Звіт з лабораторної роботи No12

Тема роботи: Створення детальної системи інвентаризації з обробкою,

аналізом та візуалізацією даних.

Виконав: Кравченко Сергій, ТЦР-12

Дата: 06.06.2025

Мета роботи:

Навчитися створювати, оновлювати та аналізувати структури для

інвентаризації товарів, використовуючи складні структури даних

(словники, списки, множини), цикли, умовні оператори та обробку

винятків.

Хід роботи:

Постановка завдання:

Завдання 1. «Інвентаризація товарів»

● Створіть список товарів у вигляді словників, кожен словник має

містити:

○ "назва товару"

○ "кількість"

○ "ціна за одиницю"

○ "категорія" (наприклад, електроніка, одяг, продукти)

● Додайте мінімум 7 різних товарів, де деякі з них повторюють

категорії.

● Виведіть створену структуру у вигляді таблиці.

Код програми: def create\_inventory():

inventory = [

{"назва товару": "Ноутбук", "кількість": 10, "ціна за одиницю": 25000, "категорія": "електроніка"},

{"назва товару": "Смартфон", "кількість": 15, "ціна за одиницю": 15000, "категорія": "електроніка"},

{"назва товару": "Футболка", "кількість": 30, "ціна за одиницю": 500, "категорія": "одяг"},

{"назва товару": "Джинси", "кількість": 20, "ціна за одиницю": 800, "категорія": "одяг"},

{"назва товару": "Яблука", "кількість": 50, "ціна за одиницю": 20, "категорія": "продукти"},

{"назва товару": "Хліб", "кількість": 40, "ціна за одиницю": 25, "категорія": "продукти"},

{"назва товару": "Навушники", "кількість": 8, "ціна за одиницю": 1200, "категорія": "електроніка"}

]

return inventory

def display\_inventory(inventory):

print("\n{:<15} {:<10} {:<15} {:<15}".format("Назва", "Кількість", "Ціна за од.", "Категорія"))

print("-" \* 55)

for item in inventory:

print("{:<15} {:<10} {:<15} {:<15}".format(

item["назва товару"],

item["кількість"],

item["ціна за одиницю"],

item["категорія"]

))

inventory = create\_inventory()

display\_inventory(inventory)

Завдання 2. «Складний пошук та редагування»

● Реалізуйте пошук товарів за:

○ назвою

○ категорією

● Користувач повинен мати можливість оновити кількість або ціну

товару після знаходження.

● Передбачте обробку помилок, коли товар не знайдено, або коли

користувач вводить некоректні значення при оновленні інформації.

Код програми:

def search\_by\_name(inventory, name):

found\_items = [item for item in inventory if item["назва товару"].lower() == name.lower()]

return found\_items

def search\_by\_category(inventory, category):

found\_items = [item for item in inventory if item["категорія"].lower() == category.lower()]

return found\_items

def update\_item(item, field, new\_value):

if field in item:

try:

if field == "назва товару" or field == "категорія":

item[field] = new\_value

else:

item[field] = int(new\_value)

print("Оновлення успішне!")

return True

except ValueError:

print("Помилка: некоректне значення для числового поля")

return False

else:

print("Помилка: такого поля не існує")

return False

def search\_and\_edit\_interface(inventory):

while True:

print("\nМеню пошуку та редагування:")

print("1. Пошук за назвою товару")

print("2. Пошук за категорією")

print("3. Повернутися до головного меню")

choice = input("Виберіть опцію: ")

if choice == "1":

name = input("Введіть назву товару для пошуку: ")

found\_items = search\_by\_name(inventory, name)

elif choice == "2":

category = input("Введіть категорію для пошуку: ")

found\_items = search\_by\_category(inventory, category)

elif choice == "3":

break

else:

print("Невірний вибір, спробуйте ще раз")

continue

if not found\_items:

print("Товари не знайдено")

continue

print("\nЗнайдені товари:")

display\_inventory(found\_items)

if len(found\_items) == 1:

item = found\_items[0]

edit\_choice = input("Бажаєте редагувати цей товар? (так/ні): ").lower()

if edit\_choice == "так":

field = input("Введіть поле для оновлення (назва товару/кількість/ціна за одиницю/категорія): ")

new\_value = input("Введіть нове значення: ")

update\_item(item, field, new\_value)

else:

print("Знайдено декілька товарів. Будь ласка, уточніть пошук.")

Завдання 3. «Аналітика складу та фінансів»

● Використовуючи цикли та словники, реалізуйте аналітичні функції:

○ Підрахуйте загальну вартість усіх товарів по кожній категорії

(кількість × ціна).

○ Визначте категорію, яка має найбільшу сумарну вартість

товарів.

○ Виведіть товари, кількість яких нижче заданого рівня

(наприклад, нижче 5 штук), для подальшого замовлення.

● Виведіть результати аналітики в зручній

Код програми:

def calculate\_category\_totals(inventory):

category\_totals = {}

for item in inventory:

category = item["категорія"]

total\_value = item["кількість"] \* item["ціна за одиницю"]

if category in category\_totals:

category\_totals[category] += total\_value

else:

category\_totals[category] = total\_value

return category\_totals

def find\_most\_valuable\_category(category\_totals):

if not category\_totals:

return None

return max(category\_totals.items(), key=lambda x: x[1])

def find\_low\_stock\_items(inventory, threshold=5):

return [item for item in inventory if item["кількість"] < threshold]

def analytics\_interface(inventory):

while True:

print("\nМеню аналітики:")

print("1. Загальна вартість по категоріям")

print("2. Найцінніша категорія")

print("3. Товари з низькими запасами")

print("4. Повернутися до головного меню")

choice = input("Виберіть опцію: ")

if choice == "1":

category\_totals = calculate\_category\_totals(inventory)

print("\nЗагальна вартість товарів по категоріям:")

for category, total in category\_totals.items():

print(f"{category}: {total} грн")

elif choice == "2":

category\_totals = calculate\_category\_totals(inventory)

if category\_totals:

category, total = find\_most\_valuable\_category(category\_totals)

print(f"\nНайцінніша категорія: {category} з сумарною вартістю {total} грн")

else:

print("Немає даних для аналізу")

elif choice == "3":

threshold = input("Введіть поріг для низьких запасів (за замовчуванням 5): ")

try:

threshold = int(threshold) if threshold else 5

low\_stock = find\_low\_stock\_items(inventory, threshold)

if low\_stock:

print(f"\nТовари з запасами менше {threshold}:")

display\_inventory(low\_stock)

else:

print(f"Немає товарів з запасами менше {threshold}")

except ValueError:

print("Помилка: некоректне значення порогу")

elif choice == "4":

break

else:

print("Невірний вибір, спробуйте ще раз")

Висновки

У ході виконання лабораторної роботи було створено систему інвентаризації товарів з використанням структур даних (словників та списків), реалізовано функції для пошуку, редагування та аналітики даних. Програма дозволяє:

1. Відстежувати товари (назва, кількість, ціна, категорія) у зручному табличному вигляді.

2. Шукати та змінювати інформацію про товари за назвою або категорією з обробкою помилок.

3. Аналізувати дані (загальна вартість за категоріями, визначення найприбутковішої категорії, пошук товарів з низькими запасами).

Робота демонструє ефективне використання циклів, умовних операторів та обробки винятків у Python для керування складом. Отримані навички можна застосовувати для розробки більш складних систем обліку та аналітики даних.