 5")

except KeyboardInterrupt:

print("\n\n⚠️ Програма перервана користувачем")

print("До побачення!\n")

break

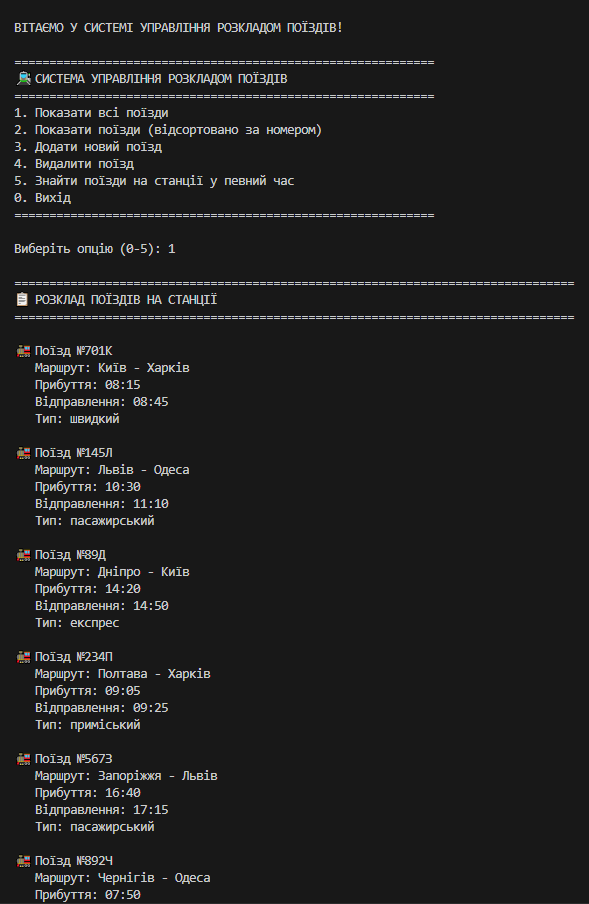
except Exception as e:

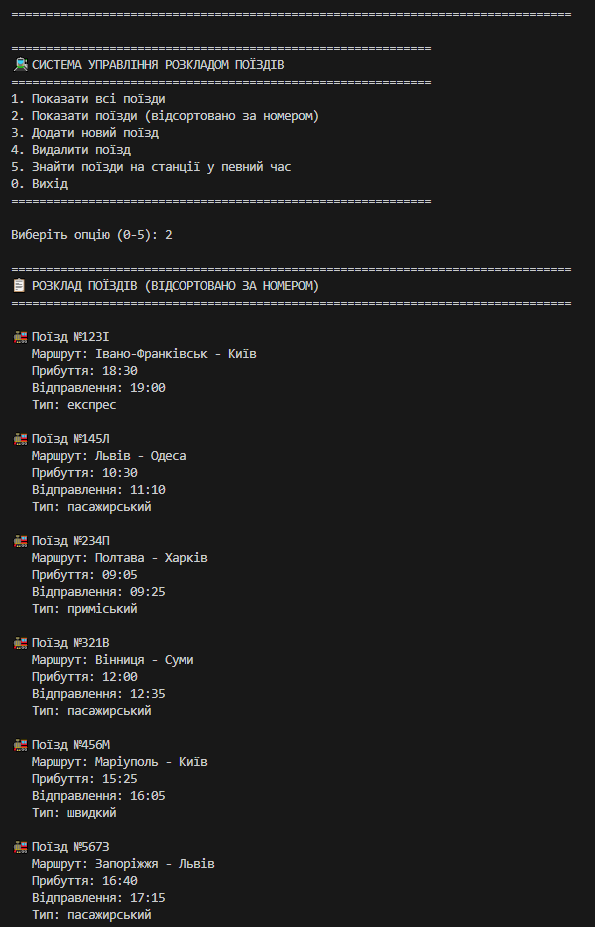
print(f"\n❌ Непередбачена помилка: {e}")

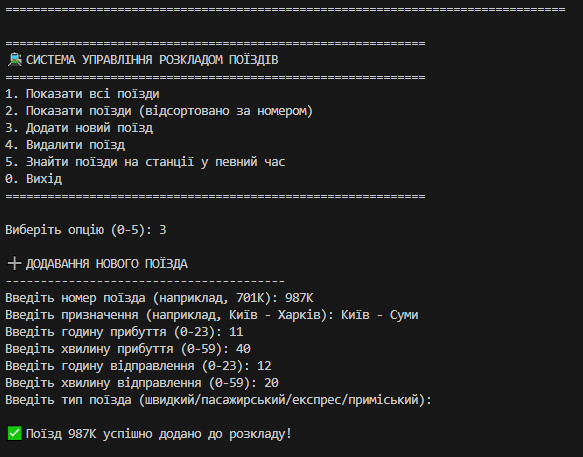
if \_\_name\_\_ == "\_\_main\_\_":

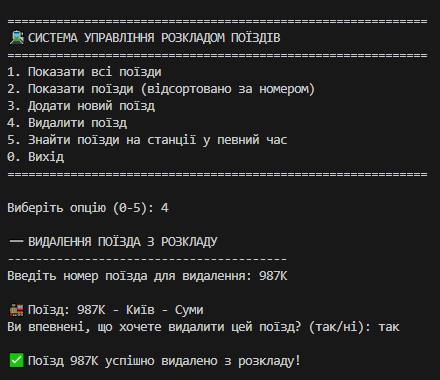
main()

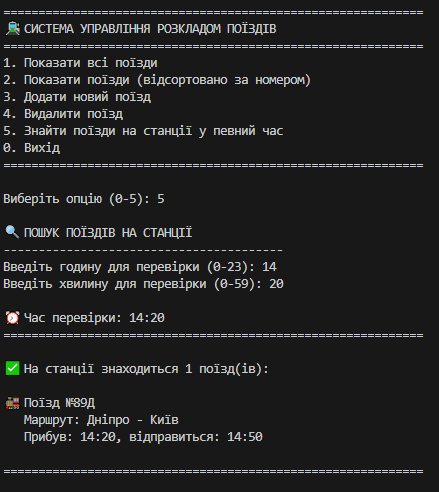
1. **Приклад роботи програми**











Посилання на GitHub з програмами [**тут**](https://github.com/TenderBly/Python-Laboratory)